QUANTUM DYNAMICS IN LASER–ASSISTED COLLISIONS, LASER–MOLECULE INTERACTIONS, AND PARTICLE–SURFACE SCATTERING

by

THOMAS NIEDERHAUSEN

Vordiplom, Justus–Liebig–Universität Gießen, Germany, 2000
Diplom, Universität Kassel, Germany, 2004

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Physics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2007

Abstract

The time-dependent Schrödinger equation is integrated on a numerical lattice for up to three-dimensional problems. The wave packet propagation technique has been applied to ion – atom collisions in a strong laser field, the vibrational nuclear motion in small homonuclear diatomic molecular ions, and for the scattering of an ion in front of a metallic surface. For laser-assisted proton – hydrogen collisions it is shown, that strong circularly polarized radiation significantly alters the capture and ionization probabilities and results in a dichroism with respect to the helicity. In a pump – control – probe scheme, "stroboscopic" exposure of a nuclear wave packet of the deuterium molecular ion by a single or a series of short and intense laser control pulses may be used to produce an almost stationary distribution of a single vibrational level, where the nodal structure can be tested using the Coulomb explosion imaging technique. Using a pump – probe setup with variable probe delays it is proposed to use Fourier analysis of the time dependence of the Coulomb explosion kinetic energy release spectrum to reveal insight into the initial vibrational state distribution for small diatomic molecules. A last application demonstrates, that resonant charge transfer for scattering of a negative hydrogen anion on a metal surface depends crucially on the position of surface and image states relative to the conduction and valence band, thereby implying different reaction mechanisms for different surface cuts of a metal.

QUANTUM DYNAMICS IN LASER–ASSISTED COLLISIONS, LASER–MOLECULE INTERACTIONS, AND PARTICLE–SURFACE SCATTERING

by

Thomas Niederhausen

Vordiplom, Justus–Liebig–Universität Gießen, Germany, 2000
Diplom, Universität Kassel, Germany, 2004

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Physics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2007

Approved by:

Major Professor
Uwe Thumm

Copyright

Abstract

The time-dependent Schrödinger equation is integrated on a numerical lattice for up to three-dimensional problems. The wave packet propagation technique has been applied to ion – atom collisions in a strong laser field, the vibrational nuclear motion in small homonuclear diatomic molecular ions, and for the scattering of an ion in front of a metallic surface. For laser-assisted proton – hydrogen collisions it is shown, that strong circularly polarized radiation significantly alters the capture and ionization probabilities and results in a dichroism with respect to the helicity. In a pump – control – probe scheme, "stroboscopic" exposure of a nuclear wave packet of the deuterium molecular ion by a single or a series of short and intense laser control pulses may be used to produce an almost stationary distribution of a single vibrational level, where the nodal structure can be tested using the Coulomb explosion imaging technique. Using a pump – probe setup with variable probe delays it is proposed to use Fourier analysis of the time dependence of the Coulomb explosion kinetic energy release spectrum to reveal insight into the initial vibrational state distribution for small diatomic molecules. A last application demonstrates, that resonant charge transfer for scattering of a negative hydrogen anion on a metal surface depends crucially on the position of surface and image states relative to the conduction and valence band, thereby implying different reaction mechanisms for different surface cuts of a metal.

Table of Contents

List of Figures

List of Tables

Acknowledgments

After spending almost six years in Manhattan, Kansas, living in two different continents, and having to deal with three time zones, it all ends now with the thesis at hand. And there are clearly many people I would like to thank for their support, help and friendship, and for their contribution for the work presented here. So all the credits go to:

- *Prof. Uwe Thumm* for welcoming me in his group and providing a virtually unlimited pool of ideas and research material on possible calculations that should be investigated in more detail, and for his great help when things often did not work out the way they should.

- My supervisory committee, *Prof. Anil Pahwa*, *Prof. Todd Cochrane*, *Prof. Vincent Ortiz*, and in particular *Prof. Lew Cocke* for his expertise on all experimental issues and several great ideas on the laser-molecule calculations, *Prof. Brett Esry* for all the valuable input on this thesis.

- All the hard workers of Cardwell Hall think tank and the James R. Macdonald Lab for the great discussions about the world, the universe, the Autobahn and of course - Physics – including (but not limited to) *Ali Alnaser*, *Bernold Feuerstein*, *Birte Ulrich*, *Boyan Obreshkov*, *Prof. Chii-Dong Lin*, *Fatima Anis*, *Himadri Chakraborty*, *Jose D'Incao*, *Marlene Wickenhauser*, *Michael Bromley*, *Remigio Cabrera-Trujillo*, *Stefan Voss*, and *Vladimir Roudnev*.

- All the great roommates of the "French House" during the last years from all over the world, promoting insight in different places and cultures: *André Liebler*, *Anthony Bodin*, *Audrey Murat*, *Edouard Baudon*, *Elie Helou*, *Flor Ponciano*, *Florian Kneiding*, *Karel Princ*, *Kerstin Würges*, *Laure Meyer Goyheneix*, *Laure Mora*, *Marine Cozic*, *Oliver Larche*, *Petra Sedlmajerova*, *Raymond Kassatly*, *Ueli Honegger*, *Victoria Elguina*, *Vincent Marque*, *Yuliya Zinova* and many more...

- The international students in Manhattan and some true friends keeping me sane: *Allen Adriani*, *Andrea Müller*, *Casey Quinn*, *Eddy Guliford*, *Isabell Gmal*, *Jana Striova*, *Junia Melin*, *Kevin Spears*, *Natascha Sidorovskaya*, *Nora Redhardt*, and *Smita Kapur* among so many other people that came to Manhattan for only a semester or two...

- The friendly people in the offices helping to keep all the good research and education at Kansas State University running, including the Graduate School, the International Student Center and of course the Physics Department.

Besonderer Dank gilt ebenfalls meinen Eltern, Großeltern und Geschwistern für die fortwährende Unterstützung meines Aufenthaltes in den Vereinigten Staaten und des gesamten Studiums.

И самое важное, я хочу выразить особую сердечную благодарность своей замечательной жене Зое Костюченко за всю ее любовь и терпение. Я смотрю в будущее с предвкушением прекрасных лет, которые ждут нас впереди.

Manhattan, Kansas

May 4$^{\text{th}}$ 2007

Dedication

To my wife, *Зоя Костюченко*.

«Simplicity is the ultimate sophistication. »
Leonardo da Vinci

Chapter 1

An Ultra-Short Introduction

In the last few decades, the scientific field of atomic, molecular and optical (AMO) physics has seen major innovative advances. Several frontiers have emerged with the ultimate goal of a better control and understanding of light and matter on the atomic length scale: ultra-cold physics leading to the experimental discovery of Bose-Einstein condensation, non-linear optical phenomena used to produce ultra-fast laser pulses or ultra-slow and even stopped light, quantum-information and coherent control schemes for the next generation computer, the quantum-computer, or nano-structured materials with enormous application potential in the applied sciences to name few topics. *Ultra* could be the defining term of present AMO physics, as it reflects the current search for the superlatives: ultra-fast, ultra-cold, ultra-precise, ultra-stable, ultra-slow or ultra-intense in the scientific literature are almost exclusively linked to the field of atomic, molecular and optical physics.

The novel progress of trapping atoms, using lasers and magnetic fields, and cooling the sample with laser-, evaporative- and magnetic-refrigeration techniques to temperatures on the nanokelvin scale, has set the record for the coldest temperature on earth[Leanhardt 03]. Consequently this allowed for the experimental confirmation of the existence of Bose-Einstein condensates[Anderson 95], where the particle wave functions collapse into a single coherent collective ground state – around 70 years after its theoretical prediction by *Satyendra Nath Bose* and *Albert Einstein* in 1925. Moreover, the technique of creating ultra-cold atoms has

laid the foundation for the investigation of other exciting disciplines such as matter waves and superfluidity[Anglin 02], or Fermi-condensates[Jochim 03].

The field of non-linear optics (for a general overview see for example the textbook by *Robert Boyd*[Boyd 02]) has opened the doorway to the latest developments in laser technology, and also lead to some counter-intuitive results: An absorption window of a medium can be turned transparent over a very narrow spectral range by the use of a strong coupling laser[Harris 97], resulting in extreme changes in the index of refraction within the resonance. This has been used for producing ultra-slow group velocities of a light pulse[Hau 99], and – as the ultimate slow-down – to completely stop a pulse in the medium for later retrieval[Liu 01]. Other pioneering work has been archived in the field of ultra-short laser pulses, where the phase of the the electric field relative to the pulse envelope becomes important[Brabec 00,Paulus 03,Roudnev 04,Tong 07]. Laser pulses in the visible regime with a pulse duration of 1.6 fs – slightly less than a single optical cycle – have been been reported[Shverdin 05], as well as down to 130 attosecond pulses in the extreme ultraviolet range[Sansone 06]. Mode-locking[Boyd 02] of femtosecond-lasers has allowed for utilizing the comb of laser-mode frequencies, spaced by the repetition rate, for highly precise optical spectroscopy with an accuracy several units of magnitude better than with current cesium atomic clock standards[Udem 02]. Nano-structured materials, in particular photonic band gap fibers[Russell 03] exhibit strong non-linearities that have been employed for a spectral broadening of the driving laser pulses, known as super-continuum generation[Dudley 06], a process relevant for the stabilization of the laser phase relative to the pulse envelope[Jones 00,Baltuška 03,Moon 06]. Apart from interest within the AMO community, commercial applications of the ultra-precise frequency metrology such as improved satellite navigation, atomic clocks and telecommunication are obvious.

With such powerful cooling, trapping and laser technology becoming available, the interest of AMO physics has slowly shifted from the observation of static properties towards the study of dynamical processes and to the control of reaction pathways[Weber 05,Hertel 06]. Time-resolved pump – probe experiments have been developed, where a first ultra-short

laser pulse induces a reaction to create a controlled initial state and a second ultra-short pulse probes the reaction outcome, for example by photo-emission, fluorescence spectroscopy or fragmentation[Zewail 00]. Powerful imaging techniques, such as recoil ion momentum spectroscopy[Cocke 91,Ullrich 97,Dörner 00,Ullrich 03] emerged in the last decade and allow for the coincident measurement of the momentum vectors of all the reaction fragments in ionizing collisions of atoms, molecules, electrons or photons. Combined with time-delay measurements using ultra-short pump – probe laser pulses, the dynamics of quantum systems can be resolved on the time scale of the nuclear motion[Niikura 02,Baltuška 03,Niikura 03,Lin 06].

1.1 Historical Overview

The origin of our understanding of matter dates back to ancient Greek philosophers. *Democritus* believed that every material is made from small units "atomos" or "atomon" - the Greek word for indivisible - embedded in the "void". And while these atoms could be characterized by a few basic properties, the diversity of matter that we experience, like color or taste is due to complicated interactions between these atoms: *«Sweet exists by convention, bitter by convention, color by convention; but in reality atoms and the void alone exist.»* It took more than two millenniums until it was discovered that even the inseparable atoms have a substructure, consisting of electrons and nuclei. Yet the atomic core itself exhibits a rather complex entity of quarks, gluons and gauge bosons and is the subject of extensive investigation in the field of high energy physics in some of the largest up to date experiments[Seife 03].

A fundamental milestone in the history of atomic physics were the early experiments of *Hans Geiger* and *Ernest Marsden* on the scattering of α-particles by a thin gold foil in 1909. It was expected that these particles would pass through the foil and get deflected by the gold atoms by a few degrees at most, but a small fraction of the scattered projectiles were found at larger angles and even scattered back. This gave rise to the famous quote by *Ernest Rutherford* when he was analyzing the experimental findings in 1911: *«It was almost*

as incredible as if you fired a fifteen-inch shell at a piece of tissue paper and it came back and hit you.» Applying laws of energy and momentum conservation, Rutherford realized, that the only possible explanation was the presence of a charged and heavy nucleus of finite size, which subsequently lead to the Rutherford model of the atom, a precursor to the Bohr model. Shortly thereafter *James Franck* and *Gustav Ludwig Hertz* measured the kinetic energy transfer of accelerated electrons with mercury vapor in a gas cell in 1914. They discovered that at certain kinetic energies of the electrons inelastic collision occur and thereby impressively demonstrated the quantization of the energy levels in an atom. A theory justifying the quantization of energies was missing, and the Bohr model provided only a phenomenological explanation.

In 1924 the French physicist *Louis de Broglie* wrote his famous dissertation *«Recherches sur la théorie des Quanta»* suggesting the wave-particle dualism for all particles. His audacious proposal was soon verified experimentally by *Clinton Davisson* and *Lester Germer* showing an interference pattern for the diffraction of an electron beam on a crystal. The concept of matter waves was born, and *Werner Heisenberg*, *Max Born*, and *Pascual Jordan* provided a theoretical framework – the matrix mechanics. Their theoretical formulation is equivalent to the *Schrödinger equation* and is the basis of the non-relativistic description of quantum dynamics.

The principle of the light amplification by stimulated emission of radiation, laser, originates from the A and B-coefficients for absorption and emission of light, introduced by *Albert Einstein's* "quantum theory of radiation" Einstein 17. Optical pumping was first proposed by *Alfred Kastler* in 1950, and in collaboration with *Jean Brossel* and *Jacques Winter* verified two years afterwards. The first working laser was constructed by *Theodore Maiman* in 1960 using optical pumping of a ruby crystal Maiman 60. Today lasers can be found in many household appliances, including CD and DVD drives, barcode scanners, printers and laser pointers and many other devices.

1.2 Chapter Outline

As in classical mechanics where already the coupled motion of the planets in our solar system can not be solved analytically in general, the same holds true for the Schrödinger equation for more than two particles. In general, numerical approximation schemes need to be implemented and often require large amounts of computing time. Simultaneously, as computing resources become more and more available, more complex systems can be treated numerically.

In the present work the theoretical tools to calculate collisions and half-collisions for atomic and molecular systems that are governed by a single active particle Schrödinger equation are presented in Chapter 2. Since for a general partial differential equation such as the time-dependent Schrödinger equation an analytical solution exists only for a few special cases, a numerical method that is based on discretizing operators and space on a lattice is applied, together with a finite-differencing scheme for the time propagation. A comprehensive summary of this method – based on the Crank-Nicholson algorithm – and several other theoretical aspects is presented, including details such as the proper treatment of the numerical boundaries, or finding the eigenfunctions for a time-independent potential.

In the following two chapters, the Crank-Nicholson propagation method is applied for the novel studies of laser-assisted collisions. From the theoretical side, the most simple collision scenario, namely a proton colliding with a hydrogen atom, is investigated with the focus on the charge-transfer and ionization probabilities. This work is motivated by the possibility to steer chemical reactions towards a preferred reaction channel by external electromagnetic radiation. To keep the computational effort manageable, in Chapter 3 a two-dimensional model is employed that restricts the motion of the electron and the plane of the electric field vector of the laser to the scattering plane. This study allows to predict qualitatively the influence of the radiation field on the reaction. For a circularly polarized laser it is found, that both capture and ionization differ significantly from field free collisions[Niederhausen 04b,Niederhausen 04a]. In Chapter 4 the laser-assisted collision scenario

is reinvestigated but in full dimensionality. Clearly, this three-dimensional ab-initio study is computationally much more demanding, and rigorous optimizations of the program, including parallel-processing and the use of distributed computations were implemented to perform this calculation. Strikingly, a very good qualitative agreement was found with the reduced dimensionality calculations[Niederhausen 06].

The two chapters thereafter describe the control and analysis of the vibrational nuclear motion in the deuterium molecular ion. Starting with the diatomic molecule in the ground state, an ultra-short pump pulse launches the nuclear wave packet on the adiabatic nuclear potential curve and thereby starts the molecular clock[Alnaser 04]. Chapter 5 discusses the influence of a single or a series of time delayed ultra-short control pulses on the vibrational state distribution and the dissociation probability. Based on these results, a method similar to the vibrational cooling scheme of *Niikura et al.*[Niikura 04] is suggested to create an almost stationary probability distribution of the nuclear wave function[Niederhausen 07]. In Chapter 6 an imaging scheme is suggested that allows one to extract the nuclear vibrational level occupation and to analyze the dynamics of the motion in an electromagnetic radiation field from a Fourier-transformation of a time-series pump-probe study[Feuerstein 07].

Chapter 7 analyzes the role of the surface state and image states with respect to the positions of valence and conduction bands on the neutralization probability for ion – surface scattering. It is found that a negative hydrogen anion has the largest ion survival probability, if the surface features a localized surface state within the band gap of the metal and possesses energetically overlapping image states within the conduction band[Chakraborty 04a,Chakraborty 04b,Chakraborty 05].

The last Chapter 8 summarizes the results presented in this dissertation, followed by three appendices containing a brief overview on the Atomic Units used throughout this work, an Alternative Derivation of the Short-Time Propagator and the Source Code Listings for a few selected program sections.

Chapter 2

Theory

«[Do not] put too much confidence in experimental results until they have been confirmed by theory. »
Sir Arthur Eddington

2.1 The Time-Dependent Schrödinger Equation

The starting point for the quantum mechanical description of atoms and molecules in intense laser fields, colliding with each other, or their interactions with metallic surfaces is the solution of the time-dependent Schrödinger equation. In atomic physics it is very useful to first introduce a set of *atomic units* (see Appendix A) with $e = \hbar = m_e = 1$, which correspond to the typical dimensions found in atomic systems, more specifically, the hydrogen atom. In atomic units, the time-dependent Schrödinger equation for a one-particle electronic wave function $\Psi(\mathbf{r}, t)$ has the simple form:

$$i\frac{\partial}{\partial t}\Psi(\mathbf{r}, t) = \hat{H}\Psi(\mathbf{r}, t), \tag{2.1}$$

with the Hamiltonian

$$\hat{H} = -\frac{\nabla^2}{2} + \hat{V}(\mathbf{r}, t). \tag{2.2}$$

Equation (2.1) can be formally integrated (see quantum mechanics textbooks[Nolting 97]) to obtain

$$\Psi(\mathbf{r}, t) = \hat{\mathscr{T}} \exp\left(-i\int_{t_0}^{t} \hat{H}(t')\mathrm{d}t'\right)\Psi(\mathbf{r}, t_0), \tag{2.3}$$

with the chronological time-ordering operator $\hat{\mathscr{T}}$. A more detailed description can be found in Appendix B. For a time-independent problem, i.e. $\hat{V}(\mathbf{r},t) = \hat{V}(\mathbf{r})$, the time-ordering and the integration become trivial:

$$\Psi(\mathbf{r},t) = \mathrm{e}^{-i\hat{H}t}\Psi(\mathbf{r},t_0) \tag{2.4}$$

For the general case of a time-dependent Hamiltonian, $t \to \Delta t$ can be replaced by a time step Δt, during which the Hamiltonian $\hat{H}(t+\Delta t) \approx \hat{H}(t)$ can be approximated as constant in time. Thus, equation (2.3) is approximated by

$$\Psi(\mathbf{r},t+\Delta t) = \mathrm{e}^{-i\hat{H}(t)\Delta t}\Psi(\mathbf{r},t), \tag{2.5}$$

and the full time-dependence of the wave function $\Psi(\mathbf{r},t)$ can be obtained from iteratively propagating small time steps, $t = N\Delta t$.

For a typical time-dependent quantum mechanical problem one can distinguish grid based methods and basis set expansion methods (see for example the review articles by *Kirchner et al.*[Kirchner 99] or *Pindzola et al.*[Pindzola 07]). The expansion method employs a finite set of basis functions, with the short time-propagation (2.5) introducing couplings between the expansion coefficients of the basis states. The difficulty lies in the proper choice of the basis set from a variety of atomic or molecular orbitals, Coulomb waves, Airy functions, plane waves, spherical harmonics, Gaussians, splines etc. of different energies, momenta and expansion centers. Calculations often use several hundreds of basis states in order to correctly describe all the relevant physics, such as electron transfer[Wang 03] or ionization[Zou 02] within the specified accuracy.

Grid methods (see for example the review by *R. Kosloff*[Kosloff 96]) are based on the direct solution of (2.5) by discretizing the operators and wave functions on a spatial and/or momentum lattice. Although this approach is computationally very demanding as the dimensionality and the number of grid point increases, recent developments in computing technology permit one to perform such calculations on personal computers with the flexibility for applying grid models to almost any one-particle potentials. The spatial accuracy is

only limited by the size of the discretized box and the spacing of the lattice. It can be partially attributed to the flexibility of grid based methods and the increasing availability of fast and large-memory computers, that the direct solution of the time-dependent Schrödinger equation has attracted rising interest in the last two decades for calculating for example photoelectron spectra[Krause 92], heavy-particle impact ionization[Schultz 03], charge transfer in ion-atom collisions[Minami 06], or laser-assisted collisions[Niederhausen 06,Anis 06].

Another approach that has been extensively used in electron scattering by atoms or molecules in the presence of an external field (see for example the review by *Ehlotzky*, *Jaroń* and *Kamiński*[Ehlotzky 98]) is the R-matrix-Floquet method[Burke 91], where the time-dependence of a periodic Hamiltonian can be eliminated using the Floquet representation[Bransden 03]. The (time-independent) R-matrix method divides the space into an inner volume and an outer volume. Typically the interaction in the external region is weak and often simple analytic solutions exist. The reaction volume is treated numerically, for example using a basis expansion or a finite grid. Both regions are linked by matching the logarithmic derivative of the wave functions at the boundary. Originally, the method has been introduced for the theory of nuclear resonance reactions[Wigner 46,Lane 58] and has been extended for atomic scattering systems[Burke 91,Thumm 91]. Other splitting methods use overlapping inner and outer regions to allow for a transition of the wave packet in the boundary region[Keller 95,Chelkowski 96] such that explicitly time-dependent problems can be studied.

2.2 Time Propagation

To obtain the wave function for a given initial state $\Psi(\mathbf{r}, t)$ for a small time step Δt later, one can write the propagated wave function $\Psi(\mathbf{r}, t + \Delta t)$ in a Taylor-expansion and apply

the Schrödinger equation (2.1) for the first time derivative:

$$\Psi(\mathbf{r}, t+\Delta t) = \Psi(\mathbf{r}, t) + \frac{\partial}{\partial t}\Psi(\mathbf{r}, t)\Delta t + \frac{1}{2}\frac{\partial^2}{\partial t^2}\Psi(\mathbf{r}, t)\Delta t^2 + \hat{\mathscr{O}}(\Delta t^3)\Psi(\mathbf{r}, t) \tag{2.6}$$

$$= \Psi(\mathbf{r}, t) - i\hat{H}(t)\Psi(\mathbf{r}, t)\Delta t - \frac{i}{2}\frac{\partial}{\partial t}\left(\hat{H}(t)\Psi(\mathbf{r}, t)\right)\Delta t^2 + \hat{\mathscr{O}}(\Delta t^3)\Psi(\mathbf{r}, t) \tag{2.7}$$

$$= \left(\hat{\mathbf{1}} - i\hat{H}(t)\Delta t - \frac{1}{2}\left(\hat{H}^2(t) + i\frac{\partial \hat{H}(t)}{\partial t}\right)\Delta t^2 + \hat{\mathscr{O}}(\Delta t^3)\right)\Psi(\mathbf{r}, t) \tag{2.8}$$

A short time propagator can thus be defined as

$$\hat{U}(t+\Delta t, t) \equiv \hat{\mathbf{1}} - i\hat{H}(t)\Delta t - \frac{1}{2}\left(\hat{H}^2(t) + i\frac{\partial \hat{H}(t)}{\partial t}\right)\Delta t^2. \tag{2.9}$$

such that (2.8) becomes

$$\Psi(\mathbf{r}, t+\Delta t) = \hat{U}(t+\Delta t, t)\Psi(\mathbf{r}, t) + \hat{\mathscr{O}}(\Delta t^3). \tag{2.10}$$

The time-evolution operator (2.9) is not unitary, i.e. $U^\dagger U \neq \hat{\mathbf{1}}$, and numerically unstable. Backward propagating in time of (2.10) gives

$$\Psi(\mathbf{r}, t) = \hat{U}(t, t+\Delta t)\Psi(\mathbf{r}, t+\Delta t) + \hat{\mathscr{O}}(\Delta t^3) \tag{2.11}$$

$$= \left(\hat{\mathbf{1}} + i\hat{H}(t+\Delta t)\Delta t - \frac{1}{2}\left(\hat{H}^2(t+\Delta t) + i\frac{\partial \hat{H}(t+\Delta t)}{\partial t}\right)\Delta t^2\right)\Psi(\mathbf{r}, t+\Delta t) + \hat{\mathscr{O}}(\Delta t^3). \tag{2.12}$$

Combining (2.10) and (2.12) for half a time step $\Delta t/2$ then leads to the identity

$$\hat{U}\left(t + \frac{\Delta t}{2}, t\right)\Psi(\mathbf{r}, t) = \hat{U}\left(t + \frac{\Delta t}{2}, t + \Delta t\right)\Psi(\mathbf{r}, t+\Delta t) + \hat{\mathscr{O}}(\Delta t^3), \tag{2.13}$$

or

$$\left(\hat{\mathbf{1}} - i\hat{H}(t)\frac{\Delta t}{2} - \frac{1}{2}\left(\hat{H}^2(t) + i\frac{\partial \hat{H}(t)}{\partial t}\right)\frac{\Delta t^2}{4}\right)\Psi(\mathbf{r}, t)$$
$$= \left(\hat{\mathbf{1}} + i\hat{H}(t+\Delta t)\frac{\Delta t}{2} - \frac{1}{2}\left(\hat{H}^2(t+\Delta t) + i\frac{\partial \hat{H}(t+\Delta t)}{\partial t}\right)\frac{\Delta t^2}{4}\right)\Psi(\mathbf{r}, t+\Delta t) + \hat{\mathscr{O}}(\Delta t^3). \tag{2.14}$$

With the operator expansion of the Hamiltonian,

$$\hat{H}(t+\Delta t) = \hat{H}(t) + \frac{\partial \hat{H}(t)}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 \hat{H}(t)}{\partial t^2}\Delta t^2 + \hat{\mathscr{O}}(\Delta t^3), \tag{2.15}$$

it follows

$$-\frac{1}{2}\left(\hat{H}^2(t+\Delta t) + i\frac{\partial \hat{H}(t+\Delta t)}{\partial t}\right)\frac{\Delta t^2}{4} = -\frac{1}{2}\left(\hat{H}^2(t) + i\frac{\partial \hat{H}(t)}{\partial t}\right)\frac{\Delta t^2}{4} + \hat{\mathscr{O}}(\Delta t^3), \tag{2.16}$$

and the Δt^2 terms in (2.14) cancel up to Δt^3:

$$\left(\hat{\mathbf{1}} - i\hat{H}(t)\frac{\Delta t}{2}\right)\Psi(\mathbf{r},t) = \left(\hat{\mathbf{1}} + i\hat{H}(t+\Delta t)\frac{\Delta t}{2}\right)\Psi(\mathbf{r},t+\Delta t) + \hat{\mathscr{O}}(\Delta t^3). \tag{2.17}$$

The Hamiltonian can be evaluated at the center point in time $t + \Delta t/2$ by the expansion

$$\hat{H}(t) = \hat{H}\left(t+\frac{\Delta t}{2}\right) - \frac{\partial}{\partial t}\hat{H}\left(t+\frac{\Delta t}{2}\right)\frac{\Delta t}{2} + \hat{\mathscr{O}}(\Delta t^2) \tag{2.18}$$

$$\hat{H}(t+\Delta t) = \hat{H}\left(t+\frac{\Delta t}{2}\right) + \frac{\partial}{\partial t}\hat{H}\left(t+\frac{\Delta t}{2}\right)\frac{\Delta t}{2} + \hat{\mathscr{O}}(\Delta t^2). \tag{2.19}$$

By inserting (2.18) and (2.19) into (2.17) the resulting terms in Δt^2 cancel exactly and one obtains:

$$\left(\hat{\mathbf{1}} - i\hat{H}\left(t+\frac{\Delta t}{2}\right)\frac{\Delta t}{2}\right)\Psi(\mathbf{r},t) = \left(\hat{\mathbf{1}} + i\hat{H}\left(t+\frac{\Delta t}{2}\right)\frac{\Delta t}{2}\right)\Psi(\mathbf{r},t+\Delta t) + \hat{\mathscr{O}}(\Delta t^3). \tag{2.20}$$

Therefore the propagated wave function can be written as

$$\Psi(\mathbf{r},t+\Delta t) = \left(\frac{\hat{\mathbf{1}} - i\hat{H}\frac{\Delta t}{2}}{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}\right)\Psi(\mathbf{r},t) + \hat{\mathscr{O}}(\Delta t^3), \tag{2.21}$$

where $\hat{H} = \hat{H}(t+\Delta t/2)$ is evaluated at the central point in time. This form of calculating the exponential in (2.5) is known as *Cayley-* or second order *Padé-* approximation. This propagator is unitary, since

$$\left(\frac{\hat{\mathbf{1}} - i\hat{H}\frac{\Delta t}{2}}{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}\right)^{\dagger}\left(\frac{\hat{\mathbf{1}} - i\hat{H}\frac{\Delta t}{2}}{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}\right) = \left(\frac{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}{\hat{\mathbf{1}} - i\hat{H}\frac{\Delta t}{2}}\right)\left(\frac{\hat{\mathbf{1}} - i\hat{H}\frac{\Delta t}{2}}{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}\right) = \hat{\mathbf{1}}, \tag{2.22}$$

has an overall error of Δt^3 and is numerically stable[Mitchell 80,Press 92,Thumm 02].

Popular alternatives (among others methods) to the direct lattice approach of the Cayley scheme (2.21), are Short Iterative Lanczos propagation[Kosloff 96] and Chebychev approximation[Boyd 00]. The first method starts from the Taylor series expansion of the exponential in (2.5):

$$\mathrm{e}^{-i\hat{H}\Delta t}\Psi(\mathbf{r},t) \approx \sum_{n=0}^{N} \frac{(-i\Delta t)^n}{n!}\hat{H}^n\Psi(\mathbf{r},t) \tag{2.23}$$

$$= \sum_{n=0}^{N} \frac{(-i\Delta t)^n}{n!}\psi_n(\mathbf{r}), \tag{2.24}$$

where the vectors $\{\psi_n\} = \{\Psi, \hat{H}\Psi, \hat{H}^2\Psi, \ldots\}$ form the basis of the *Krylov* space and are orthogonalized in a subsequent step. With the Hamiltonian matrix becoming tridiagonal in the reduced subspace of orthonormal Krylov vectors, the short-time propagated wave function can then be obtained from the diagonalization of the Hamiltonian. Already a few basis vectors are sufficient for a short-time propagation with a step size Δt considerably larger than for the Cayley approximation[Park 86,Kosloff 96].

The Chebychev method is based on the polynomial expansion of the exponential in (2.5) introduced by *H. Tal-Ezer* and *R. Kosloff*[Tal-Ezer 84,Kosloff 96]:

$$\mathrm{e}^{-i\hat{H}\Delta t} \approx \sum_{n=0}^{N} a_n T_n(-i\hat{H}\Delta t), \tag{2.25}$$

in terms of complex Chebychev polynomials T_n. The expansion coefficients are given by Bessel functions and the polynomials T_n can be obtained by a recurrence relation, i.e. $T_0(\hat{A}) = \hat{1}$, $T_1(\hat{A}) = \hat{A}$, and $T_n(\hat{A}) = 2T_{n-1}(\hat{A}) + T_{n-2}(\hat{A})$. With the polynomial approximation, the time step Δt can be chosen much larger than for the corresponding Cayley scheme at the same accuracy for a time-independent Hamiltonian.

2.3 Free Propagation

Now, the free propagation of a particle is considerered on a numerical grid. Since the potential $\hat{V}(\mathbf{r},t) = 0$, the time-dependence of the propagation is governed by the Hamiltonian $\hat{H} = \hat{T} = \hat{\mathbf{p}}^2/2$. Taking the Fourier transform of (2.1) yields the the Schrödinger

equation in momentum space,

$$i\frac{\partial}{\partial t}\tilde{\Psi}(\mathbf{p},t) = \frac{\mathbf{p}^2}{2}\tilde{\Psi}(\mathbf{p},t), \tag{2.26}$$

with the Fourier transformed wave function

$$\tilde{\Psi}(\mathbf{p},t) = (2\pi)^{-3/2}\int \Psi(\mathbf{r},t)\mathrm{e}^{-i\mathbf{p}\cdot\mathbf{r}}\mathrm{d}\mathbf{r}. \tag{2.27}$$

The time evolution of $\tilde{\Psi}(\mathbf{p},t)$ is given as in (2.5) by

$$\tilde{\Psi}(\mathbf{p},t+\Delta t) = \exp\left(-i\frac{\mathbf{p}^2}{2}\Delta t\right)\tilde{\Psi}(\mathbf{p},t). \tag{2.28}$$

The exponential in (2.28) is particularly easy to evaluate, since the momentum operator is diagonal $\hat{\mathbf{p}} = \mathbf{p}$, and the kinetic energy operator thus becomes a scalar. A common method for the free propagation on a numerical grid is to employ a Fast Fourier Transformation algorithm[Press 92,Feit 82] on the wave function $\Psi(\mathbf{r},t)$, multiply the wave function with the exponential in (2.28) for the propagation, and perform an inverse fast Fourier transformation to obtain the wave function $\Psi(\mathbf{r},t+\Delta t)$ in coordinate space again.

Another possibility to evaluate the kinetic energy operator $\hat{T}$, is to use a finite differencing scheme. Since the momentum operators commute, i.e. $[\hat{p}_i,\hat{p}_j] = 0$, the free propagator in (2.5) can be written as

$$\Psi(\mathbf{r},t+\Delta t) = \exp(-i\hat{T}_x\Delta t)\exp(-i\hat{T}_y\Delta t)\exp(-i\hat{T}_z\Delta t)\Psi(\mathbf{r},t). \tag{2.29}$$

It is therefore sufficient to only consider one coordinate direction for a one-dimensional subsystem. The numerical second derivative $\hat{T}_x = -\frac{1}{2}\partial^2/\partial x^2$ can then be expressed by a three-point formula (or 5-point formula for higher accuracy)[Abramowitz 72]:

$$\Psi''(x) = \frac{\Psi(x-\Delta x) - 2\Psi(x) + \Psi(x+\Delta x)}{\Delta x^2} + \hat{\mathscr{O}}(\Delta x^4) \tag{2.30}$$

$$\Psi''(x) = \frac{-\Psi(x-2\Delta x) + 16\Psi(x-\Delta x) - 30\Psi(x) + 16\Psi(x+\Delta x) - \Psi(x+2\Delta x)}{12\Delta x^2} + \hat{\mathscr{O}}(\Delta x^6). \tag{2.31}$$

Using the 3-point formula (2.30), the discretized $\hat{T}_x$ operator has the following tridiagonal matrix form:

$$(\hat{T}_x)_{ij} = -\frac{1}{2\Delta x^2}(\delta_{i,j-1} - 2\delta_{i,j} + \delta_{i,j+1}) \tag{2.32}$$

where $\delta_{i,j}$ is the Kronecker delta symbol. With the use of the Cayley scheme (2.21), and since the kinetic energy operators commute, each of the exponentials in (2.29) can be evaluated sequentially, thereby involving only the inversion of a one-dimensional operator in (2.21).

2.4 Crank-Nicholson Split Operator Propagation

In the previous section the special case of the free propagation with $\hat{V}(\mathbf{r}, t) = 0$ was considered. For a non-vanishing potential, it is desirable to divide the exponential in (2.5) similar to

$$\mathrm{e}^{-i(\hat{T}+\hat{V})\Delta t} \rightarrow \mathrm{e}^{-i\hat{T}\Delta t}\mathrm{e}^{-i\hat{V}\Delta t}, \tag{2.33}$$

such that the free propagation for the kinetic energy operator (2.29) can be applied, while the propagation of the potential, which is diagonal on the spatial lattice, reduces to a multiplication of the wave function with a scalar function in the exponential.

Since the differential operator for the kinetic energy $\hat{T}$ and the potential $\hat{V}$ do not commute, i.e. $[\hat{T}, \hat{V}] \neq 0$, the left and right hand side of (2.33) are not equal. The *Baker-Campbell-Hausdorff* formula[Reinsch 00] provides the correcting terms for non-commuting operators $\hat{A}$, $\hat{B}$ and $\hat{C}$,

$$\exp\left(\hat{A}\Delta t\right)\exp\left(\hat{B}\Delta t\right) = \exp\left((\hat{A}+\hat{B})\Delta t\right) + \exp\left(\frac{1}{2}[\hat{A},\hat{B}]\Delta t^2\right) + \hat{\mathscr{O}}(\Delta t^3), \tag{2.34}$$

for two operators, as well as for three operators:

$$\begin{aligned}&\exp\left(\hat{A}\Delta t\right)\exp\left(\hat{B}\Delta t\right)\exp\left(\hat{C}\Delta t\right)\\ &= \exp\left((\hat{A}+\hat{B}+\hat{C})\Delta t\right) + \exp\left(\frac{1}{2}\left([\hat{A},\hat{B}]+[\hat{B},\hat{C}]+[\hat{A},\hat{C}]\right)\Delta t^2\right) + \hat{\mathscr{O}}(\Delta t^3)\end{aligned} \tag{2.35}$$

The commutator term in (2.35) vanishes for $\hat{A} = \hat{C}$, therefore a symmetric splitting of kinetic energy and potential term $\hat{T} + \hat{V}$ gives:

$$\exp\left(-i(\hat{T}+\hat{V})\Delta t\right) = \exp\left(-i\hat{T}\frac{\Delta t}{2}\right)\exp\left(-i\hat{V}\Delta t\right)\exp\left(-i\hat{T}\frac{\Delta t}{2}\right) + \hat{\mathscr{O}}(\Delta t^3) \tag{2.36}$$

$$= \exp\left(-i\hat{V}\frac{\Delta t}{2}\right)\exp\left(-i\hat{T}\Delta t\right)\exp\left(-i\hat{V}\frac{\Delta t}{2}\right) + \hat{\mathscr{O}}(\Delta t^3), \tag{2.37}$$

and is accurate up to Δt^3. The splitting procedure allows either operator in the middle, but the distinction between (2.36) and (2.37) becomes apparent when evaluating these operators. While $\exp(-i\hat{V}\Delta t)$ is diagonal in coordinate space and therefore results in a multiplication with a scalar function, the action of the kinetic energy term involves the inversion of a matrix (2.21). Therefore, for one- two- or three-dimensional propagations, slightly different operator splittings are used in order to optimize for processing speed:

$$\text{1D}: \qquad \mathrm{e}^{-i\hat{H}\Delta t} = \mathrm{e}^{-i(\hat{T}_x+V)\Delta t} \tag{2.38}$$

$$\text{2D}: \qquad \mathrm{e}^{-i\hat{H}\Delta t} = \mathrm{e}^{-i\hat{T}_x\frac{\Delta t}{2}}\mathrm{e}^{-i(\hat{T}_y+V)\Delta t}\mathrm{e}^{-i\hat{T}_x\frac{\Delta t}{2}} \tag{2.39}$$

$$\text{3D}: \qquad \mathrm{e}^{-i\hat{H}\Delta t} = \mathrm{e}^{-iV\frac{\Delta t}{2}}\mathrm{e}^{-i\hat{T}_x\Delta t}\mathrm{e}^{-i\hat{T}_y\Delta t}\mathrm{e}^{-i\hat{T}_z\Delta t}\mathrm{e}^{-iV\frac{\Delta t}{2}} \tag{2.40}$$

Evidently, no operator splitting is necessary for the one-dimensional propagation, since the inversion of the operator in (2.21) requires the same effort if the diagonal potential term is added to the tridiagonal kinetic energy (2.32). With the same argument $\hat{T}_y + V$ can be combined for the two-dimensional propagation, such that $(\hat{T}_y + V)$ and $\hat{T}_x$ do not commute. For three (or more) dimensions, is is most efficient to place the potential propagation at both ends, and to take advantage of the commuting differential operators for the kinetic energy. A symmetric splitting of the three terms in $\hat{T} = \hat{T}_x + \hat{T}_y + \hat{T}_z$ is therefore not required.

In principle, the accuracy of the operator splitting can be improved by using higher order splitting forms[Forest 90,Yoshida 90,Suzuki 97]

$$\exp(\hat{A}+\hat{B}) = \prod_{n=1}^{N}\exp(c_n\hat{A})\exp(d_n\hat{B}), \tag{2.41}$$

with real coefficients c_i and d_i, but require increasingly more terms as well. Since the

Cayley scheme used in these calculations (2.21) has an error of Δt^3, no higher order operator splitting is necessary.

The combination of the Cayley scheme for the operator exponential (2.5), and the three-point finite differencing method for the spatial derivative (2.30) is known as *Crank-Nicholson* propagation[Press 92], with the split-operator generalization for N-dimensional problems. This method has been used throughout this work for propagation of nuclear and electronic wave functions on a uniform spatial grid in up to three dimensions.

2.5 Matrix Inversion

The Crank-Nicholson propagation scheme does involve finding the inverse of the operator

$$\hat{A} \equiv \left(\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2} \right). \tag{2.42}$$

Having $\hat{A}$ discretized on a numerical grid, this leads to the solutions of a set of linear equations in order to find $\hat{A}^{-1}$. For a general square matrix of rank N, the numerical effort grows with N^3, while more efficient algorithms exist for finding the inverse of sparse matrices. However, instead of solving for the inverse $\hat{A}^{-1}$, it is possible to find the propagated wave function directly by solving a set of linear equations: For this purpose one can define a wave function Φ such that

$$\Phi(t) \equiv 2\left(\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2} \right)^{-1} \Psi(t). \tag{2.43}$$

Φ can then be found by solving the linear equations

$$\frac{1}{2}\left(\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2} \right) \Phi = \Psi(t). \tag{2.44}$$

It follows from (2.21), that the Crank-Nicholson propagated wave function $\Psi(t+\Delta t)$ can then be obtained by

$$\Psi(t+\Delta t) = \left(\frac{\hat{\mathbf{1}} - i\hat{H}\frac{\Delta t}{2}}{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}\right)\Psi(t) \tag{2.45}$$

$$= \left(\frac{2\cdot\hat{\mathbf{1}} - \left(\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}\right)}{\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}}\right)\Psi(t) \tag{2.46}$$

$$= \left(2\left(\hat{\mathbf{1}} + i\hat{H}\frac{\Delta t}{2}\right)^{-1} - \hat{\mathbf{1}}\right)\Psi(t) \tag{2.47}$$

$$= \Phi - \Psi(t). \tag{2.48}$$

The short-time propagation is thereby reduced to the solution of a linear set of equations (2.44). For a one-dimensional system, the matrix $\hat{A}$ in (2.42), using a three-point finite difference scheme (2.32) for the differential operator $\hat{T}$, has a tridiagonal form for which highly optimized linear solver routines are commonly available[LAP] .

Furthermore, the constant off-diagonal elements in (2.32) for the kinetic energy operator, allow for additional optimization: only the diagonal elements of $\hat{A}$, containing the potential contribution, need to be stored, while the upper and lower diagonals are given by the constant $-i\Delta t/(4\Delta x^2)$. A modified tridiagonal matrix algorithm, based on simple Gaussian elimination, has been found to be the computationally most efficient. The solution of a three-diagonal linear set of equations scales linearly with the rank N of the matrix, such that the entire Crank-Nicholson scheme in one dimension also scales linearly with the number of grid points. With the use of the operator-splitting (2.39) and (2.40), the total numerical effort for the Split-Operator Finite-Difference Crank-Nicholson algorithm therefore increases linearly with the total number of grid points.

2.6 Propagation in Imaginary Time

The solution of the time-dependent Schrödinger equation (2.1) with a time-independent potential $\hat{V}(\mathbf{r})$, can be expanded in terms of eigenfunctions:

$$\Psi(\mathbf{r},t) = \sum_{n=0}^{\infty} a_n \psi_n(\mathbf{r}) \mathrm{e}^{-iE_n t}, \tag{2.49}$$

with the expansion coefficients a_n, eigenfunctions $\psi_n(\mathbf{r})$ and energies E_n. The eigenfunctions and energies satisfy the time-independent Schrödinger equation

$$\left(-\frac{\nabla^2}{2} + \hat{V}(\mathbf{r})\right) \psi_n(\mathbf{r}) = E_n \psi_n(\mathbf{r}). \tag{2.50}$$

A *Wick-rotation*, allowing for the time coordinate to become complex valued by introducing the substitution $t \rightarrow -i\tau$, is commonly used to transform the Schrödinger equation (2.1) into a heat-diffusion like equation[Esry 97,Lehtovaara 07]:

$$\frac{\partial}{\partial \tau}\Psi(\mathbf{r},\tau) = \left(\frac{\nabla^2}{2} - \hat{V}(\mathbf{r})\right) \Psi(\mathbf{r},\tau), \tag{2.51}$$

with an expansion in terms of eigenfunctions

$$\Psi(\mathbf{r},\tau) = \sum_{n=0}^{\infty} a_n \psi_n(\mathbf{r}) \mathrm{e}^{-E_n t}. \tag{2.52}$$

A comparison with (2.49) shows that the phase accumulation from the time propagation is now replaced by an exponential decay in (2.52). Therefore, a randomly chosen initial wave function $\Psi(\mathbf{r},\tau=0)$ with contributions from all eigenfunctions (all $a_n \neq 0$) is dominated by the ground state wave function ψ_0 for large propagation times τ:

$$\lim_{\tau\rightarrow\infty} \Psi(\mathbf{r},\tau) = a_0 \psi_0(\mathbf{r}) \mathrm{e}^{-E_0 \tau} \tag{2.53}$$

The imaginary time propagation is carried out with the Crank-Nicholson split-operator method from the previous sections, but with a complex time variable. For practical purposes, the wave function is renormalized after every n propagation steps and convergence is tested by

$$1 - \frac{|\langle \Psi(\tau) | \Psi(\tau + n\Delta\tau)\rangle|^2}{\|\Psi(\tau)\|^2 \|\Psi(\tau+n\Delta\tau)\|^2} < \epsilon, \tag{2.54}$$

with a sufficiently small ϵ. This scheme then yields the ground state wave function $\psi_0(\mathbf{r})$ with a ground state energy

$$E_0 = \frac{\langle\psi_0|\hat{H}|\psi_0\rangle}{\|\psi_0\|^2}. \tag{2.55}$$

Typically, $\epsilon = 10^{-15}$ and $n = 10$ has been used in the present work for one- and two-dimensinal propagations leading to convergence within a few hundred iterations. For the three-dimensional lattice the overlap (2.54) has been calculated every time step ($n = 1$), while a lesser accuracy with $\epsilon = 10^{-10}$ was chosen to reduce the numerical effort.

Excited states can be accessed by projecting out the lower energetic contributions in (2.49)[Sudiarta 07]. By removing the previously obtained ground state ψ_0 from the initial trial wave function:

$$\Psi(\mathbf{r}, \tau = 0) = \Psi^{(\text{trial})}(\mathbf{r}) - \left\langle \Psi^{(\text{trial})}(\mathbf{r})|\psi_0(\mathbf{r})\right\rangle \psi_0(\mathbf{r}). \tag{2.56}$$

Thus $a_0 = 0$ vanishes in (2.52) and the propagated wave function is dominated by ψ_1 in (2.53). Subsequent removal of an eigenstate contribution from the initial trial wave function and imaginary time propagation then leads to the convergence towards the next higher eigenstate of the system and therefore in principle all energies and states can be obtained by this iterative scheme. Practically, numerical errors limit the number of states that can be calculated with this method and although the ground state and lower laying eigenstates should have been removed from the initial wave function, the numerical noise after each propagation step can reintroduce contributions of the lower states. These states have to be projected out during the propagation and limit the number of states that can be calculated and the accuracy of the method.

For energetically closely spaced eigenstates, or degenerate states, the imaginary time propagation does not easily provide all possible states, while the method fails for a continuum of states. For the three-dimensional hydrogen atom for example, the resulting electronic wave function for the first excited state will contain contributions from the $2s$ and the $2p$ states and obtaining the corresponding wave functions requires additional symmetrization[Anis 06]. However, the ground state wave function and the energy of the first excited state

can be well reproduced with this method[Sudiarta 07].

Alternative methods for obtaining the eigenvalues and states of the Hamiltonian $\hat{H}$, such as the *ARPACK*[ARP] subroutine library, using an implicitly restarted Lanczos algorithm[Lehoucq 01] work well for reduced-dimensionality problems, but are inferior in scaling with respect to the total number of grid points compared to the linear scaling of the imaginary time propagation[Lehtovaara 07].

2.7 Coupled Channel Propagation

In the previous section, the propagation of a single-channel one-particle wave function has been carried out by the application of the split-operator Crank-Nicholson propagation scheme, and it can be used the same way for a series of uncoupled independent wave functions wave functions $\Psi_i(\mathbf{r},t)$ in different potential curves $V_i(\mathbf{r},t)$. Introducing a coupling between the Ψ_i that is local in $\mathbf{r}$ of the form

$$\Psi_i(\mathbf{r},t+\Delta t) = \hat{U}_i\left(\Delta t, \{\Psi_j(\mathbf{r},t\}\right)\Psi_i(\mathbf{r},t), \tag{2.57}$$

which depends on the value of the other wave functions Ψ_j at $\mathbf{r}$ and t allows for the transition between different channels. Such a scenario is given, for example, in Chapter 5 for the motion of a nuclear wave packet on different Born-Oppenheimer potential curves, coupled by the dipole matrix elements. In general the channel coupling can be extended for any number of channels, but is shown here only for up to three coupled wave functions. Similar calculations have been performed for two or more electronic states and are well described in the literature[De Raedt 87,Alvarellos 88,Balakrishnan 99]. The time-dependent Schrödinger equation for such a system can be written as

$$-i\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\Psi_0(t)\\ \Psi_1(t)\\ \Psi_2(t)\end{pmatrix} = (\hat{T}+\hat{V}+\hat{H}_c)\begin{pmatrix}\Psi_0(t)\\ \Psi_1(t)\\ \Psi_2(t)\end{pmatrix} \tag{2.58}$$

where the Hamiltonian $\hat{H}_0 \equiv \hat{T}+\hat{V}$ accounts for the propagation of the wave packet as in the previously considered cases, and $\hat{H}_c$ introduces the coupling between the wave functions of

different channels. Representing the kinetic energy operator $\hat{T}$ on a numerical grid with the three-point finite difference method, the total Hamiltonian on a grid then takes the form of a block-matrix, where the diagonal blocks are tridiagonal, and the off-diagonal blocks have diagonal structure arising from the coupling of the channels:

$$\hat{H} = \left(\begin{array}{c|c|c} \diagdown\!\diagdown & \diagdown & \diagdown \\ \hline \diagdown & \diagdown\!\diagdown & \diagdown \\ \hline \diagdown & \diagdown & \diagdown\!\diagdown \end{array} \right) . \tag{2.59}$$

Although the Hamiltonian is a sparse banded matrix, it does not have the simple tridiagonal form as in the previously considered uncoupled propagation, and therefore the solution of a linear equation system involving such a Hamiltonian would require significantly increased computing time.

The Hamiltonian can be divided into two parts $\hat{H} = \hat{H}_0 + \hat{H}_c$: the single channel tridiagonal Hamiltonian $\hat{H}_0$, and the coupling Hamiltonian $\hat{H}_c$. In the split-operator scheme, the short time step propagation can then be written as

$$\Psi(\mathbf{r}, t + \Delta t) = \mathrm{e}^{-i\hat{H}\Delta t}\Psi(\mathbf{r}, t) \tag{2.60}$$

$$= \mathrm{e}^{-i\hat{H}_c\frac{\Delta t}{2}}\mathrm{e}^{-i\hat{H}_0\Delta t}\mathrm{e}^{-i\hat{H}_c\frac{\Delta t}{2}}\Psi(\mathbf{r}, t) + \hat{\mathscr{O}}(\Delta t^3) \tag{2.61}$$

where $\Psi(\mathbf{r}, t) = \{\Psi_i(\mathbf{r}, t)\}$ is the vector of the coupled wave functions. Since the coupling $\hat{H}_c$ in the system Hamiltonian $\hat{H}$ does not commute with the differential operator for the kinetic energy $\hat{T}$, the split-operator form has been used in order to reduce the numerical error to the order of Δt^3. The action of $\hat{H}_0$ on the wave function vector is simply a series of independent Crank-Nicholson propagations on the components of the wave function vector. Therefore only the propagation with the coupling Hamiltonian needs to be considered here.

The Hamiltonian $\hat{H}_c$ can be rewritten as a sum of pairwise coupling matrices, i.e. $\hat{H}_c$ can be brought to a form of

$$\hat{H}_c = \left(\begin{array}{c|c|c} & \diagdown & \diagdown \\ \hline \diagdown & & \diagdown \\ \hline \diagdown & \diagdown & \end{array} \right) = \left(\begin{array}{c|c|c} & \diagdown & \\ \hline \diagdown & & \\ \hline & & \end{array} \right) + \left(\begin{array}{c|c|c} & & \\ \hline & & \diagdown \\ \hline & \diagdown & \end{array} \right) + \left(\begin{array}{c|c|c} & & \diagdown \\ \hline & & \\ \hline \diagdown & & \end{array} \right) \tag{2.62}$$

and use the split-operator technique again in (2.61). The coupling now reduces to 2×2 channel couplings $\hat{H}_c^{ij}$ acting only on the two channels i and j with $i \neq j$.

Using the series expansion of the operator exponential function involving $\hat{H}_c^{ij}$ gives

$$\mathrm{e}^{-i\hat{H}_c^{ij}\frac{\Delta t}{2}} = \sum_n (-i)^n \frac{\left(\hat{H}_c^{ij}\right)^n \Delta t^n}{n!} \tag{2.63}$$

$$\begin{aligned} =& 1 - \frac{\left(\hat{H}_c^{ij}\right)^2 \Delta t^2}{2} + \frac{\left(\hat{H}_c^{ij}\right)^4 \Delta t^4}{24} - \frac{\left(\hat{H}_c^{ij}\right)^6 \Delta t^6}{720} + \dots \\ & - i\hat{H}_c^{ij}\Delta t + i\frac{\left(\hat{H}_c^{ij}\right)^3 \Delta t^3}{6} - i\frac{\left(\hat{H}_c^{ij}\right)^5 \Delta t^5}{120} + \dots \end{aligned} \tag{2.64}$$

With this simplified coupling Hamiltonian, the expansion of the exponential operator (2.63) can be treated analytically.

It can be shown, that all the even powers of $\hat{H}_c^{ij}$ are diagonal matrices and the even powers are block matrices with the off-diagonal block being a diagonal matrix, i.e. for the channel couplings $(C)_{xy} = c(x)\delta_{xy}$ with

$$\hat{H}_c^{ij} = \left(\begin{array}{c|c} & \\ \hline & \end{array}\right) = \left(\begin{array}{c|c} & (C)_{xy} \\ \hline (C)_{xy} & \end{array}\right), \tag{2.65}$$

the powers of the coupling Hamiltonian are

$$(\hat{H}_c^{ij})^n = \begin{cases} \left(\begin{array}{c|c} ((C)_{xy})^n & \\ \hline & ((C)_{xy})^n \end{array}\right), & \text{for } n \text{ even;} \\ \left(\begin{array}{c|c} & ((C)_{xy})^n \\ \hline ((C)_{xy})^n & \end{array}\right), & \text{for } n \text{ odd.} \end{cases} \tag{2.66}$$

where $((C)_{xy})^n = c(x)^n\delta_{xy}$. Thus (2.63) can be rewritten for the pairwise coupling using trigonometric functions as

$$\mathrm{e}^{-i\hat{H}_c^{ij}\frac{\Delta t}{2}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \cos\left((C)_{xy}\frac{\Delta t}{2}\right) - i\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \sin\left((C)_{xy}\frac{\Delta t}{2}\right). \tag{2.67}$$

Thus the pairwise coupling between the two channels i and j introduces a pointwise rotation of the wave function between the channels.

It has been shown that the wave function couplings between three potential curves can be treated as a successive application of two-channel couplings involving the split-operator scheme, leading to a simple rotation of the wave function between the two channels. The

numerical error is of the order Δt^3 due to the operator splitting of the exponential and therefore of the same accuracy as the Crank-Nicholson propagator. This scheme can be applied to larger systems with more than three potential curves by an obvious extension of (2.62), although the number of pairwise couplings increases very rapidly for systems with many coupled potential curves, such that the direct numerical solution of (2.61) using $\hat{H}_c$ becomes more suitable. In this work, in particular in Chapter 5 and Chapter 6 only the coupling between two channels have been considered where an analytical form (2.67) exists.

2.8 Absorbing Boundaries

So far it was assumed, that the finite numerical grid fits the entire wave function and the wave function probability density on the grid boundary vanishes at all times. This is the case if one only considers bound states of the potential. For a finite numerical lattice where the wave function reaches the boundary one has to impose certain boundary condition, for example assuming that the wave function or the first derivative of the wave function vanish at the boundary. In both cases, a total reflection of the wave function occurs. This can also be as a consequence of the unitary time evolution operator for real valued potentials thus preserving the normalization of the wave function on the finite grid.

In order to avoid artificial reflections of the wave function, one can therefore choose a lattice large enough to contain the domain of the wave function during the propagation time, or to impose absorbing boundary conditions that reduce[Kosloff 86] or integral boundary condition that eliminate reflecions[Ermolaev 99]. If a negative imaginary potential $-iW(\mathbf{r})$ is added to the Hamiltonian $\hat{H}_0$ (2.2),

$$\hat{H}_{\text{tot}} = \hat{H}_0 + \hat{H}_{\text{abs}} = \left(-\frac{\nabla^2}{2} + \hat{V}(\mathbf{r}, t) - iW(\mathbf{r}) \right), \tag{2.68}$$

the propagated wave function, with the use of the split-operator scheme (2.36), can be written as

$$\Psi(\mathbf{r}, t + \Delta t) = \mathrm{e}^{-W(\mathbf{r})\frac{\Delta t}{2}} \mathrm{e}^{-i\hat{H}\Delta t} \mathrm{e}^{-W(\mathbf{r})\frac{\Delta t}{2}} \Psi(\mathbf{r}, t) + \hat{\mathscr{O}}(\Delta t^3). \tag{2.69}$$

The imaginary potential therefore reduces the probability density of the the wave function at every propagation step, where $W(\mathbf{r}) > 0$. It should be noted, that $\hat{H}_{\text{tot}}$ is not Hermitian, and the norm is therefore not conserved. For this reason, the normalization cannot be evaluated for testing the numerical rounding error of the propagation scheme when using absorbing boundaries.

Several other schemes have been proposed, including the use of a spatially expanding time-dependent lattice that follows the spreading unbound wave packets[Sidky 00,Zhao 02], the repetitive projection onto bound states or the propagation of a truncated set of Siegert states[Yoshida 99], complex rotation[Ho 83], and the splitting of the wave function in overlapping regions at the boundary[Keller 95,Chelkowski 96].

In this work absorbing boundary conditions are adopted, realized by a negative imaginary optical potential, or equivalently by a masking function, on a lattice stripe around the grid boundary[Kosloff 86,Hussain 00,Poirier 03,Muga 04]. While the height of the absorber defines the largest energy of a wave packet that can effectively be absorbed, following semi-classical WKB anlysis,the absorption length typically needs to be of the order of the de Broglie wavelength[Ge 98]. It has been shown, that the presence of an additional negative real potential can accelerate the wave packet and thus reduce the de Broglie wave length, such that the effective absorber length can be reduced[Hussain 00,Muga 04] over the use of a purely imaginary absorbing potential.

2.9 Summary

In this chapter a method to solve the time-dependent Schrödinger equation on a numerical grid was described: The Crank-Nicholson algorithm (Section 2.2) with an overall numerical error of $\hat{\mathscr{O}}(\Delta t^3, (\partial\hat{H}/\partial t)\Delta t^2)$ iteratively performs the time evolution of the wave function. The operator splitting procedure (Section 2.4) is used to divide a N-dimensional time propagation into a series of simple one-dimensional propagations by symmetric splitting of the Hamiltonian and thereby preserving the error of $\hat{\mathscr{O}}(\Delta t^3)$. The kinetic energy operator is

evaluated by using a three-point finite difference formula (Section 2.3) and thus discretizes on the numerical grid to a tridiagonal matrix. The propagation step is then carried out by solving a one-dimensional tridiagonal system of linear equations (Section 2.5). Since the series of one-dimensional propagations is decoupled, parallel execution has been implemented using the commonly available *OpenMP* standard[Ope] . Depending on the hardware used – single processor Intel Pentium vs. dual processor Intel Xeon, a theoretical performance gain of up to 50% is possible. However, due to the memory bandwidth limitation of the memory bus, the performance gain was found considerably reduced over the theoretical maximum.

It is obvious that a reduction of the number of grid points used to discretize the operators and the wave function will increase the computation speed. Scaled coordinates in order to avoid the use of wave function absorbers, or non-uniform grids could provide such improvements without sacrificing accuracy. The use of optimized complex absorption potentials rather than purely imaginary absorbers can also reduce the number of grid points needed for absorption. Alternative propagators, such as short-iterative Lanczos or Chebychev propagation both allow for a larger propagation time step, resulting in smaller numerical effort at equal accuracy for a time-independent Hamiltonian, but appear less suitable if the Hamiltonian changes "quickly" in time, as for the case of a moving projectile or an external laser field.

«Curiosity killed the cat, but for a while I was a suspect. »
Steven Wright

Chapter 3

Dichroism in Reduced-Dimensionality Laser-Assisted Collisions

3.1 Introduction

The study of charge exchange in ion-atom collisions dates back to the beginning of the last century, when *Henderson*[Henderson 23] experimentally discovered electron capture by α-particles passing through matter, and was pursued actively over many decades[Bransden 92]. More recently, the COLTRIMS technique[Ullrich 97,Dörner 00,Ullrich 03,Maharjan 06] has allowed for the investigation of the electron dynamics in ion-atom collisions with unprecedented resolution in energy and momentum of the interacting electrons and nuclei. Independently, the interaction of strong laser fields with atoms, ions or molecules has been addressed in a large number of experimental and theoretical investigations[Protopapas 97,Brabec 00,Joachain 00,Posthumus 04] over the past two decades. Even though the detailed investigation of laser-assisted heavy particle collisions may ultimately help in controlling quantum processes on a ultra-short time scale and in steering chemical reactions into specific reaction channels by adjusting laser parameters (intensity, wavelength, and pulse shape), the promising combination of the two research areas, laser-matter interactions and heavy particle collisions, has been the subject of only a few experiments with crossed heavy particle and laser beams, that are lim-

ited to the study of collisions between laser prepared reactants. For example, *Débarre* and *Cahuzac*[Dèbarre 86] observed laser-induced charge exchange between Sr^+ and Ba in a mixture of Strontium and Barium vapors using Nd-YAG lasers with relatively very low intensities up to 5×10^8 W/cm^2. *Grosser et al.*[Grosser 96] used a continuous beam of Sodium atoms, a pulsed beam of Krypton atoms, two pulsed laser beams in a pump – probe setup[Zewail 00], and crossed all beams in a small interaction volume to study collision reactions between laser-excited particles. They explained an oscillatory structure in the angular distribution of excited Na projectiles after collisions with Kr atoms in terms of optical molecular transitions in the transient NaKr complex.

So far, technical challenges in the generation of sufficiently long and intense laser pulses and the synchronization of laser pulses within the interaction time interval (typically not more than 10^{-13} s in slow ion-atom collisions) have prevented the more detailed experimental investigation of laser-assisted or laser-controlled charge-exchange reactions in heavy particle collisions, where the laser actively modifies the reaction dynamics. With the increasing availability of energetic lasers in atomic collision laboratories[Alnaser 03,Ullrich 03], it is expected that laser-induced effects in laser-assisted heavy particle collisions will become observable. High laser intensities, focussed to relatively large areas and long laser pulse durations will significantly improve the statistics in laser-assisted collision experiments and are expected to soon open the door towards more detailed experimental studies that may contribute substantially to our understanding of laser-controlled chemical reactions.

On the theoretical side, a variety of methods has been applied to the calculation of charge exchange and electron emission in laser-assisted heavy particle collisions. *Li et al.*[Li 97,Li 02] predicted, within lowest order perturbation theory in the electron nucleus interaction, that the dressing of atomic levels in an intense laser field leads to a significant modification of capture and ionization cross sections in fast proton-hydrogen collisions. *Voitkiv* and *Ullrich*[Voitkiv 01a] found, also within lowest order perturbation theory in the electron-projectile interaction, that a linearly polarized laser field can substantially influence the binary-

encounter electron emission process in fast collisions of α-particles with hydrogen atoms. Recent studies of *Voitkiv et al.* focus on the ionization for fast carbon C^{6+} ions with atomic hydrogen targets, assisted by a resonant $1s - 2p$ laser field[Voitkiv 06]. Close coupling calculations for heavy particle collisions, taking place in a strong laser pulse, were performed by *Madsen et al.*[Madsen 02] predicting a strong laser-induced modification of the $s \rightarrow p$ excitation probability in laser-assisted collisions between a protons and hydrogen H($1s$) or sodium Na($3s$) atoms in its ground state. Similar to the close coupling scheme above is the basis generator method[Kroneisen 99] developed and used by *Tom Kircher*, which includes a dynamically adapted wave function basis set and has been successful in the description of field free ion-atom collisions. He found a strong influence of the electron capture and loss probabilities in laser-assisted He^{2+} – H collisions on the laser wavelength and the initial phase of the laser electric field[Kirchner 02,Kirchner 04]. More recently, he has shown an enhanced electron capture by protons colliding with Argon and Neon target atoms[Kirchner 07] – an experimentally more feasible collision setup. In an attempt to obtain enhanced electron capture in α – hydrogen collisions at moderate laser intensities of 3.5×10^{12} W/cm^2, *Anis et al.* found an up to five-fold increase over the field-free collisions[Anis 06], and found good agreement between both, their grid based calculations and the electron-nuclear-dynamics expansion method[Deumens 94].

Lattice calculations on a three-dimensional cartesian grid for laser-assisted proton collisions with lithium atoms in ground and excited states by Pindzola *et al.*[Pindzola 03] show a significant modification of the charge-transfer process for moderate laser intensities of 10^{12} W/cm^2. Also within a three-dimensional lattice, *Lein* and *Rost*[Lein 03] applied a reduced dimensionality model, solved the Schrödinger equation on a two-dimensional cartesian grid, and predicted the generation of ultrahigh harmonics of the driving laser frequency in laser-assisted collisions between 2 keV protons and hydrogen atoms in a linearly polarized laser pulses of 16 optical cycles, a wavelength of 800 nm, and 10^{14} W/cm^2 intensity.

More work, both experimental and theoretical, has been done for laser-assisted electron

scattering, but even a structureless projectile constitutes a serious challenge to present theories[Mason 93,Joachain 00,Cionga 01]. The early theory of Kroll and Watson[Kroll 73] which only retained terms to first order in the photon frequency disagrees with the experimental results of *Wallbank* and *Holmes*[Wallbank 94]. This discrepancy between theory and experiment was traced to off-shell effects in the long-range polarization part of the electron-atom scattering potential[Jaroń 97]. More recently, laser-assisted electron impact ionization of Helium has been observed[Höhr 05] in a strong 4×10^{12} W/cm^2, 1064 nm laser field and clear deviations from the predictions of the first Born approximation were found. *Burke*, *Francken*, and *Joachain* introduced the non-perturbative R-matrix Floquet method[Joachain 00] which was subsequently applied to multiphoton ionization, higher harmonic generation, and laser-assisted electron atom collisions. Electron-ion collisions have recently attracted considerable interest as an integral part of the re-scattering process, in which non-sequential double ionization of an atom or molecule is explained in terms of electron impact ionization of one electron by the laser-driven and re-scattered other electron[Moshammer 02,Alnaser 03,Palaniyappan 05].

So far, laser-assisted ion-atom collisions in circularly polarized light have not been investigated. In this chapter a numerical solution of the Schrödinger-equation on a two-dimensional grid is performed and an optimal laser intensity is found for the observed dichroism in the electron capture. In the following Chapter 4, a full-dimensional simulation is carried out to provide total capture cross sections and a measure of the accuracy of the reduced dimensionality calculations. Within the reduced dimensionality model, the electronic motion and the rotating laser electric field are confined to the scattering plane. For projectiles (protons) on a classical straight-line trajectory, the dependence of the probabilities for electron loss, capture, and emission on the intensity and helicity of the laser electric field is studied. Even though experimental results are expected to differ slightly from the predictions of these two-dimensional calculations, it is expected that the results are of sufficient accuracy to provide useful estimates for optimized laser and collision parameters that most clearly display the effects of a laser pulse on the electronic dynamics in heavy particle collisions. The numerical

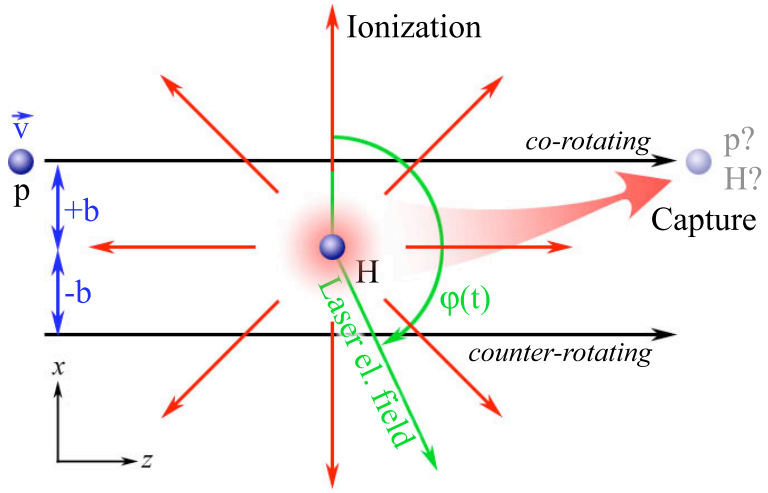

Figure 3.1: *Collision scenario for a proton on a straight-line trajectory with impact parameter b and velocity v colliding with an atomic hydrogen target. The rotating laser electric field breaks the azimuthal symmetry: For positive impact parameters, the projectile follows the rotating laser field (corotating case); for negative impact parameters, the projectile moves against the rotating electric field (counterrotating case).*

results show the strongest influence of the laser electric field on the capture probability at a laser intensity of 0.001 a.u. (3.5×10^{13} W/cm^2), i.e. when the laser electric force equals a few percent of the electrostatic Coulomb force exerted on the active electron by the target nucleus.

3.2 Potentials

Atomic units ($\hbar = m_e = e = 1$, see Appendix A) are used unless stated otherwise. For the impact energies considered in this chapter, the nucleus-nucleus interactions can be neglected and it is assumed that the projectile ion of mass m_P moves along a straight-line trajectory in z-direction,

$$\mathbf{R}(t) = b \cdot \mathbf{e}_x + v(t - t_0)\mathbf{e}_z, \tag{3.1}$$

which is characterized by the impact parameter b, the constant velocity v, and the time of closest approach t_0 (Fig. 3.1). It has been shown that the deflection angle of the projectile for even lower impact velocities is below 14° in α – hydrogen collisions and has a negligible

influence on the charge transfer[Anis 06].

Taking the location of the target atom as the coordinate origin, two-dimensional soft-core Coulomb potentials are employed with

$$V_T^{e^-} = -\frac{1}{\sqrt{x^2+z^2+a}} \tag{3.2}$$

and

$$V_P^{e^-}(t) = -\frac{1}{\sqrt{(x-b)^2+(z-v(t-t_0))^2+a}} \tag{3.3}$$

to represent the electronic interaction with the target and projectile nucleus, respectively. The "softening" parameter $a = 0.641$ regularizes the potentials at the location of the nuclei[Su 91,Feuerstein 03a] and is adjusted to reproduce the ground state binding energy of atomic hydrogen.

In dipole approximation, the interaction between the active electron and a monochromatic laser electric field of angular frequency ω,

$$E_x(t) = E_0(t)\cos(\omega(t-t_0)+\phi) \tag{3.4a}$$

$$E_z(t) = \epsilon E_0(t)\sin(\omega(t-t_0)+\phi), \tag{3.4b}$$

is given by the potential

$$V_L(x,z,t) = E_x(t)\cdot x + E_z(t)\cdot z \tag{3.5}$$

(Figure 3.1 and Figure 3.2). The parameter $\epsilon \in [-1,1]$ denotes the ellipticity of the laser light. The laser collision phase ϕ determines the direction of the laser electric field at the time of closest approach $t = t_0$ between the projectile and the target.

For the numerical applications in this work, a circularly polarized laser of positive helicity ($\epsilon = 1$) is assumed, corresponding to clockwise rotation of the laser electric field vector in the zx-plane (Figure 3.1). The wave vector of the incident laser light is directed into the collision plane in Figure 3.1. The envelope function $E_0(t)$ of the laser electric field turns the laser smoothly on during the time τ and then remains constant, once it has reached the

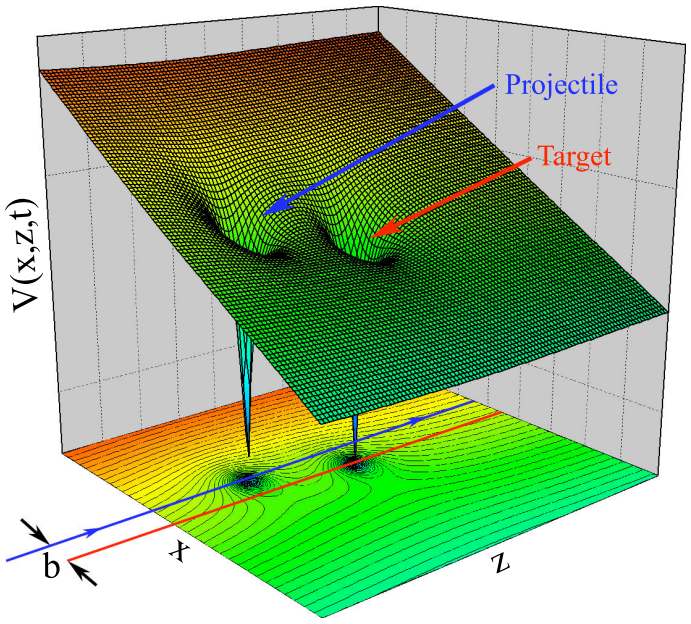

Figure 3.2: *Snapshot of the electronic potential. For negative helicity, the laser electric field causes a clockwise rotation of the inclined potential plane about the target while the projectile moves toward the right-rear end along a straight line.*

maximum field strength E_0,

$$E_0(t) = \begin{cases} E_0 \sin^2(\frac{\pi}{2} \cdot \frac{t}{\tau}) & 0 \leq t \leq \tau, \\ E_0 & t > \tau. \end{cases} \tag{3.6}$$

It is assumed that the laser ramping time τ is much shorter than t_0, i.e. $\tau \ll t_0$, such that the oscillating electric field is fully turned on before the collision. At the time of closest approach, the electric field is then given by

$$E_x = E_0 \cos\phi \qquad\qquad E_z = E_0 \sin\phi. \tag{3.7}$$

The sign of the projectile angular momentum relative to the target center of mass, $\mathbf{L} = \mathbf{R} \times m_P \mathbf{v}$, depends on the sign of the impact parameter. $\mathbf{L}$ can be either parallel or antiparallel to the laser helicity vector. In the first case the projectile moves in the same direction around the target as the laser electric field. This situation will be adressed as *corotating* scenario. Similarly, for the *counterrotating* scenario, when $\mathbf{L}$ and the helicity vector are antiparallel.

The collision process in the laser field is symmetrical with respect to the simultaneous change in sign of helicity and impact parameter. The computations can therefore be limited to a given helicity while allowing for both, positive and negative impact parameters. In all calculations a clockwise rotating laser electric field (positive helicity, i.e., $\epsilon = 1$) is assumed. For the coordinate system given in Figure 3.1 and for the laser light propagating into the plane of the figure, co (counter)-rotating collisions occur for positive (negative) impact parameters.

In order to suppress unphysical reflections of the electronic probability density at the boundaries of the rectangular numerical grid, absorbing boundaries are employed (see Section 2.8). For example, for absorption beyond x_0 in $+\mathbf{x}$-direction, this is achieved by adjusting the absorber strength s and absorber width x_a in the negative imaginary potential

$$V_A(x) = \begin{cases} -i\ s \cdot \left(\frac{x-x_0}{x_0}\right)^2 & x_0 < x < x_0 + x_a \\ 0 & \text{otherwise} \end{cases}, \tag{3.8}$$

so that the reflected probability flux becomes negligible. The net electronic potential to be used in wave function propagation is thus given

$$V(x,z,t) = V_T(x,z) + V_P(x,z,t) + V_L(x,z,t) + V_A(x,z) \tag{3.9}$$

where $V_A(x,z)$ models the absorption in all directions in obvious two-dimensional generalization of $V_A(x)$.

3.3 Wave Function Dynamics

The solution of the time-dependent Schrödinger equation $i\partial_t|\Psi(t)\rangle = \hat{H}(t)|\Psi(t)\rangle$ is formally given by the evolution of the initial wave function $\Psi(x,z,t=0)$,

$$\Psi(x,z,t) = \mathscr{T} \exp[-i \int_0^t dt' \hat{H}(x,z,t')]\ \Psi(x,z,0), \tag{3.10}$$

with the time-ordering operator $\mathscr{T}$ and the Hamiltonian

$$\hat{H}(t) = \hat{T}_x + \hat{T}_z + V(x,z,t). \tag{3.11}$$

$\hat{T}_x$ and $\hat{T}_z$ are the electronic kinetic energy operators. The numerical propagation of the Schrödinger equation (3.10) is carried out on a numerical grid using the unconditional stable Crank-Nicholson split-operator method (see Chapter 2). For a time step Δt the wave function (3.10) at time $t + \Delta t$ is recursively given in terms of $\Psi(t)$ by

$$\begin{aligned}\Psi(t+\Delta t) \approx & \exp[-i\,T_x\frac{\Delta t}{2}] \\ & \times \exp[-i\,(T_z + V(x,z,t))\Delta t] \\ & \times \exp[-i\,T_x\frac{\Delta t}{2}]\;\Psi(t)\end{aligned} \tag{3.12}$$

Equal grid spacings are chosen in x and z direction with $\Delta x = \Delta z = 0.2$. The grid covers 120 a.u. along the projectile trajectory (z-direction) and has a variable length in x-direction, depending on the impact parameter, given by $80 + |b|$. The absorbing boundaries are implemented with a width of $x_a = z_a = 20$ inside the grid boundaries and an absorption strength of $s = 0.01$. These absorber parameters produce converged results that do not differ from those obtained with altered absorbers of twice the absorption width or strength and show no signs of unphysical reflections at the grid edges.

The laser frequency is fixed in the near infra-red, $\omega = 0.043$, which corresponds to a wavelength of 1064 nm available from common $Nd:YAG$ lasers. After an initial ramping time $\tau = 450 = 10.9$ fs the electronic wave function is propagated in the laser field for 550 a.u. = 13.2 fs. A total propagation time in the laser field of $t_{max} = 1000 = 24$ fs leads to converged results for capture and ionization probabilities for all relevant values of b and ϕ and for laser intensities between $2.85\times10^{-5} = 1\times10^{12}$ W/cm^2 and $2.85\times10^{-3} = 1\times10^{14}$ W/cm^2. Time steps of $\Delta t = 0.1$ were found small enough to guarantee the long term accuracy of the propagation scheme.

At each time step the probability density is integrated over two square boxes of length 20 a.u., centered on the projectile ion and the target nucleus. For larger internuclear distances, these integrals $N_T(t)$ and $N_P(t)$ are interpreted as instantaneous electronic charge states on projectile and target, respectively. At the end of the numerical propagation, at

time $t = t_{max}$, they serve as approximations for the capture and ionization probabilities,

$$P^{\pm}_{cap}(b,\phi) = N_P(t = t_{max}) \tag{3.13a}$$

$$P^{\pm}_{ion}(b,\phi) = (1 - N_P(t = t_{max}) - N_T(t = t_{max})). \tag{3.13b}$$

The superscripts $\pm$ distinguish between co (+) and counterrotating (–) collisions. Since the laser collision phase ϕ is currently not observable nor experimentally controllable, we average over ϕ,

$$P^{\pm}_{cap}(b) = \frac{1}{2\pi}\int_0^{2\pi} d\phi\ P^{\pm}_{cap}(b,\phi) \tag{3.14a}$$

$$P^{\pm}_{ion}(b) = \frac{1}{2\pi}\int_0^{2\pi} d\phi\ P^{\pm}_{ion}(b,\phi). \tag{3.14b}$$

It is found for the capture and ionization probability, that the calculation of eight differnet different laser collision phases (between 0° and 315° with increments of 45°) is sufficient. Values for $P^{\pm}_{cap}(b,\phi)$ and $P^{\pm}_{ion}(b,\phi)$ at arbitrary values for ϕ are obtained by spline interpolation. Test calculations using 36 different phases with increments of 10° did not reveal relevant changes in the interpolated probabilities.

Finally, total cross-sections for capture and ionization are obtained by integrating over the impact parameter b,

$$\sigma^{\pm}_{cap} = 2\pi\int_0^{\infty} db\ b \cdot P^{\pm}_{cap}(b) \tag{3.15a}$$

$$\sigma^{\pm}_{ion} = 2\pi\int_0^{\infty} db\ b \cdot P^{\pm}_{ion}(b). \tag{3.15b}$$

It should be that $P^{\pm}_{cap}(b,\phi)$ and $P^{\pm}_{ion}(b,\phi)$ are calculated within a two-dimensional model, and that effects due to the reduced dimensionality are disregarded in the integration over b in $\sigma^{\pm}_{cap}$ and $\sigma^{\pm}_{ion}$. This issue will be addressed in the following Chapter 4, where a full-dimensional calculation is performed to obtain the total capture cross-section.

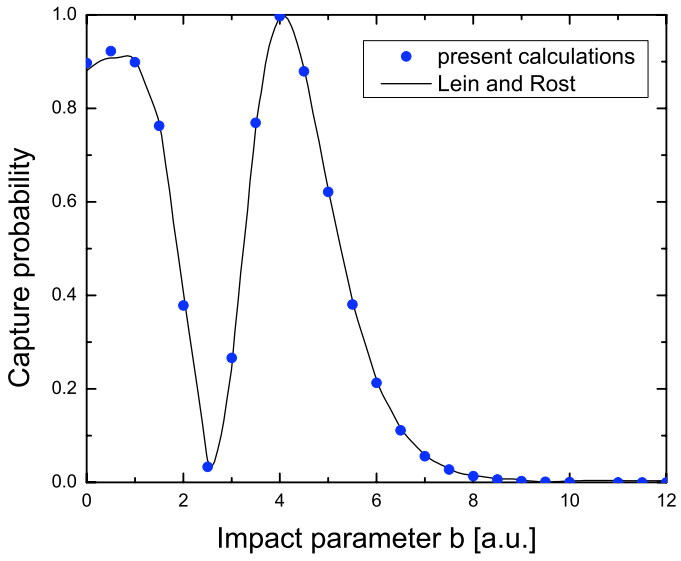

Figure 3.3: *Capture probability as a function of the impact parameter for field-free collisions of* 2 keV *protons with hydrogen atoms. Results from independent two-dimensional wave function propagation calculations: Lein and Rost*[Lein 03] *(solid curve), present results (dots).*

	Electron Capture Cross-Section		
Energy (keV)	σ_{cap}^{Th} (10^{-16} cm^2)	σ_{cap}^{Exp} (10^{-16} cm^2)	**Difference**
1	21.87	16.3±18%	34%
2	20.04	13.9±17%	44%

Table 3.1: *Comparison of the calculated (reduced dimensionality) total capture cross-sections for field-free collisions with the experiment of Gealy and Van Zyl*[Gealy 87].

3.4 Field Free Results

Reduced dimensionality numerical capture probabilities for field-free proton-hydrogen collisions have been published by *Lein* and *Rost*[Lein 03]. Their results are almost identical with the field-free capture probabilities presented here (Figure 3.3). Total capture cross-sections for collisions of $1-2$ keV protons with atomic hydrogen have been measured by *Gealy* and *Van Zyl*[Gealy 87]. For 2 keV incident kinetic energy, the presently calculated capture cross-section is 44% larger than the experimental value. For 1 keV protons it is 34% larger (Table 3.1).

The difference between the measured and calculated cross sections can be understood

in terms of a simple overlap argument. Compared to experiment or full-dimensionality calculations, the smaller phase space inherent in reduced-dimensionality calculations increases the wave function overlap between the interacting projectile and target, thus resulting in larger calculated cross sections (Table 3.1).

3.5 Circular Polarization

The presence of the laser radiation during the collision process results in an additional dependence of the electronic dynamics on the laser collision phase ϕ at the time of closest approach. The results presented here for a fixed impact parameter $b = \pm 4$ and laser intensity 5×10^{13} W/cm^2 for the capture probability as a function of ϕ show large amplitude oscillations and differ from the field-free results most strikingly for $\phi = 0°$ and 180° (Figure 3.4). They also display a strong dichroism effect, i.e., a substantial difference in the electron capture probability for positive and negative impact parameters, or, equivalently, for corotating as compared to counterrotating collisions.

In comparison to the phase-averaged results for the field-free case, it is found that the capture probabilities in both, co- and counterrotating collisions are considerably reduced. The ionization probabilities depend less sensitively on ϕ, and their phase averages (not shown in Fig. 3.4) differ much less for co- and counterrotating collisions (positive and negative impact parameters) than the phase-averaged capture probabilities. This tendency of a weak dichroism in the ionization probability extends to other impact parameters, as will be discussed below.

3.6 Electron Capture

Starting with the strong dichroism apparent in the capture probability at a fixed impact parameter b above, Figure 3.5 shows a contour plot of the electron capture probability as a function of both, the impact parameter and the laser collision phase ϕ for a laser intensity of $I = 5 \times 10^{13}$ W/cm^2. The electron capture probability shows maxima at impact parameters

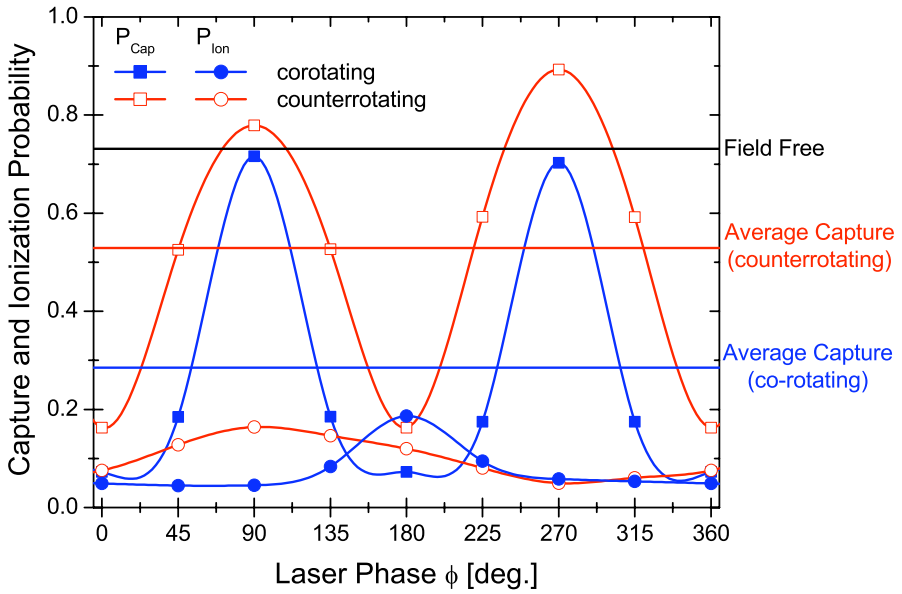

Figure 3.4: *Capture and ionization probability as a function of the laser collision phase ϕ at the time of closest approach between projectile and target for* 1 keV *p–H collisions. The impact parameter is* $b = \pm 4$ a.u. *and the laser intensity* 5×10^{13} W/cm^2. *Phase-averaged results for the capture probability differ significantly for co- and counterrotating laser-assisted collisions.*

$b = \pm 2$ and $b \approx \pm 4.0$. Similar structures appear for the field-free capture probability (Figure 3.3). They originate in the large wave function overlap of the corresponding target and projectile states near the point of closest approach.

With regard to the dependence on the phase of the rotating laser field, the capture probability shows a strong enhancement at $\phi = 90°$ and 270°, when the force exerted by the laser electric field on the electron at the time t_0 of closest approach is either anti-parallel or parallel to the direction of the projectile motion, respectively (see Figure 3.1 and note that force and electric field point in opposite directions). For these particular phases, the field-modified Coulomb potentials of the target and projectile are identical at t_0, and the internuclear axis is momentarily perpendicular to the laser electric field. This implies perfect energy level matching of field-dressed projectile and target states and explains the large resonant capture probabilities for $\phi = 90°$ and 270° in Figure 3.5.

To support this interpretation further, the following calculation considers the electron

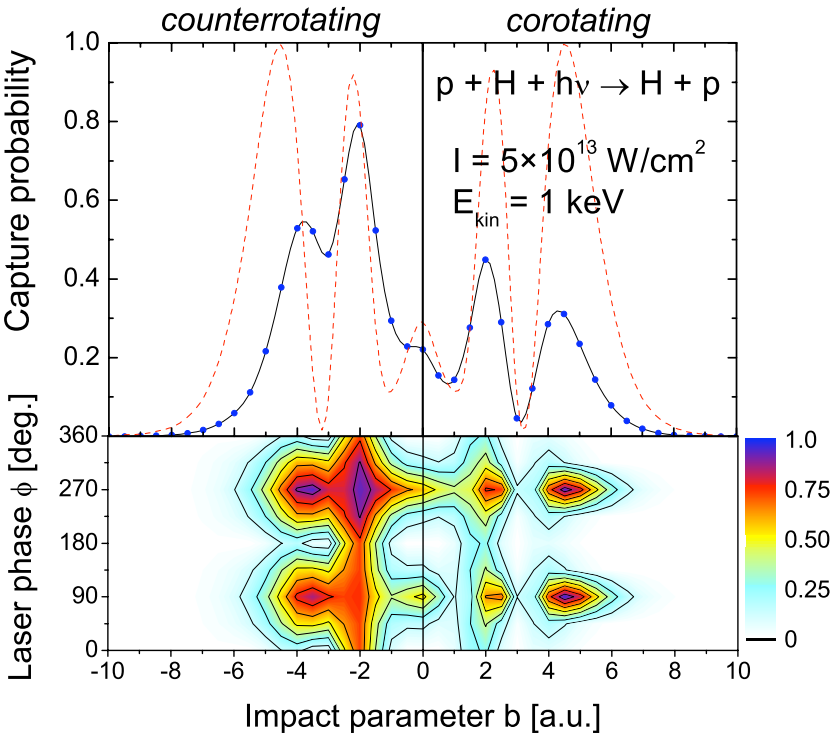

Figure 3.5: *Electron capture probability in laser-assisted* 1 keV *p - H collisions for a laser intensity of* 5×10^{13} W/cm^2. *The contour plots show the probabilities as a function of the impact parameter b and the laser collision phase* ϕ. *The probability difference between consecutive contour lines is* 0.125. *The top panel shows phase-averaged results. For comparison, the field-free capture probabilities are shown as well (dashed curve).*

capture probability for a *constant* electric field, corresponding in direction and magnitude to the laser electric field at time t_0, with otherwise identical parameters (Figure 3.6). In this simulation, ϕ parameterizes the direction of the stationary electric field. It is notable that this scenario is somewhat unrealistic, since a constant electric field would deflect the projectile ion and invalidate the assumption of a straight-line projectile trajectory. Interestingly, however, the dependence on the laser electric field direction of the capture probability in Figure 3.6 compares well with the ϕ dependence in Figure 3.5, thus adding credibility to the importance of the energy-level matching between projectile and target states at time t_0.

For the laser collision phases $\phi = 0°$ and 180° and positive impact parameters, the laser force on the electron at time t_0 points to the target or to the projectile, respectively (vice

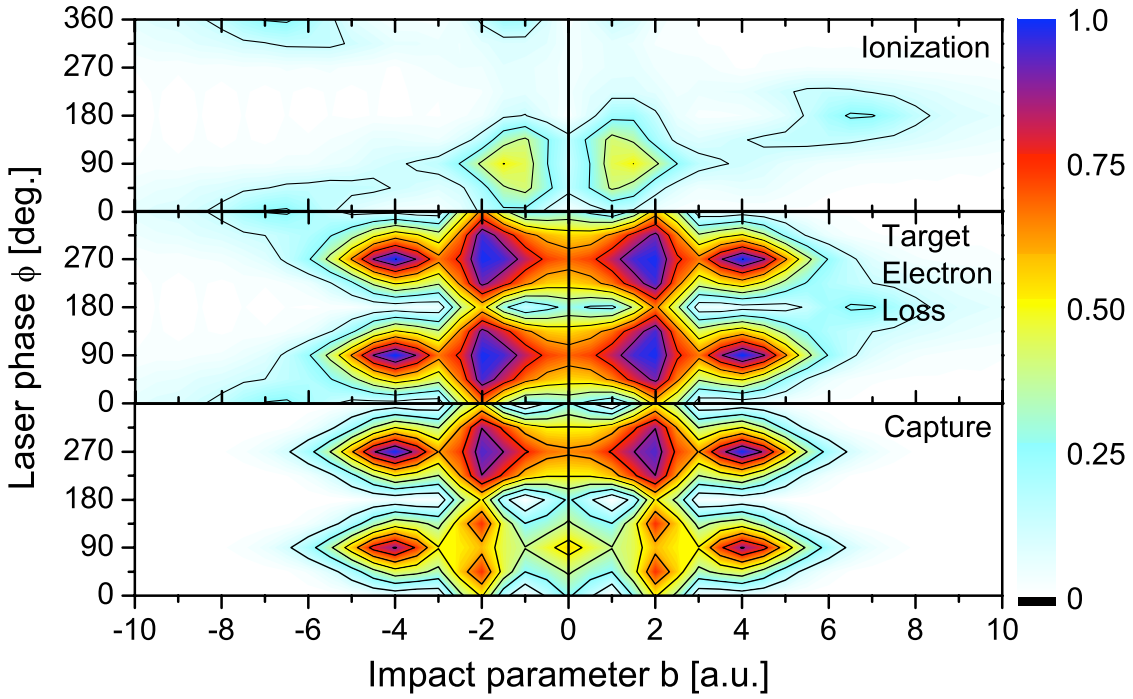

Figure 3.6: *Capture, ionization, and loss probabilities as a function of the impact parameter and the laser collision phase for the case of a static electric field, corresponding in the direction and magnitude to the laser electric field at the distance of closest approach.*

versa for negative impact parameters). The mismatch of the field-dressed hydrogen energy levels is largest at the time of closest approach, thus strongly suppressing electron capture in favor of enhanced ionization for $\phi = 180°$ at positive impact parameters and for $\phi = 0°$ at negative impact parameters (Fig. 3.8).

Compared to the laser collision phase of 270°, Figure 3.5 and Figure 3.6 show a slightly reduced capture probability at $\phi = 90°$, when the laser electric force on the electron at t_0 is anti-parallel to the projectile velocity. The target electron loss probability (Figure 3.7 and Figure 3.6) does not show this asymmetry, and the slightly larger capture probability for $\phi = 270°$ appears to be due to the "extra push" the electron receives by the laser force at t_0 in direction of the projectile motion. In contrast, for $\phi = 90°$, the electron is accelerated in the opposite direction by the laser force and a little more likely to ionize.

Overall, Figure 3.5 displays that the electron capture probability is strongly suppressed

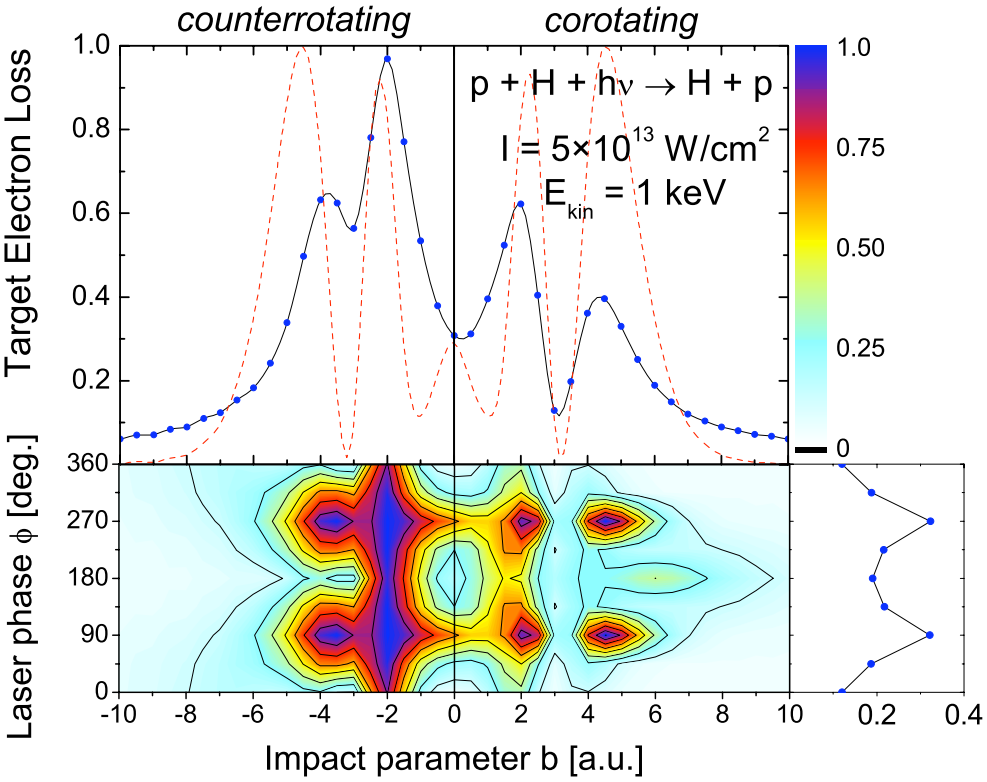

Figure 3.7: *Electron loss probability in laser-assisted* 1 keV *p - H collisions for a laser intensity of* 5×10^{13} W/cm^2. *The contour plots show the probabilities as a function of the impact parameter b and the laser collision phase* ϕ. *The probability difference between consecutive contour lines is* 0.125. *The top panel shows phase-averaged results together with the field-free electron loss probabilities (dashed curve), while the side panel displays the impact-parameter average as a function of the laser collision phase.*

for positive impact parameters, corresponding to corotating collisions, as compared with negative impact parameters for counterrotating collisions or the field-free electron capture. Also seen are much broader peaks at $\phi = 90°$ and 270° for the counterrotating case.

If the laser electric field is oriented perpendicular to the internuclear axis, both Coulomb potentials are identical and electron transfer is most likely. For corotating collisions, the relative orientation of laser electric field and internuclear axis changes much less rapidly during the collision time than for counterrotating collisions. At appropriate impact energies and impact parameters, this relative orientation is maintained throughout the projectile-target interaction for corotating collisions. During the interaction time, which is of the order

of one laser cycle, the projectile and target will then form a short-lived quasi-molecule.

In contrast, for counterrotating collisions, the angle between the rotating electric field of the laser light and the internuclear axis changes rapidly, irrespective of the value of ϕ. Level matching of projectile and target states occurs for a wide range of laser collision phases, but only for a small fraction of the interaction time. However, since the time scale of the electronic motion (1 a.u.) is about two orders of magnitude faster than a laser cycle (142.8 a.u.), the transient reflection symmetry of both Coulomb potentials still lasts long enough to enable noticeable electron transfer. In particular, at the chosen projectile velocity ($v = 0.22$) electron transfer to the projectile is relatively likely, while recapture by the target is suppressed by the rapidly increasing asymmetry between the two laser-modified Coulomb potentials. Thus, the electron capture is expected to depend most sensitively on the projectile velocity affecting the interaction time with the target.

For the given projectile speed, this explains the enhancement of capture in counterrotating collisions. For corotating collisions, the relative orientation of the laser electric field and the internuclear axis is maintained for approximately half a laser cycle, and the formation of a transient molecule decreases the probability for the electron to remain in a projectile state. In agreement with this explanation, a numerical test has shown that the capture probability in counterrotating collisions is reduced, and the difference between the co- and counterrotating electron capture becomes much less pronounced if the impact velocity is doubled ($E_{kin} = 4$ keV).

As mentioned earlier, the ϕ-dependence in laser-assisted capture cross-sections is difficult to resolve experimentally. Interestingly, however, the clear enhancement of the capture probability in counter- over corotating collisions remains after averaging over ϕ (top panel of Figure 3.5) and may be probed in angle-differential collision experiments, at appropriate projectile velocities.

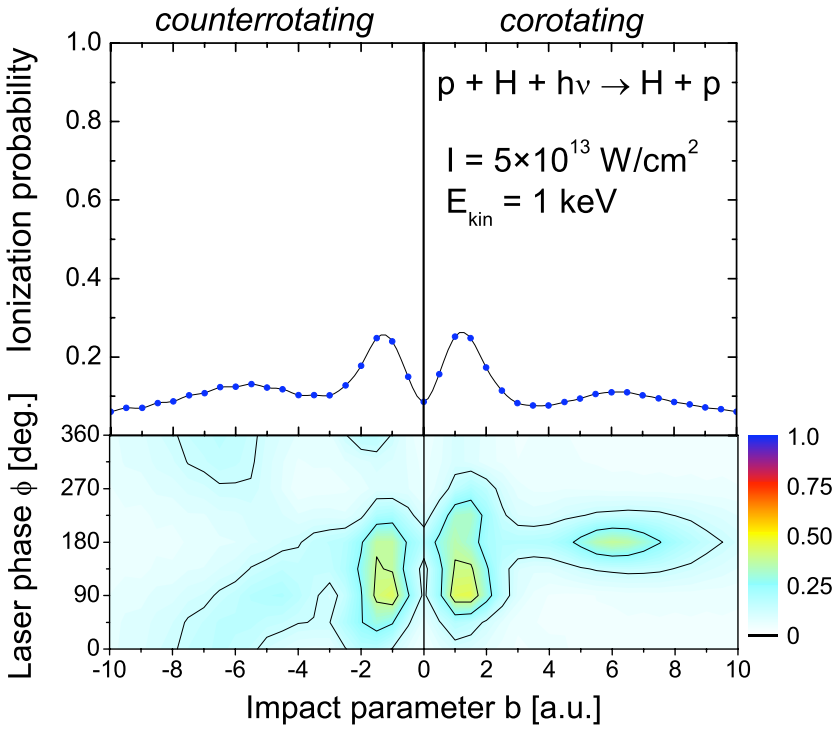

Figure 3.8: *Ionization probability in laser-assisted* 1 keV *p – H collisions for a laser intensity of* 5×10^{13} W/cm^2. *The contour plots show the probabilities as a function of the impact parameter b and the laser collision phase ϕ. The probability difference between consecutive contour lines is* 0.125. *The top panel shows phase-averaged results. No noticeable field-free ionization occurs on the scale of this figure.*

3.7 Ionization

The ionization probabilities in Figure 3.8 (and Figure 3.6 for the static field case) show a broad enhancement near $\phi = 90°$, when the laser electric force on the electron opposes the projectile motion, and for impact parameters around $b = \pm 1.5$. A less pronounced enhancement in the ionization probability occurs at $b \approx \pm 6$ (Figure 3.8).

For corotating collisions (positive b) and larger impact parameters, ionization is enhanced at a laser collision phase of 180°, while in the counterrotating case a much broader and weaker peek occurs near $\phi = 0°$. In both cases the laser force on the electron points towards the projectile at the time of closest approach (Figure 3.9). This signature of enhanced ionization in Figure 3.8 at $\phi = 0°$ and 180° corresponds to the known charge-resonance

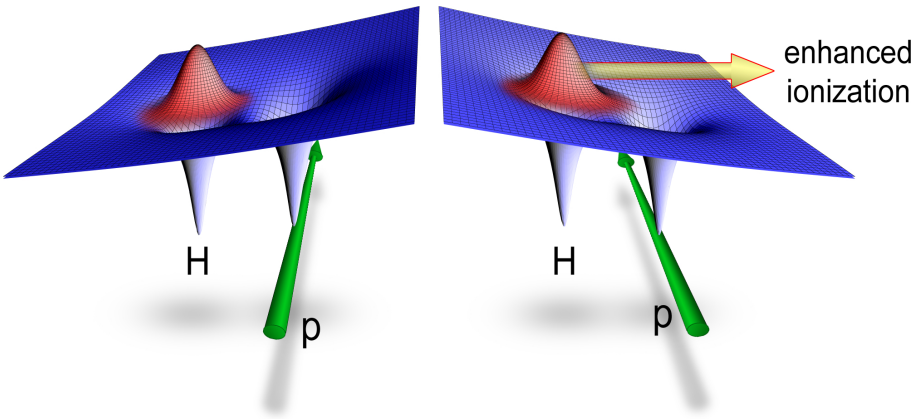

Figure 3.9: *Illustration of the enhanced ionization process occurring when the projectile Coulombic potential and the electric field of the laser exert a force on the electron simultaneously (right). No enhanced ionization appears when the Coulomb potential of the projectile does not lower the potential barrier in the ionization direction given by the laser electric field (left). Resonant enhancement has been predicted for an internuclear separation around* 6 a.u. *and* 10 a.u.[Zuo 95,Bandrauk 99]

enhanced ionization mechanism (CREI) during the fragmentation of diatomic molecules in strong laser fields at larger internuclear distances[Seideman 95,Zuo 95,Bandrauk 99].

The broadening of the ionization peak for counterrotating collisions (negative values of b) is identical to the corresponding feature in the capture probability discussed earlier. In the corotating scenario, while near the target, the projectile moves along with the laser electric field vector. The Coulomb and laser electric force then add to their maximal possible magnitude for a relatively long time. The time during which a maximal force is exerted on the electron is much smaller for the counterrotating case. Therefore, in counterrotating collisions, the CREI peak around $\phi = 0°$ is weaker and more diffuse than the CREI peak in corotating collisions at $\phi = 180°$ (Figure 3.8). Averaging over all laser collision phases ϕ removes the dichroism effect almost entirely (top panel in Figure 3.8), and, although the laser collision phase will not be experimentally controllable in the near future, the distinct CREI peak might still enable its measurement.

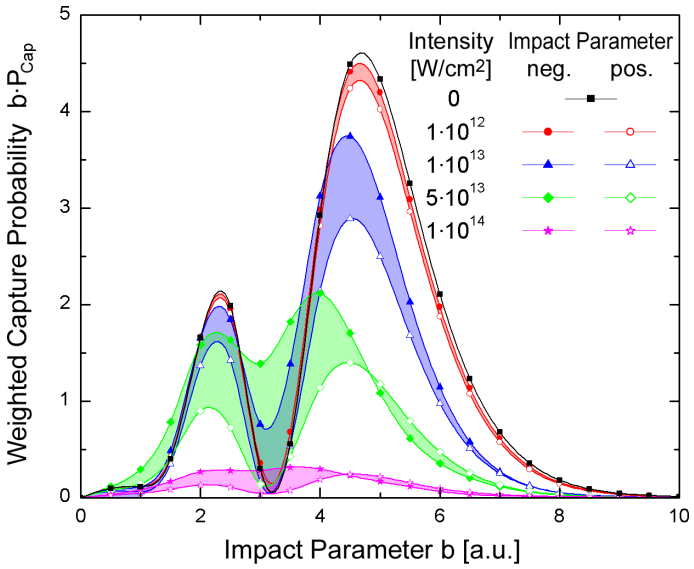

Figure 3.10: $b \cdot P_{cap}$, *averaged over the laser collision phase, at different laser intensities for corotating (positive impact parameter) and counterrotating (negative impact parameters) collisions.*

3.8 Laser Intensity Dependence

Shown in Figure 3.10 are the laser collision phase-averaged results for the weighted electron capture probability $b \cdot P_{cap}$ at different laser intensities for co- and counterrotating collisions. Noticeable differences between co- and counterrotation appear above the laser intensity of 2×10^{12} W/cm^2. The capture probability rapidly decreases above 1×10^{14} W/cm^2, when ionization begins to be the dominating process, but is smaller for all intensities than for field-free collisions. The relative difference

$$\Delta = \frac{|\sigma^+_{cap} - \sigma^-_{cap}|}{\sigma^+_{cap} + \sigma^-_{cap}|} \tag{3.16}$$

in the total electron capture cross section as a function of the laser intensity is shown in Figure 3.11. The difference in the capture cross section for co- and counterrotating collisions amounts to maximum of 40% at a laser intensity of 5×10^{13} W/cm^2 (Table 3.2). These differences should be considered as upper limits for the dichroism effect and it is expected that they will decrease slightly in full three-dimensional calculations, since an added degree of freedom no longer limits the electronic motion to the plane in which the laser field rotates.

Intensity (W/cm^2)	**Electron Capture Cross-Section**	
	corotating	**counterrotating**
0	78.08	78.08
$1 \cdot 10^{12}$	72.67	76.47
$5 \cdot 10^{12}$	57.85	70.41
$1 \cdot 10^{13}$	47.53	64.52
$5 \cdot 10^{13}$	26.18	43.74
$1 \cdot 10^{14}$	4.97	7.63

Table 3.2: *Comparison of the total capture cross-section for corotating and counterrotating collisions at different laser intensities.*

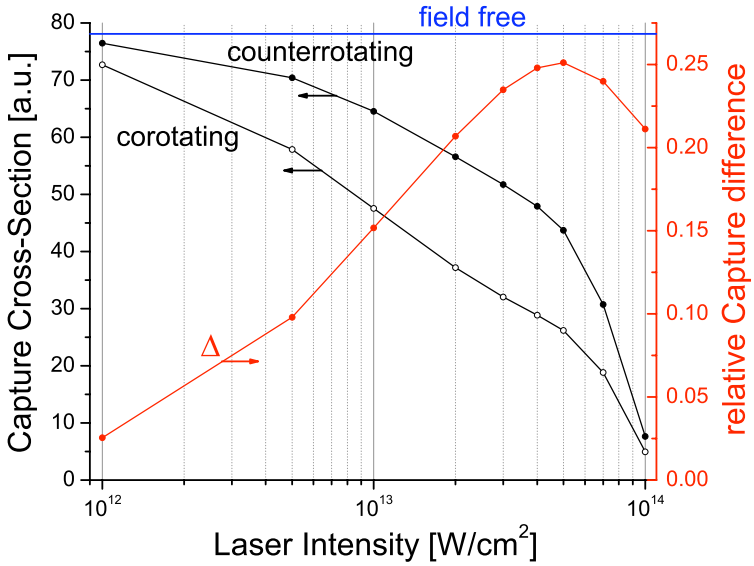

Figure 3.11: *Total electron capture cross-sections as a function of the laser intensity for co- and counterrotating collisions. Also shown is the relative difference* Δ *(3.16), which is largest at a laser intensity of* 5×10^{13} W/cm^2.

The results in Figure 3.11 relate to total cross sections only. Experimentally, scattering-angle differential cross sections (and $P(b)$) can be measured either directly, by detecting the projectile scattering angle, or indirectly, by observing the recoil direction of the target. For scattering angles that correspond to impact parameters with the largest circular dichroism in $P(b)$, the dichroism effect is more pronounced in differential cross sections than in total cross sections (cf. Fig. 3.5).

«An expert is a person who has made all the mistakes that can be made in a very narrow field. »
Niels Bohr

Chapter 4

Capture and Ionization in Laser-Assisted Proton – Hydrogen Collisions

4.1 Motivation

Driven by laser systems with increasing intensities and improving control and refined techniques for the momentum-resolved detection of ions, atoms, and photo-emitted electrons, interactions of intense laser fields with atoms and molecules continue to attract attention[Protopapas 97,Brabec 00,Joachain 00,Posthumus 04]. Ever since the invention of the laser, the challenge to efficiently control chemical reaction pathways with laser light has remained attractive for both fundamental and applied reasons. Electron emission and transfer processes have been studied in ion-atom collisions for many decades and have significantly contributed to our understanding of electronic dynamics in complex, three- (or more) body Coulomb systems. However, the theoretical consideration of laser-assisted ion-atom collisions, in which charge transfer and ionization processes of an ordinary collision reaction are modified due to the presence of an intense external electromagnetic field, has emerged only very recently (see[Voitkiv 01b,Madsen 02,Li 02,Pindzola 03,Lein 03,Niederhausen 06,Anis 06,Kirchner 07] and references therein) as a new line of research that promises to combine the benefits of atomic collision stud-

ies (e.g., the improved understanding of chemical reactions) and laser physics (control). It has been shown theoretically, that the proper choice of the laser parameters leads to significantly enhanced electron capture in asymmetric collisions[Kirchner 04,Anis 06,Kirchner 07], the modification of ionization probabilities[Li 02], the ponderomotively shifted emission of binary-encounter electrons in fast collisions[Voitkiv 01b], and the generation of ultra-high harmonics of the fundamental laser frequency[Lein 03].

While laser-assisted electron-atom collisions[Mason 93,Ehlotzky 98,Joachain 00,Höhr 05] have been investigated theoretically and experimentally for more than one decade, so far no experiments were carried out for strong-field-assisted ion-atom encounters. This is due to the challenging task of intersecting – in space and time – a strong laser pulse with the ion-atom interaction, while discriminating the laser-assisted events from the field-free collisions. However, recently improved momentum spectroscopy methods[Dörner 00,Ullrich 03] in combination with currently being developed pico-pulsed ion beams[Carnes 06] and new experiments with ns ion beams[Feuerstein 06] may soon provide experimental data. Theoretically, laser-assisted collisions have been described within basis-set expansions (atomic two-state approximations[Ferrante 81], and adaptive basis generator calculations[Kirchner 04]), perturbative calculations[Li 02], and grid-based methods[Pindzola 03,Lein 03,Niederhausen 04b,Niederhausen 06,Anis 06].

4.2 Theoretical Method

In the present chapter the laser-assisted collision between a 1.21 keV proton and an atomic hydrogen target, exposed to a circularly polarized laser field of 5×10^{13} W/cm^2 intensity (Figure 4.1) is investigated in full dimensionality. A brief summary of the theory in Chapter 3 is presented below with the focus on the differences with respect to the previous reduced dimensionality model. The presence of the laser breaks the cylindrical symmetry of the $p + H(1s)$ collisions and allows one to distinguish the following three special cases:

- *corotating collisions*, where the impact angle α (see Figure 4.1) is zero, and the angular momentum of the projectile $\mathbf{L} = \mathbf{R} \times m_p\mathbf{v}$, is parallel to the helicity vector $\mathbf{h}$ of the

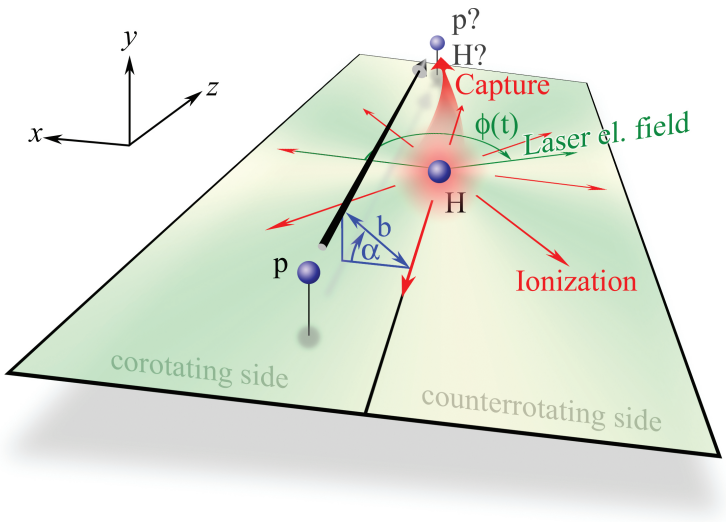

Figure 4.1: *Scenario of the collision between a moving proton and a hydrogen atom, characterized by the impact parameter b and the angle α with the xz-plane in which the laser electric field rotates. The overall laser collision phase $\phi = \phi(t_0)$ is the direction of the electric field at the time $t = t_0$ of closest approach.*

laser (in other words, the projectile rotation around the target nucleus follows the same direction as the rotating laser-electric field),

- *counterrotating collisions* with $\alpha = 180°$ where $\mathbf{L}$ and $\mathbf{h}$ are anti-parallel (referring to a rotation of the projectile around the target in opposite direction as the rotation of the laser-electric field), and in contrast to Chapter 3,

- *off-plane collisions* with $\alpha = \pm 90°$ where the internuclear axis is perpendicular to the plane of the laser-electric field at the time of closest approach.

In the length form and in dipole approximation for the electric field, the Hamiltonian for the collision system reads

$$\hat{H}(t) = -\frac{1}{2}\nabla^2 - \frac{1}{r} - \frac{1}{|\mathbf{r} - \mathbf{R}(t)|} + \mathbf{r} \cdot \mathbf{E}(t), \tag{4.1}$$

where $\mathbf{R}(t) = (b\cos\alpha, b\sin\alpha, vt)$ denotes the projectile trajectory in straight-line approximation with projectile velocity v, impact parameter b, and impact angle α relative to the

xz-plane. Atomic units (see Appendix A) are used throughout this chapter unless stated otherwise.

The circularly polarized electric field of the laser radiation is confined to the xz-plane,

$$\mathbf{E}(t) = E_0(t)\left(\cos\left[\omega t + \phi\right], 0, \sin\left[\omega t + \phi\right]\right). \tag{4.2}$$

$E_0(t)$ is the envelope function and ϕ the laser collision phase at the time ($t = 0$) of closest approach between the projectile and the target. It is assumed that the laser pulse has been ramped adiabatically to the constant electric field amplitude E_0 long before the collision takes place.

The time-dependent Schrödinger equation is solved on a three-dimensional lattice, using the Crank-Nicholson propagation scheme (see Chapter 2). Equal grid spacings are used in all three coordinate directions $\Delta x = \Delta y = \Delta z = 0.25$ for a numerical grid that covers 80 a.u. in z-direction and a length of at least 60 a.u. in x and y-direction, adjusted depending on the value of the impact parameter (vector) b. An absorbing optical potential (see Section 2.8) is employed to suppress non-physical effects due to reflections of the electronic wave function at the grid boundaries, identical as in the previous Section 3.2. A small "soft-core" to the nuclear potential is introduced that, in contrast to the reduced dimensionality model in Chapter 3, is not used for adjusting the binding energy but to avoid the Coulomb-singularities at the nuclei. This is implemented by truncating the electronic potential below a certain threshold, $-1/(2\Delta x)$ where Δx is the uniform grid spacing, thus affecting at most one single grid point of the ion potential. The approach is different from the softening-parameter a introduced in (3.2) and (3.3), as the exact analytical value for the three-dimensional calculation is of course $a = 0$ in order to reproduce the binding energy, but is certainly easier to use than avoiding the Coulomb singularity by adjusting the trajectory to move between grid points[Pindzola 03,Anis 06]. The resulting binding energy is found within 5% above the correct value $E_0 = -0.5$ a.u. for the grid used in the calculations of this chapter.

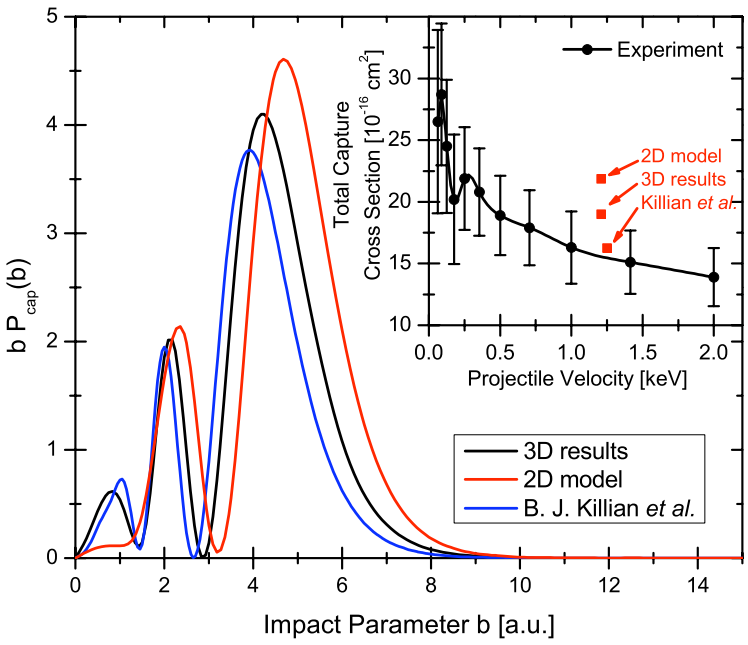

Figure 4.2: *Comparison of the weighted electron capture probabilities* $b \cdot P_{\mathrm{cap}}$ *without the presence of a laser field as a function of the impact parameter* b *for the current three-dimensional calculation*[Niederhausen 06] *(black line), the simplified two-dimensional model*[Niederhausen 04b] *(red line) and the END calculation by Killian et al.*[Killian 04] *(blue line). Shown in the inset is the measured total electron capture cross section by Gealy and Van Zyl*[Gealy 87] *as a function of the projectile energy as well as the theoretical prediction of the three methods above (red dots).*

Source	σ_{tot} $(10^{-15}$ $cm^2)$
3D grid[Niederhausen 06]	19.0
2D model[Niederhausen 04b]	21.9
END method[Killian 04]	16.3
experiment[Gealy 87]	15.6±18%

Table 4.1: *Comparison of the total capture cross-sections for field-free collisions. Results for 3D and 2D grid calculations, basis-set (END) calculations, and experimental data.*

4.3 Field Free Collisions

The numerical scheme was tested for field-free collisions and total (angle-integrated) capture cross sections,

$$\sigma_{\mathrm{cap}}^{\mathrm{tot}} = \int_0^\infty \mathrm{d}b\, b P_{cap}(b), \tag{4.3}$$

were found in agreement with experimental data[Gealy 87]. In comparison with the electron – nuclear dynamics approach (END) by *Killian et al.*[Killian 04], the present b-dependent capture probabilities are in fair agreement but tend to be shifted towards larger impact parameters (Figure 4.2). The full three-dimensional calculations lead to about 15% smaller total capture cross sections than previous two-dimensional reduced dimensionality results[Niederhausen 04b] (Chapter 3) and therefore constitute a significant quantitative improvement (Table 4.1).

4.4 Field Assisted Collisions

For the laser-assisted calculations, the wavelength 1064 nm of a Nd:YAG laser is used, corresponding to the frequency $\omega = 0.044$. For a typical impact parameter of $b = 1$, Figure 4.3 shows the dependence of the electron capture probability on the laser collision phase ϕ at the time of closest approach. Maxima appear at $\phi = 90°$ and $\phi = 270°$, when the force of the laser electric field at the time of closest approach is either parallel or antiparallel to the motion of the projectile. This agrees with an energy matching argument found previously in Section 3.6: Resonant capture occurs predominantly, when the electronic energies of the laser-dressed projectile and target states are identical, i.e., when the laser-electric field vector is oriented perpendicularly to the internuclear axis. Clearly, the time interval during which level matching near the point of closest approach can approximately be maintained, allowing for resonant transitions between the symmetric nuclei potentials, is larger for corotating than for counterrotating collisions. For the projectile speed and impact parameter in Figure 4.3, the collision time is of the order of half a laser cycle (≈71.4 a.u.), while the electronic time scale is about two orders of magnitude faster (≈1 a.u.). This enables the transient formation of a molecular state, even for counterrotating collisions where the matching condition only holds for a small fraction of one laser cycle. In contrast, for corotating collisions, level matching is maintained much longer and for approximately one half of a laser cycle. For the given parameters this favors recapture by the target and explains that the capture cross sections in Figure 4.3 are smaller for corotating collisions.

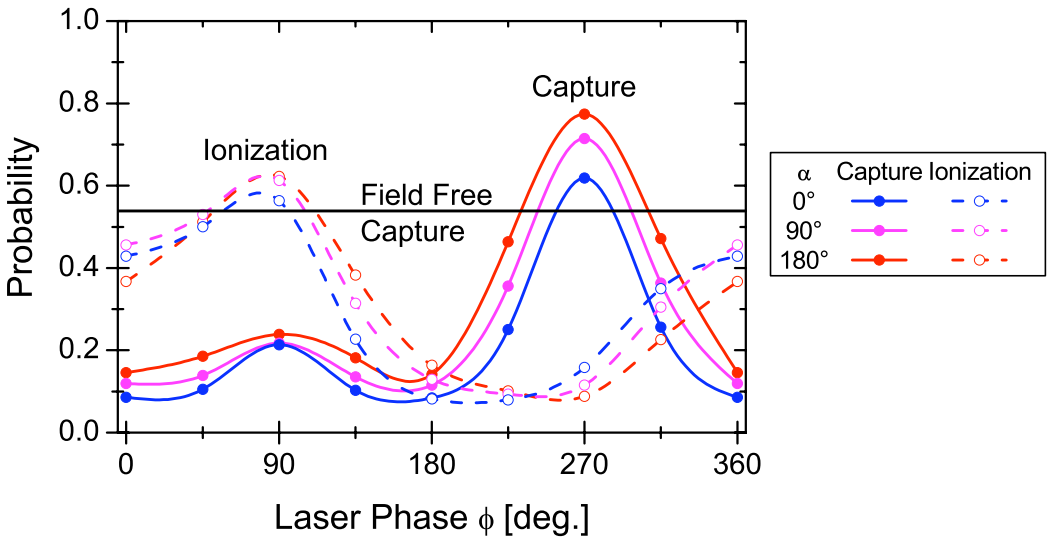

Figure 4.3: *Laser collision phase dependence of the electron capture and ionization probability at fixed impact parameter $b = 1$ for three different impact angles α. Also shown is the field-free capture probability. The field-free ionization probability is negligible.*

The phase dependence of the electron capture probability near $\phi = 90°$ is broadened for counterrotating as compared with off-plane collisions at $\alpha = \pm 90°$. For corotating collisions, in contrast, the capture probability changes faster near $\phi = 90°$ than for $\alpha = \pm 90°$ off-plane collisions.

For $b = 1$, the ionization probability has a maximum at $\phi = 90°$ (Figure 4.3). At this laser collision phase, the laser electric force $\mathbf{F}_L(t = 0)$ on the electron at the time of closest approach is anti-parallel to the projectile velocity $\mathbf{v}$, thereby reducing the chance for electron capture and recapture. In contrast, near $\phi = 270°$, $\mathbf{F}_L(t \approx 0)$ is nearly parallel to $\mathbf{v}$ and facilitates capture. This effect explains relatively small ionization and large capture probabilities near $\phi = 270°$.

4.5 Electron Capture

In Figure 4.4 the dependence of the electron capture probability on both the impact parameter b and the laser collision phase ϕ is shown for the five cases: corotating with $\alpha = 0°$ (Figure 4.4(a)) and $\alpha = \pm 45°$ (Figure 4.4(b)), off-plane with $\alpha = \pm 90°$ (Figure 4.4(c)) and

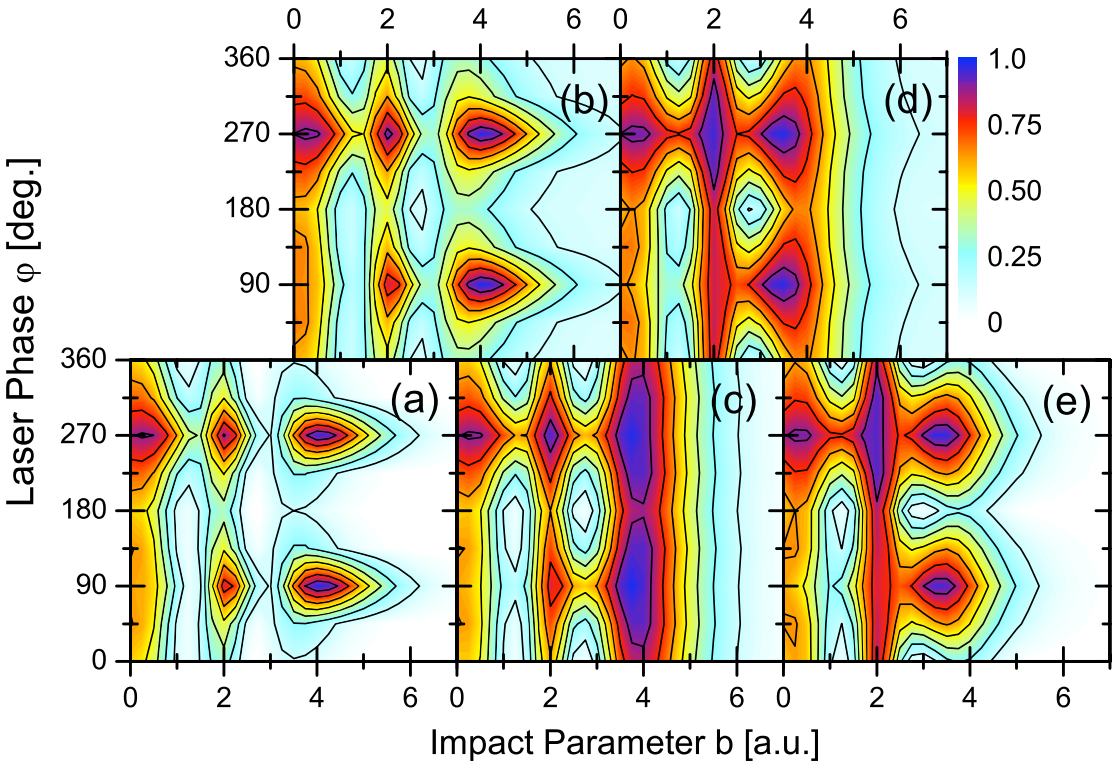

Figure 4.4: *Contour plots of the electron capture probability as a function of the impact parameter and laser collision phase for different tilt angles α between the laser and the collision plane. (a) corotating collision with $\alpha = 0°$, (b) corotating collision with $\alpha = \pm 45°$, (c) off-plane with $\alpha = \pm 90°$, (d) counterrotating with $\alpha = \pm 135°$ and (e) counterrotating with $\alpha = 180°$.*

counterrotating collisions with $\alpha = \pm 135°$ (Figure 4.4(d)) and $\alpha = 180°$ (Figure 4.4(e)). The graphs exhibit the features discussed above for fixed impact parameters, namely enhanced capture at laser collision phases of $\phi = 90°$ and $\phi = 270°$. The capture probability at $\phi = 270°$ is slightly larger than at $\phi = 90°$ due to stronger ionization in the later case. Furthermore, the capture probability is enhanced for the counterrotating collision as compared to the corotating scenario. While for $b \leq 3$ the off-plane capture probabilities with $\alpha \pm 90$ lie in general somewhere in between the co- and counterrotating results, the electron capture probabilities at larger distances ($b > 3$) become closer to the field-free values, and the laser collision phase dependence nearly disappears (Figure 4.4(c)).

Since the dependence on the laser collision phase ϕ cannot yet be controlled experimen-

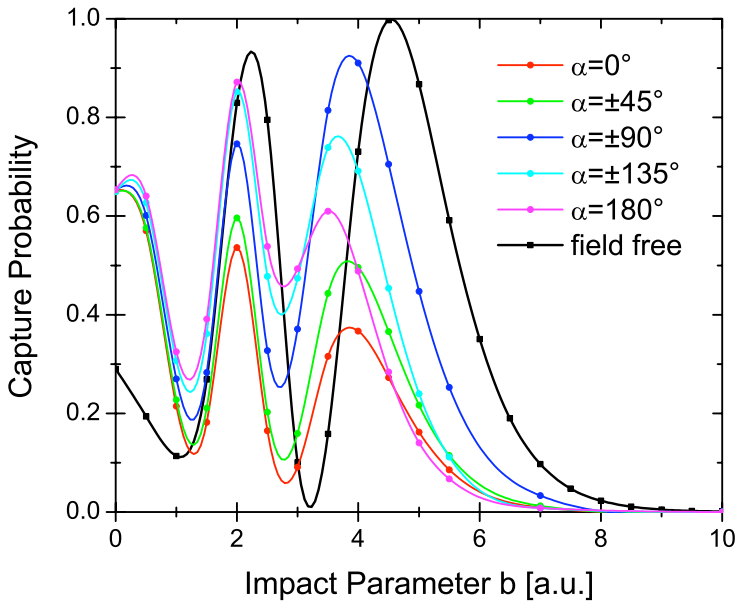

Figure 4.5: *Phase-averaged capture probabilities as a function of the impact parameter b for corotating ($\alpha = 0°$ and $\pm 45°$), off-plane ($\alpha = \pm 90°$) and counterrotating collisions ($\alpha = \pm 135°$ and $180°$). For comparison the field-free capture probabilities are shown as well.*

tally, the capture probability P_{cap} is averaged over all possible laser collision phases to obtain the phase-averaged capture probability,

$$\bar{P}_{cap}(b,\alpha) = \frac{1}{2\pi}\int_0^{2\pi} \mathrm{d}\phi P_{cap}(b,\phi,\alpha), \tag{4.4}$$

shown in Figure 4.5. For most impact parameters, the phase-averaged electron capture probability is larger for counterrotating than for corotating collisions, and noticeably smaller than for the field-free capture. For large impact parameters, the phase-averaged off-plane capture probabilities are significantly larger than for both co- and counterrotating collisions, and get closer to the field-free results.

It is found that the dependence of capture and ionization probabilities on the impact angle is rather smooth, such that no more than five different values $\alpha \in [0, 180°]$ need to be evaluated for the integration over $b\,\mathrm{d}b\,\mathrm{d}\alpha$ to yield sufficiently accurate total cross sections. By limiting the range of impact-vector orientations, the total capture cross section

α	$0°$	$\pm 90°$	$180°$
2D model	26.2		43.7
3D (fixed α)	26.3	63.4	40.5
3D	44.9		51.9

Table 4.2: *Comparison of total laser-assisted capture cross-sections for co-, counterrotating, and off-plane collisions (see text).*

is partitioned,

$$\sigma_{cap} = \int_{\alpha_{min}}^{\alpha_{max}} \mathrm{d}\alpha \int_{0}^{\infty} \mathrm{d}b\, b \bar{P}_{cap}(b,\alpha) \tag{4.5}$$

with $\alpha_{min} = 0°$, $\alpha_{max} = 360°$, into co- and counterrotating contributions, σ_{cap}^{co} and $\sigma_{cap}^{counter}$, by selecting $\alpha_{min} = -90°$, $\alpha_{max} = 90°$ and $\alpha_{min} = 90°$, $\alpha_{max} = 270°$, respectively. It obviously follows that the total cross section $\sigma_{cap} = \sigma_{cap}^{co} + \sigma_{cap}^{counter}$ is the sum of the co- and counterrotating half spaces. It is found that the contribution of counterrotating collisions is larger by

$$\frac{\sigma_{cap}^{counter} - \sigma_{cap}^{co}}{\sigma_{cap}} = 7.3\%, \tag{4.6}$$

as seen in the last line of Table 4.2. Also shown in Table 4.2 are total cross sections from the previous two-dimensional model[Niederhausen 04b] (Chapter 3), which neglects all off-plane contributions. The second row of Table 4.2 includes results for the full three-dimensional calculation where, however, only projectile trajectories with a fixed value for α are included in (4.5). Strikingly, for the reduced-dimensionality model (first row), the integrated in-plane cross sections do not differ significantly from full dimensionality "fixed-α" cross sections (second row), where the projectile moves in the plane of the laser-electric field, i.e., where $\alpha = 0°$ or $180°$ for corotating or counterrotating collisions, respectively. Thus for co- and counterrotating collisions the two-dimensional model simulates three-dimensional results for trajectories with $\alpha = 0°$ and $180°$ surprisingly well. A quantitative analysis, however, still requires the full three-dimensional calculation, i.e., the addition of off-plane trajectories (third row) for the cross-sections.

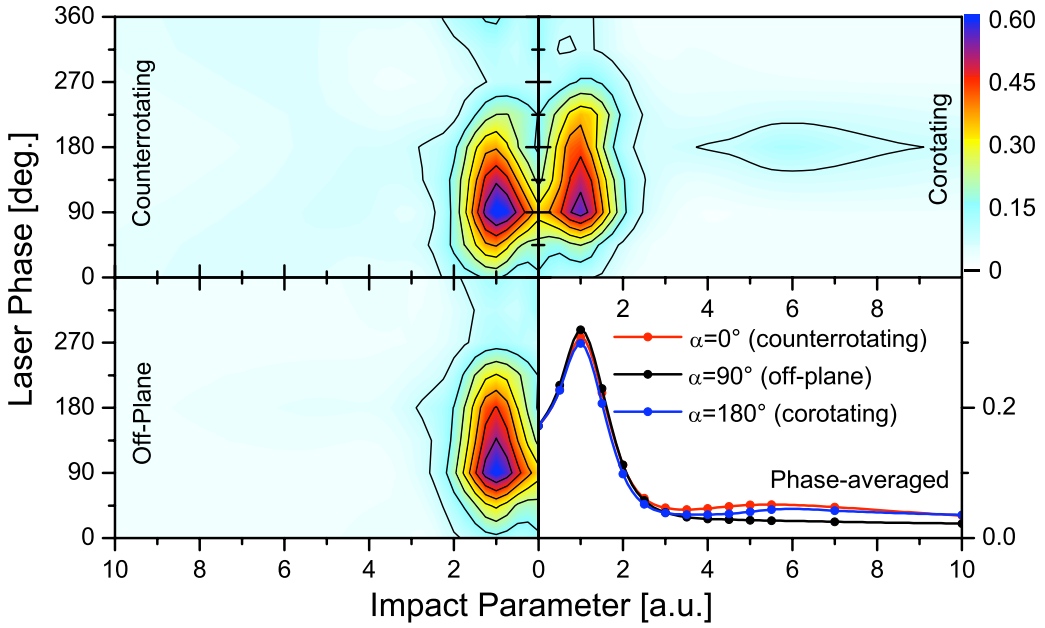

Figure 4.6: *Contour plots of the ionization probability as a function of the impact parameter and the laser collision phase for corotating, off-plane with $\alpha = \pm 90°$, and counterrotating collisions. Also shown are the phase-averaged results for the ionization for the three selected tilt angles α between the laser and collision plane.*

4.6 Ionization

The ionization probability in Figure 4.6 shows two main features:

- A maximum at $b \approx 1$ and a laser collision phase $\phi = 90°$, when the laser-electric field asserts an electric force on the electron in opposite direction to the projectile motion, and
- weak evidence for charge-resonant-enhanced ionization at $b \approx \pm 6$ and $\phi \approx 180°$ for corotating collisions (and a much broader and even weaker peak at $\phi \approx 0°$ for counterrotating collisions).

After phase-averaging, the ionization probability does not reveal a significant α-dependence or dichroism-effect as in the capture probability (Figure 4.6). Interestingly, the enhancement of the ionization at the classical Bohr-radius $b = 1$ is due to a combined effect of the

collision and the laser field. The field-free ionization probability for $0 < b < 7$ is very small (below 0.07) and cannot be distinguished from zero on the scale of Figure 4.6. Similarly, no significant ionization occurs for the interaction of the laser field with the target atom alone. However, due to the laser-electric field, the phase-averaged ionization probability increases dramatically and shows a pronounced maximum at $b = 1$. As a tentative explanation, a two-step process is suggested, where the collision promotes the electron into an excited target state, which subsequently gets ionized by absorbing a few photons from the laser-field.

«Why are things as they are and not otherwise? »
Johannes Kepler

Chapter 5

Controlled vibrational quenching of nuclear wave packets in D_2^+

5.1 Motivation

Enabled by significant advances in the technology of generating ultra–short and intense laser pulses over the past two decades, the nuclear dynamics in molecules has become observable in the time–domain[Posthumus 04,Stolow 04,Alnaser 04,Hertel 06,Lin 06]. In particular, the motion of vibrational and rotational wave packets in small diatomic molecules can now be observed at a time scale of 10 fs and below, i.e., at and below the molecules' natural vibrational and rotational time scales[Stapelfeldt 98,Litvinyuk 03,Ergler 06b,Rudenko 06].

In typical pump–probe experiments[Ergler 06b,Rudenko 06], the fast ionization of neutral D_2 molecules in an intense femtosecond pump laser pulse leads to the formation of molecular ions in a coherent superposition of excited rotational and vibrational states, i.e., to a moving nuclear (ro–) vibrational wave packet. Once launched, these wave packets propagate in the lowest $1s\sigma_g$ adiabatic potential curve of the molecular ion. The anharmonicity of this potential curve entails the rapid dephasing of the wave packet's vibrational state components within a few vibrational periods $T_{D_2^+} \approx 24$ fs of the initial wave packet[Ergler 06b]. This leads to the eventual collapse of the nuclear wave packet, which has been predicted theo-

retically[Feuerstein 03b] and recently confirmed experimentally[Ergler 06b,Rudenko 06,McKenna 07]. This collapse is – many vibrational periods later – followed by quarter (after ≈ 280 fs) and half revivals (after ≈ 560 fs)[Robinett 04], indicating a localized periodic motion of the wave packet as the vibrational states interfere in-phase again.

Going beyond the observation of the nuclear dynamics, possibly making an important contribution towards the overarching goal of achieving coherent control in chemical reactions[Hertel 06,Brixner 04], *Niikura et al.*[Niikura 04,Niikura 06] addressed the possibility of actively controlling the motion of vibrational wave packets by one additional short laser pulse. Using an intense "control" laser of relatively long pulse duration as compared to the present study, they investigated the controlled cooling, heating, and vibrational quenching into the vibrational ground state[Niikura 04]. More recently, the same group measured a strong increase of the dissociation yield of D_2^+ by applying a 8 fs control laser pulse when the vibrational wave packet is near its outer classical turning point[Niikura 06].

The idea of applying one (or several) control pulse(s) to a vibrational wave packet in the hydrogen molecular ion or its isotopes is the starting point for this chapter. A scheme of the pump – control – probe setup is shown for D_2^+ ion in Figure 5.1. It will be demonstrated that an appropriately delayed control pulse with respect to the pump pulse can modify the quantum dynamics of a nuclear wave packet by stimulating Raman transitions that alter the vibrational state composition of the wave packet in a controlled way, and that this control can be systematically improved by replacing a single control pulse with a carefully timed sequence of two or more short control pulses. In particular, it is found that through the appropriate choice of the control pulse parameters (delay, duration, and peak intensity), a given lower excited stationary vibrational state can be selected. The quality of this Raman–control mechanism can be tested experimentally by fragmenting the molecular ion with an intense probe pulse and by identifying the nodal structure of the surviving vibrational state in the kinetic energy release spectrum of the molecular fragments. The combined population and probing of a specific vibrational state points to a possibility for imprinting and retaining

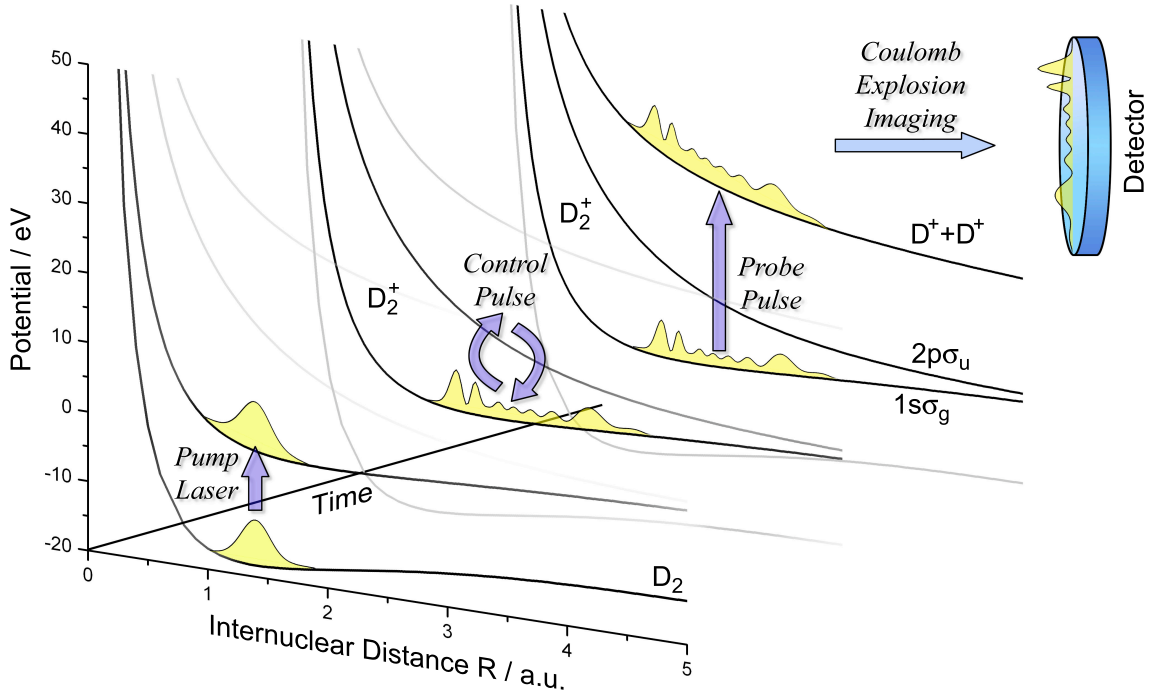

Figure 5.1: *Schematic diagram showing the ionization of* $D_2(\nu = 0)$ *by a pump pulse, followed by the modification of the vibrational wave packet on the* D_2^+ $1s\sigma_g$ *potential curve by a control pulse, and the final destructive analysis through Coulomb explosion imaging by a probe pulse.*

information in the nuclear wave function using a three (or more) laser pulse setup.

5.2 Theoretical Model

Starting with neutral D_2 molecules in their vibrational and electronic ground state, given by the product wave function $\chi^{D_2}_{\nu=0}(R)\psi_0(\mathbf{r}_1, \mathbf{r}_2)$, the generation of the D_2^+ vibrational wave packet is modeled by tunnel ionization in a short intense pump pulse of 6 fs (FWHM) and 1×10^{14} W/cm^2 peak laser intensity. In Born–Oppenheimer approximation the ionization can be treated as instantaneous, while the nuclei are fixed, since the time scale for the nuclear vibrations is two orders of magnitude slower than the electronic motion. This decoupling of the electronic from the nuclear wave function leads to a vertical transition from the D_2 to the D_2^+ potential curve at fixed internuclear distance R, known as *Franck–Condon principle* [Bransden 03], illustrated in Figure 5.2. In this approximation, the transition from the

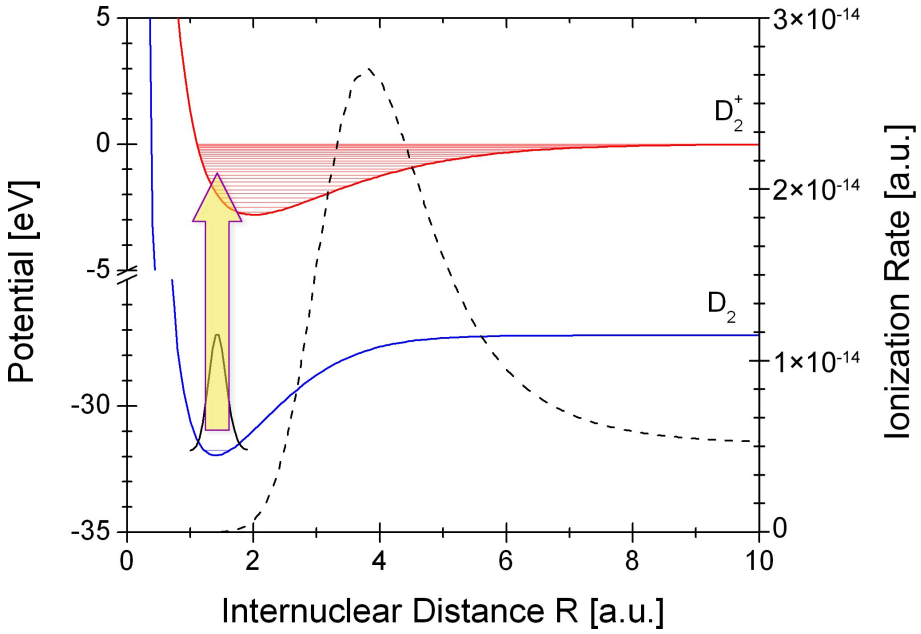

Figure 5.2: *Illustration of the instantaneous Franck–Condon transition from the* D_2 *ground electronic state to the* D_2^+ *potential curve, while the nuclear wave packet remains unchanged. The dotted line shows the ADK–based ionization rate at an electric field strength of* 2.75×10^{10} V/m *leading to a deviation from the vibrational distribution of the* D_2^+ *levels from the Franck–Condon model.*

vibrational state ν in D_2 to the vibrational state ν' in D_2^+ is given by the Franck-Condon factors

$$f_{\nu',\nu} = \int_0^\infty \chi_{\nu'}^{D_2^+ *}(R)\chi_\nu^{D_2}(R)\mathrm{d}R. \tag{5.1}$$

Because of the rapid increase of the molecular ionization rate[Saenz 00] with the internuclear distance R, shown by the dotted line in Figure 5.2, the applicability of the Franck–Condon approximation is questionable and known to generate a vibrational state distribution $\{\chi_\nu^{D_2^+}\}$ of the molecular ion that overestimates the population of higher excited vibrational states[Urbain 04]. The Franck–Condon factors (5.1) need to be modified accordingly to approximately account for the R–dependent transition,

$$f_{\nu',\nu} = \int_0^\infty \chi_{\nu'}^{D_2^+ *}(R)\sqrt{\gamma(R)}\chi_\nu^{D_2}(R)\mathrm{d}R, \tag{5.2}$$

where $\gamma(R)$ is the R-dependent ionization probability[Posthumus 04]. As a result, the relative

occupation of vibrational levels in the D_2^+ becomes laser–intensity dependent[Urbain 04], with a shift towards a larger occupation of the lower vibrational states. In addition, the R–dependent ionization rate predominately depletes the nuclear wave packet of the neutral molecule at larger values of R, thereby creating an oscillating nuclear wave packet on the D_2 potential curve, termed by the German word for hole–burning, "Lochfraß" [Goll 06,Ergler 06a]. In this work, however, only the vibrational motion of the nuclei on the ionized D_2^+ potential curves is considered.

The ionization of the neutral ground state molecular ion is modeled in terms of the molecular *Ammosov-Delone-Krainov* (ADK) tunneling ionization rate $\Gamma_{\mathrm{ADK}}(R,E)$ that depends on the peak electric field strength E of the pump laser and the internuclear separation R. A rate formula for the $\Gamma_{\mathrm{ADK}}(R,E)$ is given by *Brichta et al.*[Brichta 06]:

$$\Gamma_{\mathrm{ADK}}(R,E) = \left(\frac{3\mathrm{e}}{\pi}\right)^{3/2} \frac{\kappa(R)^{9/2}}{Z^{5/2}} \left(\frac{4\mathrm{e}\,\kappa(R)^4}{Z|E|}\right)^{2Z/\kappa(R)-3/2} \exp\left(-\frac{2\kappa(R)^3}{3|E|}\right), \tag{5.3}$$

where $\mathrm{e} = 2.718$ is the Euler constant, $Z = 2$ the charge after the ionization step, and $\kappa(R)$ is given by

$$\kappa(R) = \sqrt{2\left(V_{D_2^+ 1s\sigma_g}(R) - V_{D_2}(R)\right)}. \tag{5.4}$$

This rate is based on the original rate for tunneling ionization of (one–electron) atoms in a static electric field[Ammosov 86] and owes its dependence on R to the implicit generalization of the atomic ionization potential to the vertical (at a given value of R) energy gap between the adiabatic energy of the neutral diatomic molecule and its daughter molecular ion in their respective ground electronic states in $\kappa(R)$. Even though $\Gamma_{ADK}(R,E)$ does not depend on the molecular orientation relative to the electric field vector $\mathbf{E}$ of the pump laser, for single ionization of H_2 molecules that are perpendicularly oriented to $\mathbf{E}$, the isotropic rate leads to good agreement with the measured relative population of vibrational excited states in the H_2^+ isotope[Urbain 04]. It should be noted that for arbitrary orientation of the molecule relative to the laser electric field, orientation dependent molecular ADK rates[Tong 02] would produce more accurate results. However, since the molecular orientation can be determined

experimentally[Alnaser 04], for the purpose of this investigation, the orientation dependence of molecular ionization is disregarded.

The depletion of the initial vibrational ground state wave packet of the neutral D_2 molecule is calculated for a short 6 fs pump pulse of 1×10^{14} W/cm^2 intensity at 800 nm wavelength available from common Ti:S lasers,

$$\chi^{D_2}_{\text{pump}}(R) = \chi^{D_2}_{\nu=0}(R) \exp\left(- \int_{-\infty}^{\infty} \frac{\Gamma_{\text{ADK}}(R, E(t))}{2} \mathrm{d}t \right). \tag{5.5}$$

The function $\kappa(R)$ in (5.4) is obtained by using the hydrogenic ground state nuclear potential curve $V_{D_2}(R)$ for the H_2 or D_2 neutral molecule from the calculations by *Kołos* and *Wolniewicz*[Kołos 64,Kołos 65,Wolniewicz 93], and the binding ground state molecular ion potential $V_{D_2^+ 1s\sigma_g}(R)$ from *Bates et al.*[Bates 53]. The normalized initial wave packet on the D_2^+ potential curve is then approximated by the difference between the ADK-depleted and the initial nuclear wave function on the neutral molecular potential curve,

$$\chi^{D_2^+}_{\text{initial}}(R, t=0) = \frac{\chi^{D_2}_{\nu=0}(R) - \chi^{D_2}_{\text{pump}}(R)}{\left\| \chi^{D_2}_{\nu=0}(R) - \chi^{D_2}_{\text{pump}}(R) \right\|}. \tag{5.6}$$

In comparison, the Franck–Condon approximation assumes complete depletion of the neutral molecule leading to an initial wave packet on the upper potential curve of

$$\chi^{D_2^+}_{\substack{\text{Franck-Condon} \\ \text{initial}}}(R, t=0) = \chi^{D_2}_{\nu=0}(R). \tag{5.7}$$

For the time evolution of $\chi^{D_2^+}_{\text{initial}}(R,t)$ it is assumed that the wave packet is launched at the peak of the pump laser pulse, thereby defining the time $t = 0$ and the start of the molecular clock[Alnaser 04]. The field-free time-dependence of the nuclear wave packet is given by the decomposition into vibrational eigenstates $\chi^{D_2^+}_{\nu}(R)$,

$$\chi^{D_2^+}_{\text{initial}}(R,t) = \sum_{\nu} a_\nu \chi^{D_2^+}_{\nu}(R) \mathrm{e}^{-iE_\nu t}, \tag{5.8}$$

where the coefficients a_ν are obtained by projecting $\chi^{D_2^+}_{\text{initial}}(R)$ onto the known eigenstates $\chi^{D_2^+}_{\nu}$ (which are obtained from diagonalizing the Hamiltonian of the $D_2^+ 1s\sigma_g$ channel),

$$a_\nu = \left\langle \chi^{D_2^+}_{\nu}(R) | \chi^{D_2^+}_{\text{initial}}(R) \right\rangle. \tag{5.9}$$

Without further illumination by a laser pulse, the coefficients a_ν are time independent, while the wave packet propagating on the D_2^+ $1s\sigma_g$ potential curve would undergo cycles of dephasing and revivals[Feuerstein 03b,Robinett 04,Ergler 06b,Rudenko 06] due to different phase accumulations of the vibrational decomposition. However, the influence of one (or more) short and intense control pulses at variable delay times changes the shape of the wave packet $\chi^{D_2^+}(R,t)$ by altering the vibrational amplitudes $\{|a_\nu|\}$ due to Raman transitions and dissociation.

Allowing for Raman transitions between the two lowest adiabatic potential curves in D_2^+, a two–state model is adopted for the propagation of the nuclear wave packet on the $1s\sigma_g$ and $2p\sigma_u$ potential curves (unless stated otherwise atomic units are used),

$$i\frac{\mathrm{d}}{\mathrm{d}t}\left|\begin{array}{c}\chi_g\\ \chi_u\end{array}\right\rangle = \left(\hat{T}+\hat{V}+\hat{H}_c\right)\left|\begin{array}{c}\chi_g\\ \chi_u\end{array}\right\rangle \tag{5.10}$$

where the initial conditions for the nuclear wave packet components on the D_2^+ potential curves of gerade and ungerade symmetry are given by

$$\chi_g(R,t=0) = \chi^{D_2^+}_{\mathrm{initial}}(R) \tag{5.11}$$

$$\chi_u(R,t=0) = 0. \tag{5.12}$$

The kinetic energy operator $\hat{T} = \hat{\mathbf{p}}^2/(2\mu)$ includes the reduced mass $\mu = 1835$ of the two nuclei. The adiabatic electronic potential curves $1s\sigma_g$ and $2p\sigma_u$ of D_2^+ by *Bates et al.*[Bates 53] form the diagonal elements of the potential

$$\hat{V} = \begin{pmatrix} V_g(R) & 0 \\ 0 & V_u(R)\end{pmatrix}. \tag{5.13}$$

The dipole coupling between the gerade and ungerade potential curves in D_2^+, induced by one (or several) control pulses, is included in the off–diagonal elements of the coupling operator $\hat{H}_c$ and depends on the electronic dipole moment between the two adiabatic electronic states (ψ_g and ψ_u) $d_{gu}(R) = \langle\psi_u\,|z|\,\psi_g\rangle$ and the control laser electric field $E(t)$,

$$\hat{H}_c = \begin{pmatrix} -\frac{i}{2}\Gamma_g(R) & d_{gu}(R)E \\ d_{gu}(R)E & -\frac{i}{2}\Gamma_u(R)\end{pmatrix}. \tag{5.14}$$

The dipole couplings $d_{gu}(R)$ have been taken from the literature[Kulander 96]. Included in the diagonal elements of $\hat{H}_c$ are the isotropic R-dependent molecular ADK rates[Brichta 06] $\Gamma_g(R)$

and $\Gamma_u(R)$ for the ionization of D_2^+ similar as in (5.3) in order to account for Coulomb explosion during the control pulse(s). The time propagation is carried out by solving the time dependent Schrödinger equation (5.10) on a numerical grid, using the split–operator Crank–Nicholson scheme (see Chapter 2). The numerical grid in R extends from 0.05 to 30 with a spacing of 0.05, and a time step $\Delta t = 1$ for the nuclear motion is used. A quadratic optical potential $V_{\text{opt.}}(R) = -iW(R)$ (see Section 2.8) with

$$W(R) = \begin{cases} 0 & R < 20 \\ 0.01 \left(\frac{R-20}{10}\right)^2 & R \geq 20, \end{cases} \tag{5.15}$$

has been introduced at the outer grid end to avoid reflections, covering a width of 10 a.u..

The probability $P_\nu(t) = |a_\nu(t)|^2$ for finding the system at time $t > 0$ in the ν–th vibrational eigenstate is calculated as the quantum mechanical overlap of the bound wave function $\chi_g(R,t)$ with the known eigenfunctions $\chi_\nu^{D_2^+}(R)$ of the $1s\sigma_g$ curve (obtained by diagonalization of the Hamiltonian $\hat{T} + \hat{V}_g$),

$$P_\nu(t) = |a_\nu(t)|^2 = \left| \left\langle \chi_\nu^{D_2^+} | \chi_g(t) \right\rangle \right|^2 . \tag{5.16}$$

5.3 Vibrational Revivals

As a preliminary test of the propagation, the nuclear wave function revivals of the D_2^+ molecular ion are calculated. Figure 5.3(a) shows the coherent motion of the nuclear wave packet following a vertical Franck-Condon transition from the D_2 ground state, reproducing previous calculations by *Feuerstein* and *Thumm*[Feuerstein 03b]. The $\langle R \rangle$ expectation value shows a few oscillations before the nuclear wave packet collapses due to the anharmonicity of the potential, followed by a partial revival seen around $t = 560$ fs. The revival position is estimated from Figure 5.3 by taking the inner turning point of the wave packet at the approximate center of the revival structure. A weak indication for a quarter revival with the double oscillation frequency[Robinett 04] at $t = 290$ fs is also visible in the plotted probability density of the wave function.

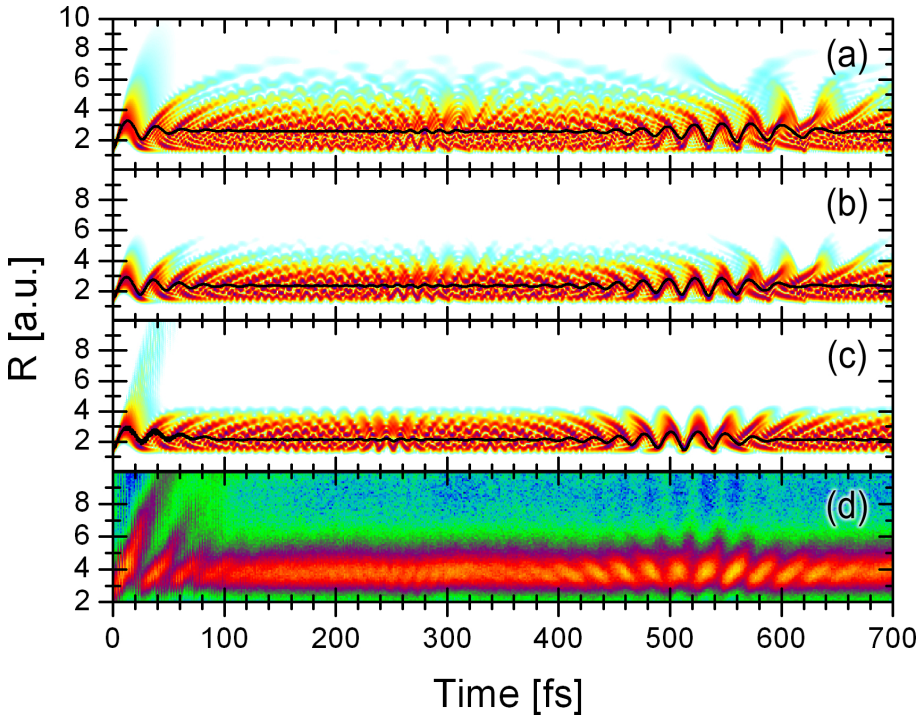

Figure 5.3: *Evolution of the wave function probability density for the* D_2^+ *nuclear wave packet in logarithmic scale. The black line in (a) – (c) shows the* $\langle R \rangle$ *expectation value. The initial wave packet is a (a) Franck–Condon distribution, (b) and (c) ADK ionization weighted wave packet for a* 6 fs, 3×10^{14} W/cm^2, *pump pulse, (c) with an additional* 50 fs, 3×10^{13} W/cm^2 *pedestal of the pump pulse centered at* $t = 0$. *Panel (d) shows the reconstructed distribution of the wave packet probability density as a function of* R *and the pump probe delay following Coulomb's law for the mapping of the kinetic energy release to the internuclear distance by Ergler et al.*[Ergler 06b,Ergler 06c] *for a pump pulse as in (b).*

The time evolution of the nuclear density can be obtained in great detail by applying sub–fs probe pulses in a pump – probe setup from mapping the kinetic energy release on the Coulomb–explosion $1/R$ curve[Tong 06]. The kinetic energy release measures the sum of kinetic energy and Coulomb energy at the moment of explosion for the particles, but is largely dominated by the Coulomb part, as the kinetic energy for the bound states is rather small. This allows the for the approximate energy – position mapping[Feuerstein 03a] by neglecting the kinetic energy of the wave packet at the time of the ionization and using Coulomb's law for the reconstruction of the nuclear wave function probability density. The experimental observation by *Ergler et al.*[Ergler 06b,Ergler 06c] is shown in Figure 5.3(d). The time–resolution is

limited as they employ 7 fs probe pulses, which is consistent with the reconstructed resolution predicted by *Lin et al.*[Lin 06]. Apart from the good qualitative agreement between experiment and theory, a shift in the revival position is clearly noticeable. While the predicted position of the half–revival using a Franck–Condon initial state is around 560 fs, the experimental findings suggest a revival around 520 fs.

To address this discrepancy, Figure 5.3(b) shows the evolution of the nuclear density for an improved initial wave packet (5.6), obtained from the R–dependent ionization of the neutral molecule in the pump pulse. The parameters of the pump pulse (6 fs, 800 nm, 3×10^{14} W/cm^2) are identical with the experiment[Ergler 06b]. The shift of the revival position towards smaller times is evident, but at around 540 fs slightly above the experimental value.

So far it was assumed, that the nuclear wave packet on the $1s\sigma_g$ potential curve is created instantaneously at the center of the pump pulse and propagates field–free thereafter. Including the remaining 3 fs tail of the electric field of the pump pulse in the calculations, thereby coupling the nuclear wave function with the $2p\sigma_u$ channel, did not reveal a different revival position. That is because the dipole coupling to the upper potential curve is weak at the inner turning point where the wave packet starts, and with an oscillation period of 24 fs the wave packet does not move significantly in the $1s\sigma_g$ curve during the remaining few fs of the pump pulse. As a possible explanation for the additional shift of the revival position, a longer background pedestal of the pump pulse is suggested: Figure 5.3(c) demonstrates good quantitative agreement with the experimental positions of the half and quarter revivals, if a long 50 fs pedestal of the pump pulse at 10% of the peak intensity is used.

5.4 One Control Pulse

In this numerical application, the influence of a single control pulse is investigated. For a 6 fs pulse with 1×10^{14} W/cm^2 intensity, more than 94% of the initial molecule remains bound. No ionization of the D_2^+ from the control laser occurred at the chosen laser intensity. Compared with the vibrational period $T_{D_2^+} \approx 24$ fs of the molecule seen in the oscillations

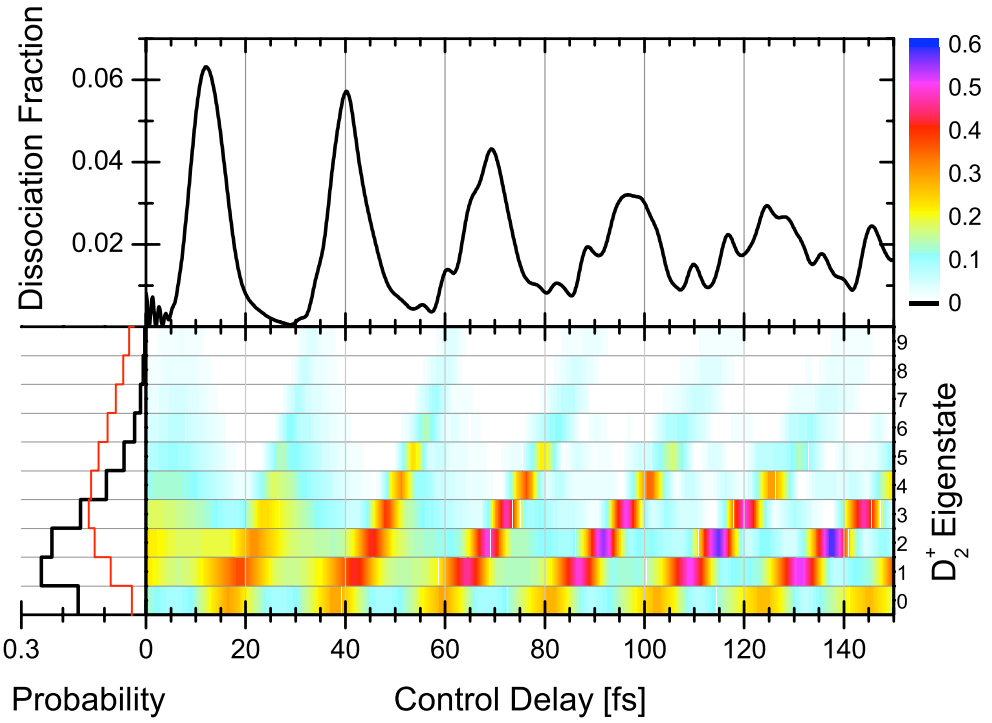

Figure 5.4: *D_2^+ dissociation probability (top) and vibrational state distribution (bottom) after the control pulse as a function of the control pulse delay. The initial vibrational state population for D_2^+ after the pump pulse is shown at the left (thick black line) and compared with the distribution arising from a vertical Franck–Condon transition (thin red line). The pump pulse and the control pulse are both* 6 fs *(FWHM) Gaussian pulses with an intensity of* 1×10^{14} W/cm^2.

of the $\langle R \rangle$ expectation value in Figure 5.3, the control pulse is considered short and leads to a nearly instantaneous transition between the different electronic states. The initial wave packet (5.6) is obtained from the ADK–depleted ground state of the neutral molecule using a 6 fs, 1×10^{14} W/cm^2 pump pulse. Figure 5.4 shows the final vibrational state distribution (5.16) as a function of the control pulse delay. It is seen that after certain delay times the vibrational wave packet has largely collapsed to one specific vibrational level. The control delay times at which the relative contribution of a stationary vibrational state $\chi_\nu^{D_2^+}$ is most prominently enhanced, is related to the classical period T_ν for the motion of a particle (of mass μ) on the binding potential curve V_g with the energy E_ν of the vibrational

state $\chi_\nu^{D_2^+}$,

$$T_\nu = 2 \int_{R_{\min}}^{R_{\max}} \mathrm{d}R \sqrt{\frac{\mu}{2(E_\nu - V_g(R))}} \tag{5.17}$$

where $R_{\min}$ and $R_{\max}$ are the classical turning points. However, since the center of the initial wave function $\chi_{\text{initial}}^{D_2^+}(R)$ is created outside the inner turning point, the delay times $\tilde{T}_\nu$ at which the first relative population surge of a given stationary vibrational state contribution occurs, is shifted to slightly smaller delays ($\tilde{T}_\nu < T_\nu$). Due to the anharmonicity of the potential, and consistent with the increase of T_ν with ν, the time delay difference between population surges of different vibrational state contributions increases with increasing ν.

Due to the rapid increase of the dipole coupling $d_{gu}(R)$ with R, the dissociation yield is largest near the outer turning point and negligible near the inner turning point (top graph in Figure 5.4). For a few vibrational periods, the dissociation yield oscillates with the vibrational period of the molecular ion. However, as the wave packet dephases[Feuerstein 03b,Robinett 04], the oscillations decrease in amplitude and become increasingly smeared out.

Figure 5.5 (top panel) shows the time evolution of the wave packet's probability density without a control pulse. The wave function starts to quickly dephase due to different phase accumulations of its stationary vibrational states components (5.8), and the center of the nuclear wave packet (superimposed curves in Figure 5.5, computed as the expectation value of R), becomes stationary at $\langle R \rangle \approx 2.6$ after a propagation time of approximately 80 fs. Deviations from this equilibrium distance (see Section 5.3) only occur many vibrational cycles later, due to wave packet revivals[Feuerstein 03b,Robinett 04,Ergler 06b,Rudenko 06]. The two particular control delay times of 91.7 fs (Figure 5.5, middle panel) and 96.3 fs (bottom panel) equal $4 \times T_{\nu=2}$ and $4 \times T_{\nu=3}$ for the second and third vibrationally excited state, respectively. As seen in Figure 5.4, a control pulse applied at delay times that equals a multiple of the oscillation period (5.17) of a particular state, strongly enhances the population of that state. At the chosen delay times for the control pulse, an optimal enhancement of the vibrational occupation for the $\nu = 2$ and $\nu = 3$ state is obtained, with $P_2(91.7\text{ fs}) = 55.1\%$

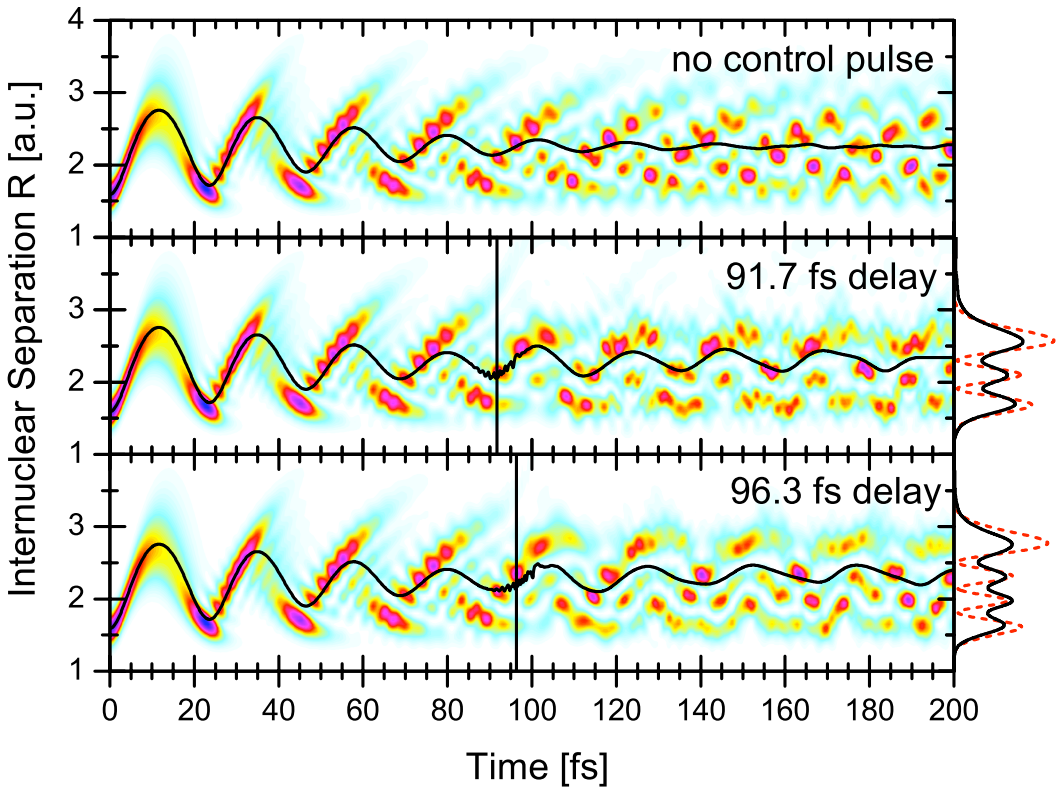

Figure 5.5: *Time evolution of the nuclear wave function probability density for the field free case (top) and with a control pulse at a delay of* 91.7 fs *(middle) or* 96.3 fs *(bottom). The solid line represents the center of the bound wave packet* $\langle R \rangle$*. The two vertical lines indicate the control pulse delays. The dashed line in the side panel shows the known nodal structure for the 2nd and 3rd vibrational excited state, respectively, while the thick line is the average of the wave packet probability density over the* 100 fs *directly following the control pulse.*

and $P_3(96.3 \text{ fs}) = 49.9\%$, respectively. Also seen from the top panel of Figure 5.4 is that more than 97% of the molecule remains bound at the two control pulse delay times.

If the control pulse completely quenches the wave packet into a single vibrational eigenstate, its probability density would be stationary and display the nodal structure of that state in terms of horizontal "stripes" in Figure 5.5 (middle and bottom panel). This nodal structure could be imaged by further ionizing the molecular ion in a sudden vertical transition onto the repulsive $D^+ + D^+$ potential curve with an intense and short probe pulse. This technique of mapping nuclear probability densities onto the kinetic energy release spec-

trum of the emitted fragments is well established and commonly referred to as laser induced "Coulomb explosion imaging" [Alnaser 04,Ergler 06b,Niikura 06,Stapelfeldt 98,Litvinyuk 03]. It is thus possible, with existing technology, to quantify the extent to which the vibrational distribution of the nuclear wave packet can be compressed. In the present examples, the more than 50% population of the second or third vibrational excited state contributions explains the emergence of the nodal structure of these states as wavy lines with small amplitudes in the middle and bottom panel of Figure 5.5 and can more clearly be seen in the time–average over the 100 fs immediately after the control pulse (side panels of Figure 5.5): minima and maxima appear at the same positions as for the stationary $\chi_2^{D_2^+}$ and $\chi_3^{D_2^+}$ wave functions. Further refinement of this selection can be achieved by adjusting the shape of the control pulse or by applying several control pulses. This would tend to more distinctly display wave function nodes in the kinetic energy release spectrum.

The ripples near the control delay time in the expectation value $\langle R \rangle$ in Figure 5.5, appear due to the Rabi oscillations between the gerade and ungerade potential curves and explain the physical mechanism behind the controlled change of the vibrational wave packet: successive Raman transitions between vibrational states on the $1s\sigma_g$ potential curve, mediated through dipole transitions between the two lowest adiabatic electronic states of D_2^+.

5.5 Two Control Pulses

In an attempt to further control the "slowing down" of the vibrational wave packet and with the ultimate goal in mind of "stopping" [Niikura 04] it into a given stationary vibrational state, the action of two control pulses is now investigated. Both control pulses are 6 fs (FWHM) at a laser intensity of 1×10^{14} W/cm^2. A scheme for enhancing a particular (lower) vibrational state is demonstrated for the $\nu = 2$ state. For the first control pulse an optimal delay time of 70.7 fs was found that produces a nuclear wave packet with maximum population in the $\chi_2^{D_2^+}$ and $\chi_3^{D_2^+}$ vibrational states (Figure 5.6). The delay time of the second control pulse with respect to the first control pulse is then varied and a strong enhancement of the $\nu = 2$

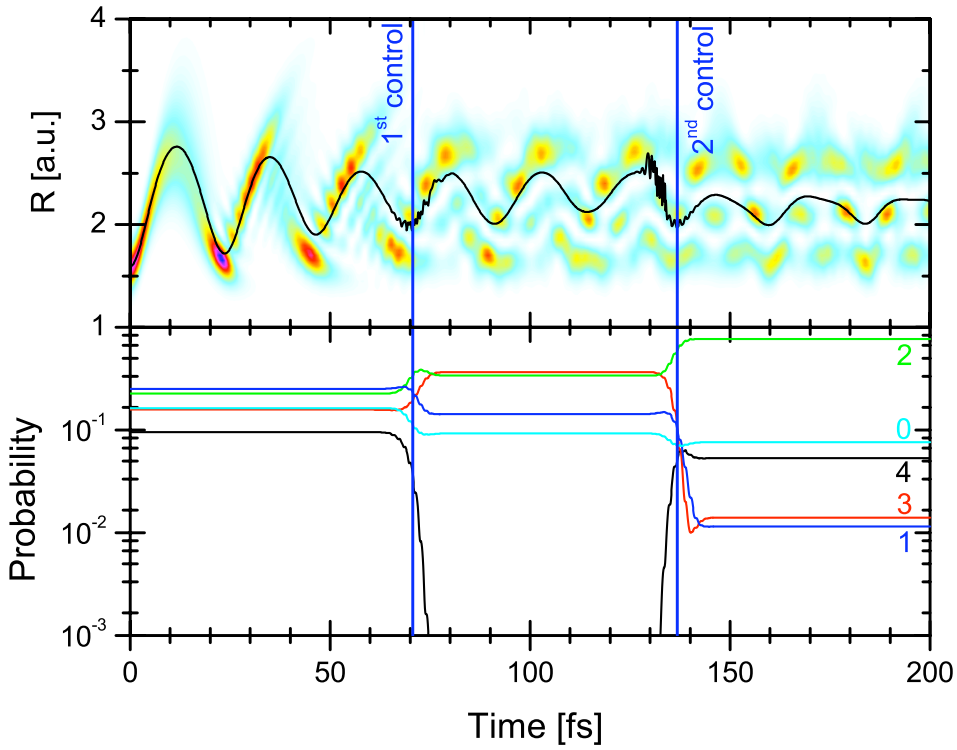

Figure 5.6: *Time evolution of the nuclear wave function probability density (top) and the few lowest vibrational states (bottom, logarithmic scale) for two* 6 fs *(FWHM),* 1×10^{14} W/cm^2 *control pulses with delay times of* $\tau_1 = 70.7$ fs *and* $\tau_2 = 136.8$ fs *relative to the start of the wave packet. The superimposed curve in the top graph shows the center motion* $\langle R \rangle$ *of the bound wave packet.*

vibrational state contribution is found at a delay time of 66.1 fs when the second control pulse vibrationally cools the wave packet mainly from $\chi_3^{D_2^+}$ to $\chi_2^{D_2^+}$ state. The underlying Raman transitions can be seen in the evolution of the vibrational states in Figure 5.6. For this sequence of control pulses the bound vibrational wave packet of the molecular ions owes 77.3% to the stationary $\nu = 2$ state, while the total dissociation yield is below 3%. This is in contrast to the control scheme by *Niikura et al.*[Niikura 04] that simulates nearly complete vibrational cooling in the $\nu = 0$ vibrational ground state at higher dissociative loss (about 25%).

With regard to the influence of the laser carrier envelope phase, no significant effect on the vibrational decomposition of the wave packet has been obtained. Investigating the

control pulse–length–dependence, it is found that longer pulses increase the dissociation yield and blur the enhancement of specific vibrational states. The latter can be understood by considering the bandwidth of the control pulse that narrows for longer pulses and thereby reducing the appropriate frequencies for Raman transitions. Higher laser intensities (above 2×10^{14} W/cm^2) significantly increase the break up of the molecule while the degree of control for preparing a nuclear wave packet in a certain vibrational eigenstate decreases at lower intensities.

«Fourier's Theorem is not only one of the most beautiful results of modern analysis, but it may be said to furnish an indispensable instrument in the treatment of nearly every recondite question in modern physics. »
William Thomson and P. G. Tait

Chapter 6

Time–Series Analysis of Vibrational Wave Packets

6.1 Introduction

The anticipated efficient catalysis and control of chemical reactions with laser light has been the motivation for many detailed theoretical and experimental investigations over the past two decades[Weber 05,Hertel 06]. Laser pulses of a single optical cycle have been produced[Shverdin 05] and sub–femtosecond pulses in the XUV range routinely become available in several laboratories worldwide[Baltuška 03,Agostini 04,Shan 05,Sansone 06,Scrinzi 06]. This allows for "photographing" atoms and molecules with intense ionizing short laser pulses, and in combination with a series of time–resolved images to observe the dynamics on a quantum scale[Niikura 02,Niikura 03]. Particularly, the vibrational[Alnaser 05,Katsuki 06,Ergler 06b] and rotational[Litvinyuk 03,Dooley 03] motion in small diatomic molecules has recently become accessible.

Typically, the time–dependent nuclear motion is detected by using a sequence of two short laser pulses (Figure 6.1), such that a first pump pulse generates a coherent nuclear wave packet on the single ionized nuclear potential curve, while a second time–delayed probe pulse fragments the molecule by Coulomb explosion[Stapelfeldt 98]. The measured kinetic energy release of the fragments can then be accurately mapped onto the internuclear distance R at

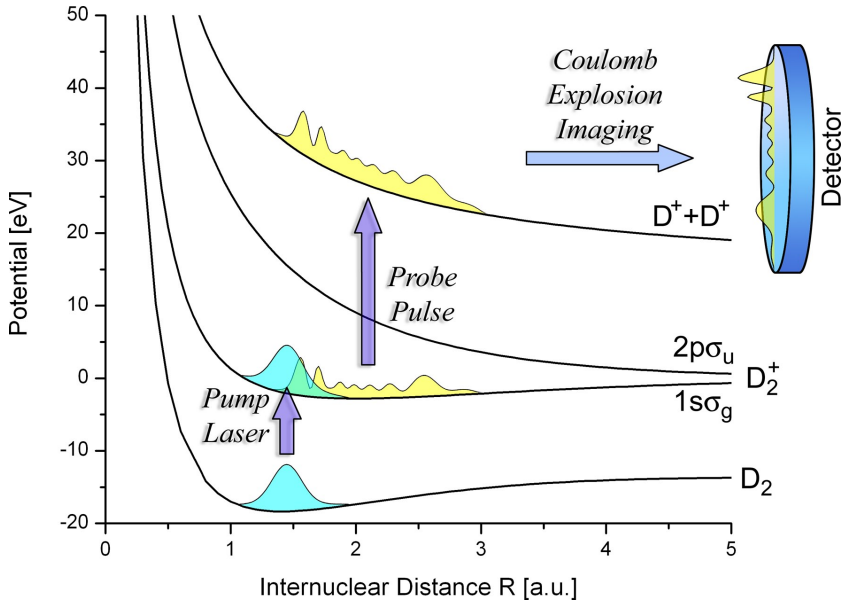

Figure 6.1: *Schematic diagram of the pump–probe setup. A first laser pulse launches a nuclear wave packet on the D_2^+ $1s\sigma_g$ potential curve and starts the molecular clock. After a certain delay time, an intense short probe pulse promotes the nuclear wave packet onto the $2D^+$ repulsive $1/R$ Coulomb–explosion curve and allows for the detection of the fragment kinetic energy distribution, which in turn is used for obtaining a "snapshot" of the nuclear wave function probability density at the time of fragmentation.*

the time of the ionization following Coulomb's law. Since each measurement destroys the molecule, recording a series of kinetic energy spectra for variable delay times of the probe laser, reveals the motion of the nuclear wave packet. After initial indication for vibrational oscillations in D_2^+ using 8 fs probe pulses[Alnaser 05], recent experiments[Ergler 06b] have observed the vibrational motion of D_2^+ over several picoseconds and confirmed theoretical predictions of vibrational revival times[Feuerstein 03b] (see also Section 5.3).

The purpose of this chapter is to present an imaging scheme, based on the Fourier analysis of the time and internuclear separation R dependent wave function probability density $\rho(R,t)$, that allows one to directly obtain the distribution of vibrational states and the mapping of the nuclear potential curves. Experimentally, $\rho(R,t)$ can be obtained from delay dependent time–series measurements of the kinetic energy release spectrum in a pump – probe scenario (Figure 6.1). This method is demonstrated theoretically for the D_2^+ molecule

but could be extended to more complex molecules as well. In addition, the present method could be applied to observe other processes such as dissociation, temporary bond formation, ionization or vibrational excitation and is the subject of ongoing investigation[Feuerstein 07]. As shown in Figure 6.1, a coherent superposition of the D_2^+ rovibrational wave packet is created by tunnel ionization from the neutral D_2 ground state by a short and intense laser pulse. The numerical simulation of the wave packet propagation on the $1s\sigma_g$ and $2p\sigma_u$ potential curves is identical to the previous Chapter 5. Without the presence of an external field, the motion of the wave packet is restricted to the $1s\sigma_g$ curve and will be considered first. In the second half of this chapter the influence of a pump and a probe laser pulse on the dynamics of the vibrational wave packet will be investigated.

The initial wave packet is created from the neutral molecule with a short and intense pump pulse as a coherent superposition of vibrational states. The vibrational state distribution is either determined from the Franck–Condon principle, or by using the R–dependent tunnel ionization rate from the neutral molecule[Saenz 00] (see Section 5.2).

6.2 Bound Nuclear Motion

Following the ionization of a short intense pump pulse, the D_2^+ molecule is created in a coherent superposition of vibrational eigenstates $\chi_\nu(R)$:

$$\Psi(R,t) = \sum_\nu a_\nu \mathrm{e}^{-iE_\nu t}\chi_\nu(R), \tag{6.1}$$

where the wave packet is completely characterized by the complex amplitudes $\{a_\nu\}$ of the eigenstates χ_ν. Clearly, without the presence of an external perturbation, the a_ν are explicitly time–independent and the nuclear wave function evolves on the adiabatic binding $D_2^+ 1s\sigma_g$ potential curve. The anharmonicity of the potential results in a quick dephasing of the wave packet and only after a long propagation time the vibrational states overlap in–phase, leading to wave function revivals. The time evolution of the nuclear wave packet up to 3 ps is shown in Figure 6.2(a), and except for the longer propagation time, is identical

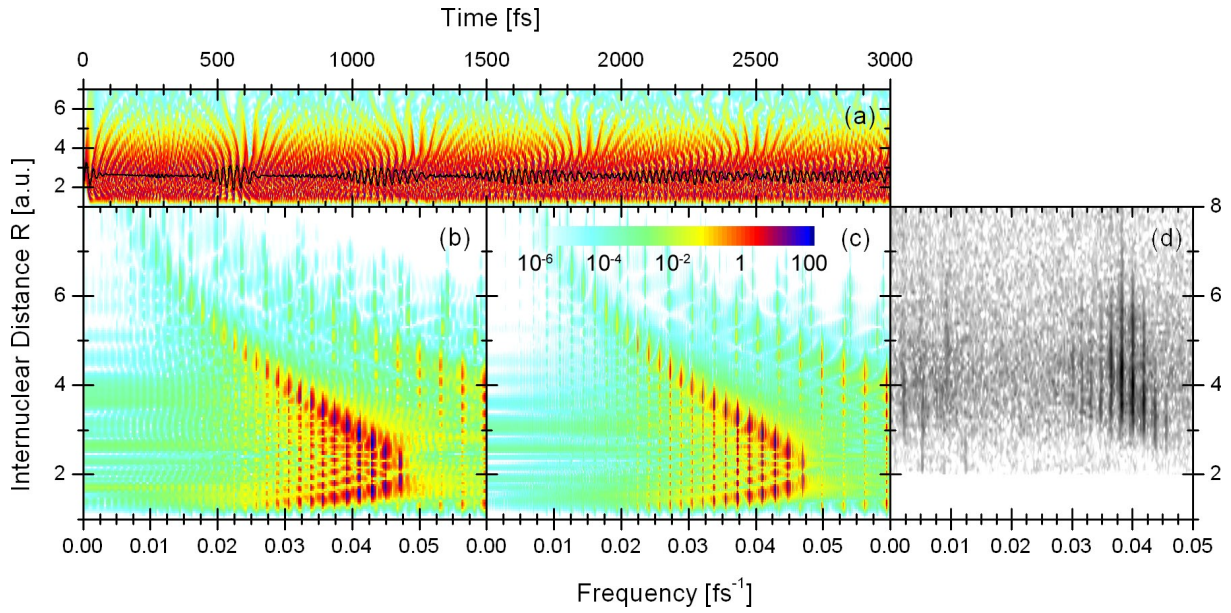

Figure 6.2: *Time evolution of the D_2^+ vibrational wave function. Probability density and the expectation value of the internuclear distance $\langle R\rangle$ (thick line) (a). The power spectrum (squared Fourier amplitudes) of the probability density shown in (a) is plotted in logarithmic scale for a* 3 ps *(b) and a* 10 ps *(c) propagation time. Panel (d) exhibits the experimental distribution extracted from the kinetic energy release measurement of coincident D^+ pairs and mapping on the internuclear distance using Coulomb's law*[Feuerstein 07] *for parameters as in (b).*

to previous work of *Feuerstein* and *Thumm*[Feuerstein 03b]. Additional wave function revivals can be seen in the $\langle R\rangle$ expectation value at times 1.1 ps, 1.65 ps and faintly around 2.2 ps.

In an experiment only the kinetic energy release after a probe pulse is observable, which can be mapped using Coulomb's law to reconstruct the evolution of the probability density $\rho(R,t) = |\Psi(R,t)|^2$. Then the nuclear density $\rho(R,t)$ as a function of time becomes

$$\rho(R,t) = \sum_{\mu,\nu} a_\mu^* a_\nu \mathrm{e}^{-i(E_\nu - E_\mu)t} \chi_\mu^*(R)\chi_\nu(R) \tag{6.2}$$

$$= \sum_{\nu} |a_\nu|^2 |\chi_\nu(R)|^2 + \sum_{\mu\neq\nu} a_\mu^* a_\nu \mathrm{e}^{-i(E_\nu - E_\mu)t} \chi_\mu^*(R)\chi_\nu(R). \tag{6.3}$$

The first diagonal term does not depend on time and gives an incoherent background to the wave function probability density. Of course, for a single vibrational state the second term vanishes and the wave function density becomes stationary (cf. Chapter 5).

The diagonal contribution in (6.3) can be removed by subtracting the time averaged probability density, and defining a coherent part c,

$$c(R,t) \equiv \rho(R,t) - \frac{1}{T}\int_0^T \rho(R,t)\mathrm{d}t \tag{6.4}$$

$$= \sum_{\mu\neq\nu} a_\mu^* a_\nu \mathrm{e}^{-i(E_\nu - E_\mu)t} \chi_\mu^*(R)\chi_\nu(R), \tag{6.5}$$

which converges to the double–sum over the vibrational states for the off–diagonal contribution in (6.3) in the limit of large propagation times $T \to \infty$. The Fourier transform of $c(R,t)$ at a fixed internuclear distance R yields

$$\tilde{c}(R,\omega) = \frac{1}{\sqrt{2\pi}}\int_0^T c(R,t)\mathrm{e}^{-i\omega t}\mathrm{d}t \tag{6.6}$$

$$= \sqrt{2\pi}\sum_{\mu\neq\nu} a_\mu^* a_\nu \chi_\mu^*(R)\chi_\nu(R)\delta_T(\omega_{\mu,\nu} - \omega), \tag{6.7}$$

where $\omega_{\mu,\nu} = \omega_\mu - \omega_\nu$ and $\delta_T(\Omega)$ is defined as

$$\delta_T(\Omega) = \frac{1}{2\pi}\int_0^T \mathrm{e}^{i\Omega t}\mathrm{d}t, \tag{6.8}$$

reflecting the finite sampling time, hence a limitation on the frequency resolution. In the limit $T \to \infty$, δ_T approaches the δ–function, $\lim_{T\to\infty} \delta_T(\Omega) = \delta(\Omega)$. The nuclear wave packet in the frequency (energy) domain in (6.7) enables one to obtain the energy spectrum of vibrational states $\sum_{\mu\neq\nu}\delta_T(\omega_{\mu,\nu}-\omega)$, the spatial distribution of the eigenstates $\chi_\mu^*(R)\chi_\nu(R)$, and the complex expansition coefficients $a_\mu^* a_\nu$. Evidently, the integral of $\tilde{c}(R,\omega)$ over R has to vanish, since the vibrational states are orthogonal, and it can therefore be utilized as a numerical test. In the following, the relative phase in $\tilde{c}(R,\omega)$ is not of interest and only the power spectrum, i.e. the squared amplitudes of $\tilde{c}$ are considered:

$$C(R,\omega) \equiv |\tilde{c}(R,\omega)|^2 = 2\pi\left|\sum_{\mu\neq\nu} a_\mu^* a_\nu \chi_\mu^*(R)\chi_\nu(R)\delta_T(\omega_{\mu,\nu}-\omega)\right|^2 \tag{6.9}$$

$$\approx 2\pi\sum_{\mu\neq\nu} |a_\mu|^2|a_\nu|^2|\chi_\mu(R)|^2|\chi_\nu(R)|^2\delta_T(\omega_{\mu,\nu}-\omega) \tag{6.10}$$

Approximating the square of the sum by the squares of the summands is justified, as the δ–function only contributes non–vanishing terms for a finite set of frequencies $\omega = \omega_{\mu,\nu}$.

From the numerically propagated wave packet $\rho(R,t)$ in Figure 6.2(a), the power spectrum $C(R,\omega)$ (squared Fourier amplitudes) is shown in Figure 6.2(b) and (c) for two maximum propagation times $T = 3$ ps (b) and $T = 10$ ps (c). Clearly seen is that the longer propagation time leads to a better frequency resolution of the vertical stripes. However, the relevant structure is already resolved at a total propagation time of 3 ps and allows for comparison with first experimental results[Feuerstein 07], shown in Figure 6.2(d). Distinct vertical lines are seen in all three figures, (b) – (d), corresponding to the coherence frequency $\omega_{\nu+1,\nu}$ of neighboring vibrational states $\chi_{\nu+1}$ and χ_ν. Due to the anharmonicity of the $D_2^+ 1s\sigma_g$ potential curve, the energy spacing between adjoint vibrational levels $\omega_{\nu+1,\nu}$ decreases with increasing ν, such that the lowest states appear at larger frequencies. For example, for the ground state $\nu = 0$ and the first excited state $\nu = 1$, a frequency spacing of $\omega_{1,0} \sim 2\pi \times 0.47$ fs^{-1} is seen in the figure, corresponding to a vibrational period of 21.3 fs.

Upon closer inspection, each vertical line reveals a substructure that owes its R dependence to the overlap of the wave function product,

$$C(R,\omega_{\nu+1,\nu}) = 2\pi |a_{\nu+1}|^2 |a_\nu|^2 |\chi_{\nu+1}(R)|^2 |\chi_\nu(R)|^2, \tag{6.11}$$

That is, the nodal structure of the both vibrational eigenstates is directly seen in Figure 6.2(b) and (c), with the number of nodes increasing for the higher excited states and mapping the shape of the binding potential curve. Yet, the experimental resolution in Figure 6.2(d) does not allow the identification of wave function nodes.

6.3 Vibrational State Distribution

So far only interferences between neighboring vibrational states with $\Delta\nu = 1$ have been considered. Higher series with $\Delta\nu = 2$ and $\Delta\nu = 3$ are seen in Figure 6.3(b) and (c) for different distributions of vibrational states. In Figure 6.3(c) a Franck–Condon distribution

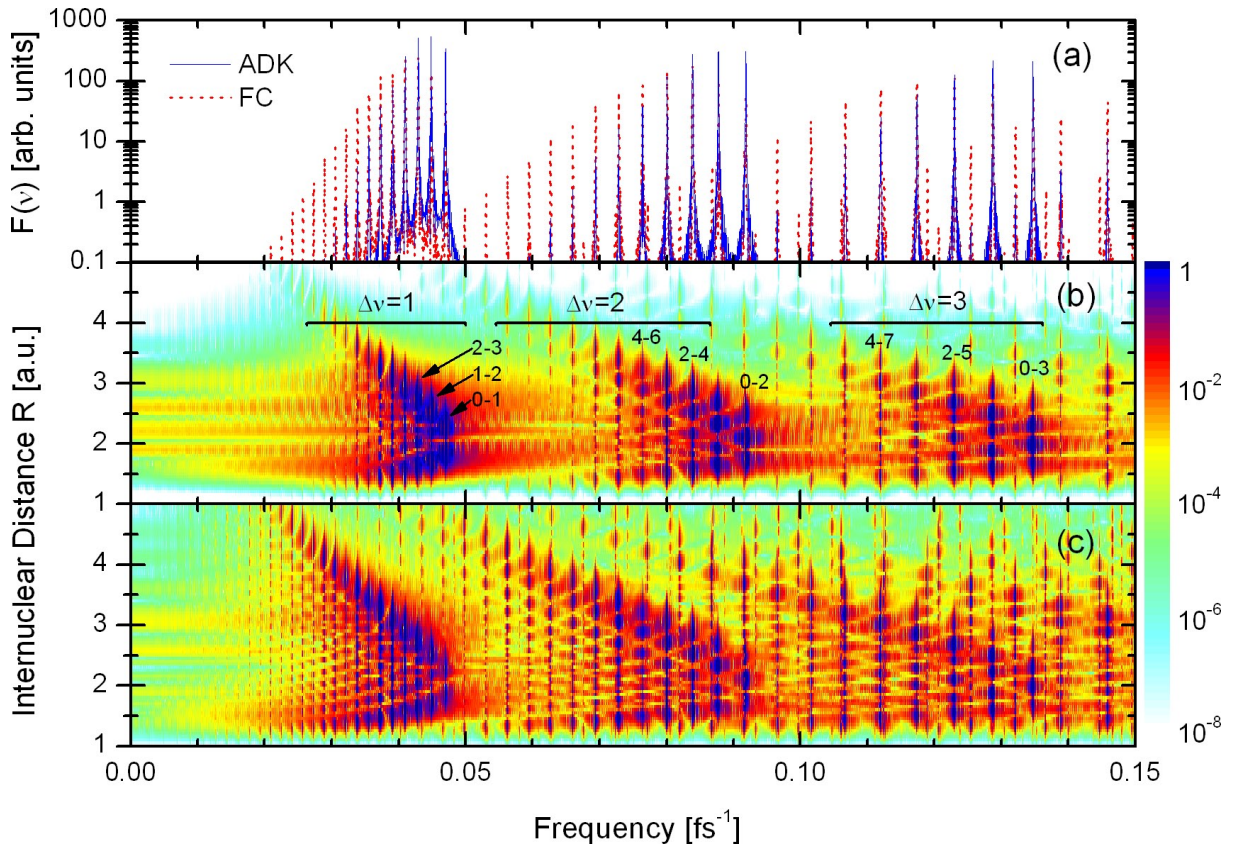

Figure 6.3: *The bottom two panels show $C(R,\omega)$ as in Figure 6.2(c). Differences are seen for an initial Franck–Condon wave packet (c) and the ADK–modified initial wave function (b). Indicated are three series for $\Delta\nu = 1, 2$ and 3 of the vibrational eigenstates and a few pairs for the coherent contributions between selected vibrational states. The panel (a) shows the projections $F(\omega)$ of (b) and (c) on the frequency axis (integration over R).*

is assumed, while in Figure 6.3(b) a more realistic ADK–model has been used, taking the R–dependent ionization rate from the neutral molecule in the first (6 fs, 1×10^{14} W/cm^2, 800 nm) pump pulse into account[Saenz 00] (see Section 5.2 for details). Especially for the Franck–Condon distribution, known to overestimate the population of the higher excited vibrational states[Urbain 04], the series of different $\Delta\nu$ are seen to overlap each other. In particular, contributions from higher $\Delta\nu > 3$ series are seen to reach into the range of frequencies for $\Delta\nu = 3$. A better separation of the different series is found for the ADK populated initial state.

Since each of the vertical lines is a distinct measure of the occupation of the contributing

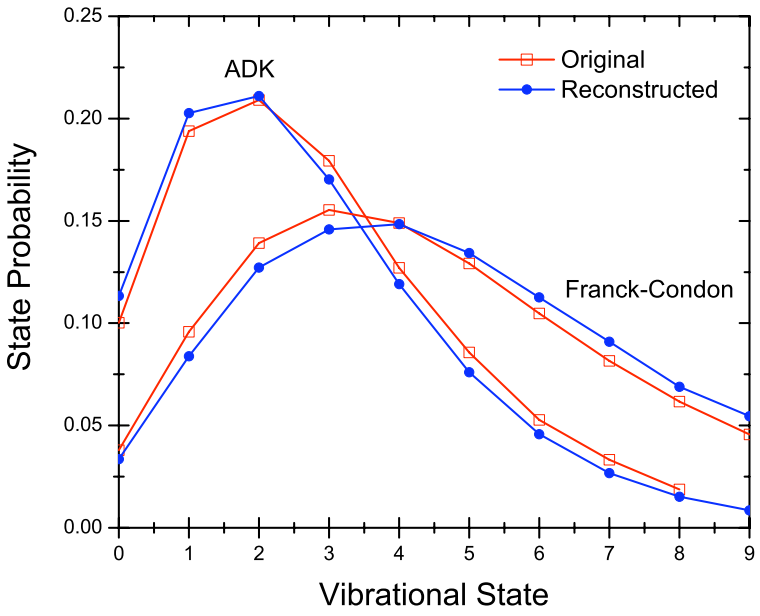

Figure 6.4: *Original (squares) and reconstructed (dots) vibrational distribution* $\{|a_\nu|^2\}$ *for Franck–Condon and ADK–modified initial states.*

pair of vibrational states, integrating (6.10) along the radial coordinate gives

$$F(\omega) \equiv \int_0^\infty C(R,\omega)\mathrm{d}R = 2\pi \sum_{\mu\neq\nu} |a_\mu|^2 |a_\nu|^2 f_{\mu,\nu}\, \delta_T(\omega_{\mu,\nu} - \omega), \tag{6.12}$$

where $f_{\mu,\nu}$ is defined as the spatial overlap integral of the pair of vibrational eigenfunctions χ_μ and χ_ν:

$$f_{\mu,\nu} \equiv \int_0^\infty |\chi_\mu(R)|^2 |\chi_\nu(R)|^2 \mathrm{d}R. \tag{6.13}$$

The values of $f_{\mu,\nu}$ can easily be obtained from the known eigenfunctions, which are computed by diagonalization of the potential. In Figure 6.3 the projection $F(\omega)$ is shown for both initial state distributions (Franck–Condon and ADK model). The area $A_{\mu,\nu}$ under each peak in (6.12), corresponding to the pair of states μ and ν, is related to product of the vibrational state occupation:

$$A_{\mu,\nu} = N |a_\mu|^2 |a_\nu|^2 f_{\mu,\nu}, \tag{6.14}$$

where N is an overall normalization constant. By recursively applying (6.14) to all peaks in a complete series of $\Delta\nu$, the probabilities $|a_\nu|^2$ of the initial state distribution $\{a_\nu\}$ can

be reconstructed. This is shown in Figure 6.4, where for both initial distributions the first few state occupation probabilities have been retrieved from the first series with $\Delta\nu = 1$. For the normalization, one additional peak in the $\Delta\nu = 2$ series has been used, since the ratio of the pair occupation is not preserved by an overall normalization factor. Comparing reconstructed and the known initial vibrational state occupations, very good agreement is found for both Franck–Condon and ADK–based distributions, providing hope that the proposed imaging method might soon be utilized experimentally.

6.4 Probe Pulse

Apart from the study of the nodal structure of the bound vibrational wave functions and the retrieval of the vibrational state probability distribution, the method introduced above may also be found suitable for observing dynamical characteristics of the moving wave packet in a laser field, such as bond-softening[Bandrauk 81,Bucksbaum 90], bond-hardening[Frasinski 99] or above threshold dissociation[Giusti-Suzor 90,Staudte 07] within the Floquet picture[Barone 77]. However, as the Floquet approach is based on the periodicity of the electric field of the laser, it can only serve as an approximate the description of broad-band few-cycle pulses interacting with molecules. Both, pump and probe pulse for the Fourier imaging method need to be sufficiently short on the time scale of the nuclear motion in order to resolve the wave packet dynamics. Realistic ultra-short few-femtosecond pulses on the other hand almost unavoidably sit on a rather long pedestal – typically around 100 fs long at 5% of the peak intensity for 7 fs laser pulses[Légaré 05,Rudenko 06] – such that the Floquet picture appears valid for the motion of the nuclear wave packet directly following the pump pulse until the leading edge of the probe pulse peak.

In this section the influence of a pedestal at the pump and the probe pulse is investigated. Figure 6.5(a) shows a calculation for two 50 fs, 1×10^{13} W/cm^2 Gaussian pedestals – the first centered at $t = 0$ and the second directly preceding a probing pulse. The wave function is then analyzed at the center of the second pulse. In a series of pump – probe delay

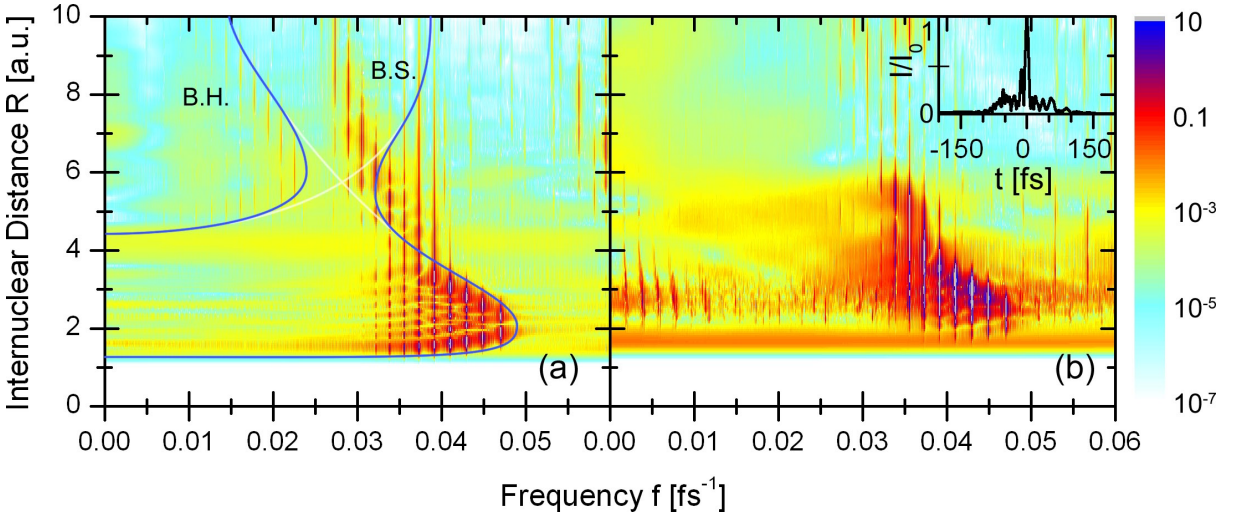

Figure 6.5: *Power spectrum as a function of the frequency f and the internuclear distance R. (a) Propagation of an initial Franck-Condon wave packet with the inclusion of a* 1×10^{13} W/cm^2, 50 fs *(FWHM) pedestal at the pump pulse and the probe pulse causing 'bond hardening' (B.H.) and 'bond softening' (B.S.). The wave packet is sampled at the center of the probe pedestal. (b) Initial ADK-populated wave packet after a Gaussian* 3×10^{14} W/cm^2, 6 fs *pump pulse. An actual laser pulse profile (see inset) is used for the propagation during the remainder of the pump pulse and for the Coulomb explosion during the probe pulse with a peak intensity of* 6×10^{14} W/cm^2.

calculations, the evolution of the vibrational wave packet at the center of a probe pulse (without the actual fs probe pulse) is obtained. For simplicity and in order to have higher vibrational states populated, an initial Franck-Condon distribution is assumed.

The figure shows several interesting features compared to field free calculations (c.f. Figure 6.2(c)): A significant probability density of the wave packet is seen in the classically forbidden region at large R for frequencies around $f \approx 0.035$ fs^{-1}. In addition, trapped states appear at large $R \approx 6$ and small energies $f \approx 0.02$ fs^{-1}. In the Floquet picture[Giusti-Suzor 90], the two field-free photon-energy shifted potential curves $1s\sigma_g$ and $2p\sigma_u$ combine to form the field-dressed adiabatic potential curves indicated in Figure 6.5(a). Bond softening is responsible for the dissociation of the higher vibrational states, seen as vertical lines extending to large internuclear separations in the avoided crossing region. As the electric field increases during the pulse, the gap between the adiabatic potential curves increases and allows for

the dissociation and tunnel-ionization of lower vibrational states. A second light-induced potential well causes the temporal trapping of a part of the wave function density, known as bond hardening. The nodal structure of the bond hardening states is visible in the figure and indicates the shape of the trapping potential well.

Equally interesting is the inclusion of a realistic laser pulse shape that has been obtained by the SPIDER imaging technique[Iaconis 98] for the experiment in Figure 6.2(d)[Feuerstein 07] and thus allows for the direct comparison with the numerical simulation. The intensity profile is shown in the inset of Figure 6.5(b) and exhibits a pedestal of approximately 100 fs as well as a pronounced pre-pulse before the main peak. For the calculation shown in Figure 6.5(b), the peak laser intensity of the pump pulse is 3×10^{14} W/cm^2 and 6×10^{14} W/cm^2 for the probe pulse, matching the parameters in the experiment. The Coulomb explosion during the probe pulse is modeled by the R-dependent ADK rates $\Gamma_g(R, E(t))$ and $\Gamma_u(R, E(t))$ (5.3) from the $1s\sigma_g$ and $2p\sigma_u$ channels and incoherently adding the probabilities

$$\begin{aligned}\rho_{\mathrm{CE}}(R) = &\int_{-\infty}^{\infty} \left(1 - \mathrm{e}^{-\Gamma_g(R,E(t))t}\right) |\chi_g(R,t)|^2 \mathrm{d}t \\ &+ \int_{-\infty}^{\infty} \left(1 - \mathrm{e}^{-\Gamma_u(R,E(t))t}\right) |\chi_u(R,t)|^2 \mathrm{d}t. \end{aligned} \tag{6.15}$$

This model neglects the kinetic energy of the wave packet at the time of the ionization, but includes the R-dependence of the Coulomb explosion. The spectrum in Figure 6.5(b) is then obtained by performing a time-series calculation with a variable probe pulse delay time t_D, and Fourier analyzing $\rho_{\mathrm{CE}}(R, t_D)$. The result agrees well with the experimental data shown in Figure 6.2(d). The vertical lines between $f = 0.03$ fs^{-1} and $f = 0.04$ fs^{-1} extending to large R are caused by the dissociation due to the pulse pedestal, while no evidence for the occurrence of bond hardening states is found for the given laser parameters. Also seen is that the nodal structure even for the lower bound vibrational states disappears. This blurring is the result of the finite probe pulse length of approximately 7 fs, such that the bound nuclear wave packet moves during the ionization of the laser pulse[Lin 06]. With the emerging

single-attosecond XUV laser pulses, it appears feasible that the motion of vibrational wave packets can soon be resolved with much greater detail.

6.5 Decoherence

In order to discuss a scheme for quantifying the degree of coherence in a nuclear vibrational wave packet, the present analysis is extended by allowing for an incoherent mixture $\{p_\nu, \chi_\nu\}$ of bound vibrationally excited states. The numbers $p_\nu = |a_\nu|^2$ represent the probabilities for finding a given stationary vibrational state χ_ν in the statistical mixture. The corresponding density operator

$$\rho^{\text{inc.}} = \sum_\nu p_\nu |\chi_\nu\rangle\langle\chi_\nu| \tag{6.16}$$

is represented by the diagonal matrix $\rho^{\text{inc.}}_{\mu,\nu} = p_\nu \delta_{\mu,\nu}$ relative to the basis $\{\chi_\nu\}$. Without external interactions, $\rho^{\text{inc.}}$ is time independent. Denoting the time–dependent density matrix of the pure state (6.1) as $\rho^{\text{coh.}}(t) = |\Psi(t)\rangle\langle\Psi(t)|$ results in the relation

$$\rho^{\text{coh.}}_{\mu,\nu} = a^*_\mu a_\nu \mathrm{e}^{-i\Delta\omega_{\mu,\nu}t} \tag{6.17}$$

$$= \rho^{\text{inc.}}_{\mu,\nu} + \rho^{\text{coh.}}_{\mu,\nu}(1 - \delta_{\mu,\nu}). \tag{6.18}$$

In contrast to $\rho^{\text{inc.}}$, $\rho^{\text{coh.}}$ is not diagonal in the $\{\chi_\nu\}$ basis and includes time–dependent off–diagonal elements. The degree of (de–)coherence is given by the relative importance of these off-diagonal elements.

In terms of density matrix elements, the probability density can be written as a sum of an incoherent (time–independent) and a coherent (time–dependent) term[Feuerstein 03b],

$$|\Psi(R,t)|^2 = \sum_{\mu,\nu} \rho^{\text{coh.}}(t)_{\mu,\nu}\chi_\mu(R)\chi_\nu(R) \tag{6.19}$$

$$= \sum_\nu p_\nu |\chi_\nu(R)|^2 + \sum_{\mu\neq\nu} \rho^{\text{coh.}}_{\mu,\nu}(t)\chi_\mu(R)\chi_\nu(R). \tag{6.20}$$

In the harmonic analysis the incoherent contribution at $\omega \approx 0$ is imaged as a diffuse background over the range of classically allowed internuclear distances and is subtracted in the

definition $c(R,t)$ in (6.5). This separation of coherent and incohernet contributions in (6.20) suggests that the continuous loss of coherence of the vibrational wave packet due to weak randomly fluctuating external forces can be tested by a sequence of finite–time Fourier transformations (6.7) for a series of increasing sampling times T_i.

More specifically, by modelling decoherence in terms of an additional random phase factor in each expansion coefficient of the wave packet, i.e., by replacing the set of complex numbers $\{a_\nu\}$ that characterize the free wave packet by $\{|a_\nu|\exp(i\alpha_\nu(t)\}$ with random phases $\alpha_\nu(t)$, the harmonic analysis of (6.20) for finite T results in

$$\tilde{c}(R,\omega;T) \equiv \frac{1}{\sqrt{2\pi}}\int_0^T |\Psi(R,t)|^2 \mathrm{e}^{-i\omega t}\mathrm{d}t \tag{6.21}$$

$$= \frac{1}{\sqrt{2\pi}}\left(2\pi\delta_T(\omega)\sum_\nu p_\nu|\chi_\nu(R)|^2 + \sum_{\mu\neq\nu}\chi_\mu(R)\chi_\nu(R)\int_0^T a_\mu^*(t)a_\nu(t)\mathrm{e}^{i(\omega_{\mu,\nu}-\omega)t}\mathrm{d}t\right). \tag{6.22}$$

The coherent contributions in (6.20) tend to cancel in the average over large time intervals $[0,T]$. Thus, for increasing sampling times, lines in $\tilde{c}(R,\omega;T)$ at frequencies $\omega_{\mu,\nu} \neq 0$ fade away. Eventually, as T becomes much larger than the typical decoherence time T_d of the molecules in their environment, only the incoherent contribution at $\omega = 0$,

$$\tilde{c}(R,\omega;T \gg T_d) \rightarrow \sqrt{2\pi}\delta(\omega)\sum_\nu p_\nu|\chi_{\nu'}(R)|^2, \tag{6.23}$$

remains. The time series of spectra $\{\tilde{c}(R,\omega,T_i) \mid T_1 < T_2 < \ldots\}$, obtained by measuring the kinetic energy release over a range of pump–probe delays of the order of T_d thus constitutes a means to quantify the degree of (de–)coherence in the nuclear motion.

«Observations always involve theory. »
Edwin Hubble

Chapter 7

Resonant Neutralization of H^- Anions in front of Metal Surfaces

7.1 Motivation

In this last application of the lattice based solution of the time–dependent Schrödinger equation for a single active particle, the charge–transfer process in ion–surface scattering is investigated. While describing the surface of a metal as a one–electron potential is a significant simplification of the electronic structure, a comparison with experimental data will show the validity of the use of the pseudo–potential. Moreover, it shows that the Crank–Nicholson propagation method (Chapter 2) can be successfully applied to larger and more complex systems as well.

The charge–transfer process on surfaces is of fundamental importance, as it determines the final charge state of the scattered particle and plays a prominent role in the chemical reaction dynamics. Possible applications reach from control of plasma–wall interactions in ion–sources, surface chemistry and analysis, reactive ion etching, or semiconductor miniaturization via thin film deposition[Gauyacq 96,Shao 94,Los 90]. For surface scattering with atomic anions, the Fermi–energy of the metal is typically below the affinity level of the ion such that the transfer dynamics is dominated by the single active–electron transitions over other

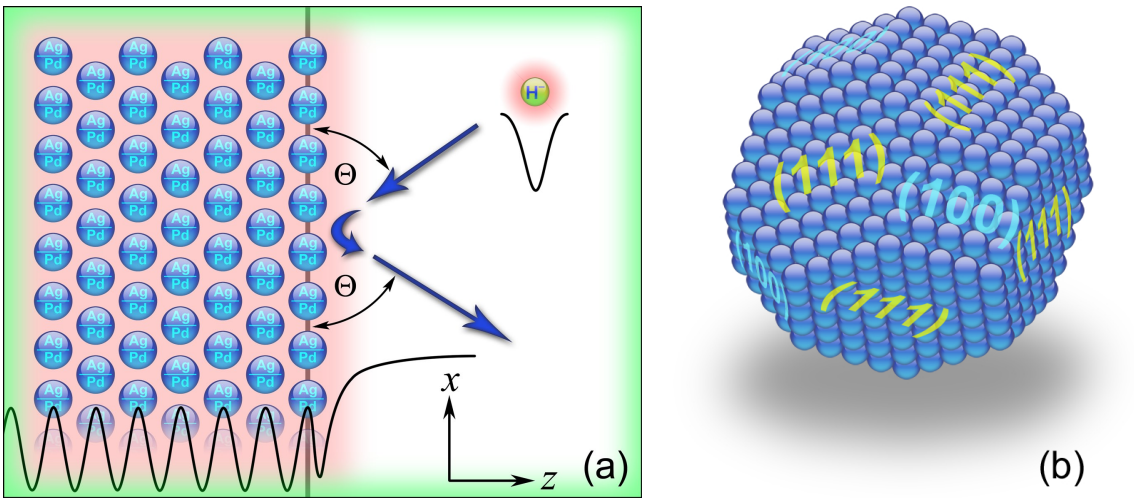

Figure 7.1: *(a) Schematic illustration of the neutralization of the H^- anion in–front of a metal surface. The anion follows a classical trajectory and allows for the interaction of the loosely bound single active electron with the band structure of the metal. The electronic states of the metal are filled up to the Fermi level and are excluded in our calculations. Also shown is a sketch of the surface and ion potential. Absorbing boundaries (green) (see Section 2.8) are used for suppression of reflections on the grid edges. (b) Two different surface morphologies for face–centered metals are considered in this chapter: the (111) and the (100) cut of the surface. The geometry of the different cut can be shown by the facets of a truncated octahedron, where the surface atoms for the (111) side form an equilateral triangle, whereas for the (100) the atoms are aligned in a square lattice.*

charge–transfer processes[Los 90].

In this comparative study, the role of surface and image states on the ion neutralization can be illustrated by the scattering of H^- anions on different metal surfaces in terms of a single–electron transfer, due to tunneling of the bound electron through the potential barrier formed by the ion and the bulk–vacuum interface (Figure 7.1 (a)). This results in either ion–survival of the surface scattered anion or neutralization, known as resonant charge transfer (RCT).

Because of its importance, RCT has been the focus of several experimental and theoretical studies[Los 90,Maazouz 97,Kürpick 97,Kürpick 98,Sanchez 99,Hecht 00,Guillemot 99,Thumm 00,Yang 02]. It has been predicted that for projectile states within the projected band gap of a metal, the tunneling rate along the surface normal, which is the preferred direction of RCT, is sig-

nificantly reduced[Borisov 98,Borisov 99,Chakraborty 05] and was subsequently confirmed experimentally[Guillemot 99,Hecht 00,Wethekam 03].

Neutralization and ion–survival is strongly influenced by surface properties such as the projected band gap and the relative positions of surface and image states and the conduction band. Different surface cuts of a metal, resulting in altered surface electronic structures, therefore lead to different neutralization dynamics. For example, the atoms in the (100) cut of a surface are located in a square lattice arrangement, whereas the (111) surface exhibits a hexagonal structure with a $\approx 15\%$ larger layer spacing in direction normal to the surface (see Figure 7.1(b)). In this study, Silver (111), Palladium(111) and Palladium (100) surfaces are good candidates to address the role of image[Chakraborty 05] and surface states[Borisov 99] on the RCT process, because:

- The Fermi–energy of all three surfaces remains below the H^- anion affinity level for sufficiently small collision energies,
- The Ag(111) surface state is localized within the band gap, but the image states are degenerate with the conduction band,
- The Pd(111) surface offers localized surface state and image states within the band gap, and
- The Pd(100) surface state is embedded in the valence band, but the image states are localized in the band gap.

7.2 Surface Potential

The surface of the metal is modeled by a one–dimensional semi–empirical single–electron effective potential, constructed from a self-consistent pseudo-potential local–density approximation, and adjusted to reproduce experimental and first principle values of the band edges,

surface states and the image states[Chulkov 99]:

$$V_{\text{surf}}(z) = V_1(z) + V_2(z) + V_3(z) + V_4(z), \tag{7.1a}$$

$$V_1(z) = A_{10} + A_1 \cos\left(\frac{2\pi}{a_s} z\right) \qquad \text{for } z < 0, \tag{7.1b}$$

$$V_2(z) = -A_{20} + A_2 \cos(\beta z) \qquad \text{for } 0 < z < z_1, \tag{7.1c}$$

$$V_3(z) = A_3 e^{-\alpha(z-z_1)} \qquad \text{for } z_1 < z < z_{\text{im}}, \tag{7.1d}$$

$$V_4(z) = 27.21 \text{ eV} \frac{e^{-\lambda(z-z_{\text{im}})} - 1}{4(z - z_{\text{im}})} \qquad \text{for } z_{\text{im}} < z. \tag{7.1e}$$

The periodic part $V_1(z)$ (7.1b) describes the atomic layers of the bulk metal, spaced by the lattice constant a_S along the z-coordinate normal to the surface. The second contribution $V_2(z)$ (7.1c) produces an additional binding potential well in the vacuum side, $z > 0$, to model the existence of a localized surface state. The last term $V_4(z)$ (7.1e) reproduces the correct $(4z)^{-1}$ attractive image potential, derived from the polarization of a conducting surface, while $V_3(z)$ (7.1d) smoothly connects the surface and image potential contributions V_2 and V_4, respectively. The topmost layer of lattice points defines $z = 0$. The eleven parameters in (7.1) are not all independent: only five parameters are required to reproduce the shape of the potential, given in Table 7.1. The remaining six parameters A_{20}, A_3, α, λ, z_1, and the position of the image plane z_{im} are obtained by enforcing a continuous logarithmic derivative of the potential at the matching points. A free electronic motion is assumed in direction parallel to the surface in the metal continuum $V_{\text{surf}}(\mathbf{r}) = V_{\text{surf}}(z)$ (see Figure 7.3 (b)). The resulting band structure for a free electronic motion parallel to the surface leads to the dispersion curves shown in Figure 7.2 and numerical values are given in Table 7.2.

7.3 Ion Potential

The potential of the negative H^--ion is described by a spherical symmetric effective single–electron potential (Figure 7.3(a)), modeling the interaction of the loosely–bound active

	A_{10} (eV)	A_1 (eV)	A_2 (eV)	a_s (a_0)	β (a_0^{-1})
Ag(111)	−9.640	4.30	3.8442	4.430	2.5649
Pd(100)	−8.480	7.10	5.0481	3.680	1.8460
Pd(111)	−8.570	6.60	4.6728	4.250	1.9071

Table 7.1: *Set of five independent parameters in (7.1) used to describe the three surface potentials*[Chulkov 99].

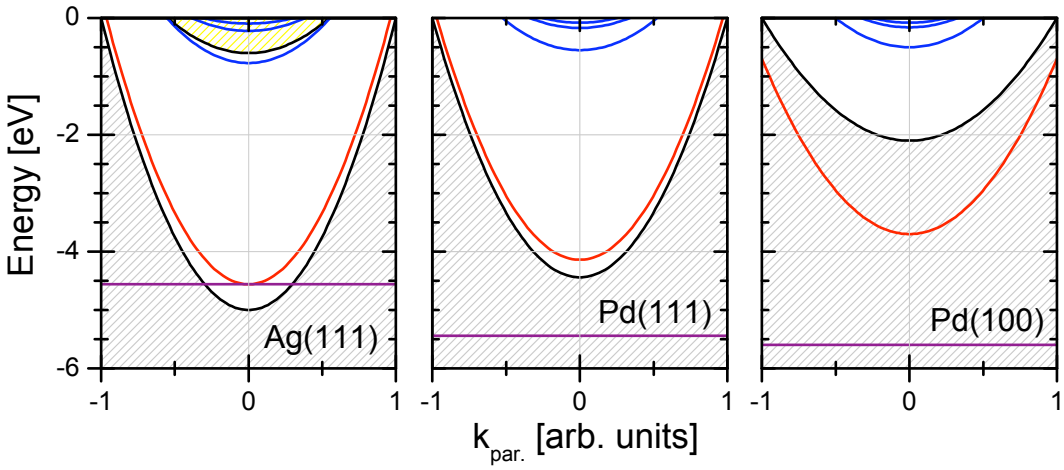

Figure 7.2: *Schematic of the electronic band structure of the Ag(111), Pd(111), and Pd(100) surfaces as a function of the parallel momentum $k_{\text{par.}}$. The band gap is delimited by the valence band (grey shaded) and the conduction band (yellow shaded). The conduction band for the Pd surfaces is above the energy $E = 0$. Also shown are the surface state band (red) and the lowest three image state bands (blue) as well as the Fermi–energy (purple).*

	Ag(111)	Pd(100)	Pd(111)
Fermi–energy	−4.56	−5.60	−5.44
Valence band top	−4.96	−2.10	−4.44
Conduction band bottom	−0.66	5.00	2.16
Surface state	−4.63	−3.70	−4.14
First image state	−0.77	−0.50	−0.55
Second image state	−0.22	−0.16	−0.17
Third image state	−0.10	−0.08	−0.08
Image plane z_{im}	2.22	2.29	2.29

Table 7.2: *Electronic band structure of the three considered surfaces for zero parallel momentum. All energies are given in electron volts (eV). Also shown is the position of the image plane z_{im} in a.u. obtained from the model potential*[Chulkov 99].

electron with a polarizable core[Cohen 86]:

$$\begin{aligned} V_{H^-}(r) &= -V_{\text{core}}(r) + V_{\text{pol}}(r) \\ &= -\left(1+\frac{1}{r}\right)\mathrm{e}^{-2r} - \frac{1}{2}\frac{\alpha_H}{r^4}\mathrm{e}^{-r_0^2/r^2}. \end{aligned} \tag{7.2}$$

The polarizability of the hydrogen core is $\alpha_H = \frac{9}{2}a_0^3$ and the parameter $r_0^2 = 2.547a_0^2$. Since the numerical accuracy rapidly decreases at the singularity for $r \to 0$, the potential V_{H^-} has been regularized[Ermoshin 96,Borisov 99]:

$$V_{H^-}^{\text{reg.}}(r) = \frac{\gamma V_{H^-}(r)}{\sqrt{\mu^2 V_{H^-}(r)^2 + 1}}, \tag{7.3}$$

On a three–dimensional grid, the parameters $\gamma = 1.107$ and $\mu = 0.34$ reproduce the electron affinity level of 0.76 eV[Ermoshin 96]. However, in order to reduce the computational effort, a two–dimensionality model is adopted (see also Chapter 3) and the electronic motion is limited to the scattering plane (Figure 7.1). The stronger potential binding in the reduced dimensionality model is accounted for by reparametrizing (7.3) $\gamma_{2\text{D}} = 0.3923$ and $\mu_{2\text{D}} = 0.3764$ to ensure the correct electron affinity level[Chakraborty 04a,Chakraborty 04b], which is calculated by imaginary time propagation (Section 2.6).

The motion of the anion is assumed to be classical along a trajectory $\mathbf{R}(t)$, that is given by the net repulsive force between the surface and the ionic core. In particular, the modeling is based on the *Biersack–Ziegler* repulsive interaction $V_{\text{BZ}}(z)$[Biersack 82], that incorporates a fitted pairwise potential, and plane averaging over the topmost layer of surface atoms. The trajectory of the incident ion is then characterized by the kinetic energy E and the angle with the surface plane Θ (Figure 7.1). The motion in parallel direction is of constant velocity

$$v_{\text{par.}} = \sqrt{\frac{2E}{m}}\cos\Theta. \tag{7.4}$$

For the motion in normal direction $v_{\text{nor.}}$, the ion decelerates until a distance of closest approach D_{cls} and gets reflected with the exit angle $\Theta_{\text{out}} = \Theta_{\text{in}}$. Consequently, the effect of a variable grazing angle Θ at a fixed projectile energy is twofold: As Θ increases to 90° (normal incidence), the distance of closest approach decreases and the ion "probes" a

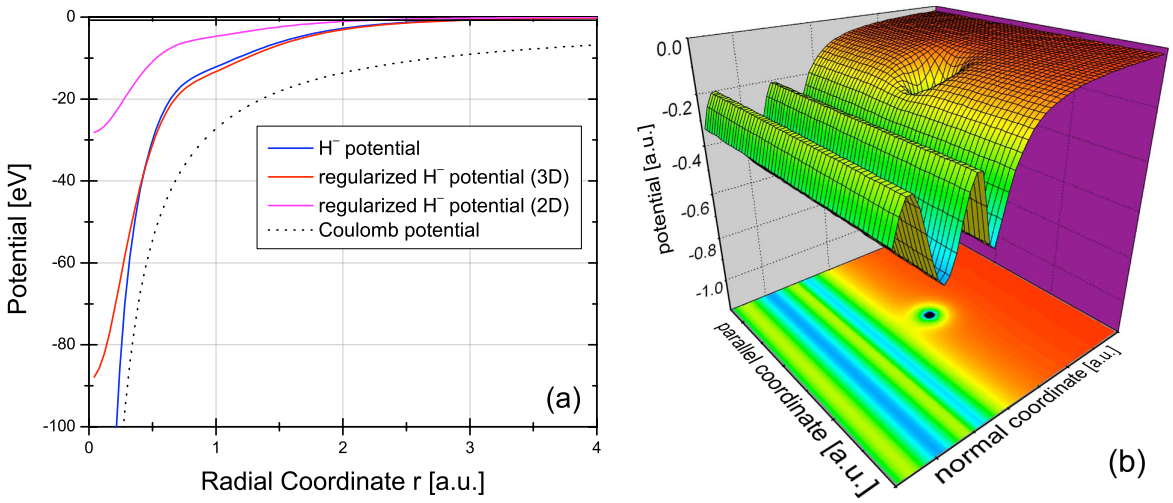

Figure 7.3: *Electronic potential of the H^- anion (a), modeling the ionic core and polarization interaction. A regularized version (red) avoids the singularity at the origin (blue). Due to the stronger potential binding in reduced dimensionality calculations (cf. Chapter 3), a reparametrized regularization leads to a higher potential curve (magenta) in order to obtain a binding energy of* 0.76 eV *(black line). The Coulomb potential is also shown for comparison. The total potential for the single active electron includes the surface and the H^- anion interaction (b).*

larger range of ion – surface distances. Simultaneously, the time spend in the vicinity of the surface decreases as $v_{\text{nor.}}$ becomes larger, and the ion has less time to explore the electronic configuration of the surface.

7.4 Propagation

For the time propagation on a two–dimensional grid the Crank–Nicholson scheme (Chapter 2) is employed with an initial free H^- electronic wave function $\phi_{\text{ion}}(\mathbf{r})$. The electronic wave function $\Phi(\mathbf{r}, t)$ is the solution of the time–dependent Schrödinger equation with the Hamiltonian (in atomic units):

$$\hat{H} = \frac{\hat{\mathbf{p}}^2}{2} + V_{\text{surf}} + V_{H^-}^{\text{reg.}}(\mathbf{R}(t)) + V_{\text{abs.}}, \tag{7.5}$$

containing the surface potential V_{surf} (7.1), the regularized potential $V_{H^-}^{\text{reg.}}$ of the moving H^- anion (7.3), and an absorbing potential $V_{\text{abs.}}$ at the grid boundaries in order to avoid

reflections (see Section 2.8). Converged results were obtained with quadratic absorbers of 100 a.u. width in x– and positive z–direction. In negative z–direction (bulk side of the metal), a cubic absorber of 15 a.u. width was found sufficient[Chakraborty 04b].

The initial wave function $\phi_{\text{ion}}(\mathbf{r})$ is calculated with the imaginary time propagation (Section 2.6) in the $V_{H^-}^{\text{reg.}}$ potential, and multiplying the wave function with the proper phase factor, $\exp\left(-i(\mathbf{v}\cdot\mathbf{r} + v^2 t/2)\right)$, corresponding to the Galilean transformation due to the moving projectile.

The ion survival amplitude is calculated for every time step by means of the overlap with the initial wave function of the ion (autocorrelation function):

$$A(t) = \langle \Phi(\mathbf{r}, t | \phi_{\text{ion}}(\mathbf{r}) \rangle \,, \tag{7.6}$$

such that the survival probability is given by

$$P_{\text{surv.}}^{(2\text{D})}(E, \Theta) = \lim_{t\to\infty} |A(t)|^2. \tag{7.7}$$

Here, the upper index refers to the reduced dimensionality of the calculation. The full three–dimensional survival probability can be approximated from the adiabatic transition rates $\Gamma(d)$ at fixed distance d to the surface and integrating over the projectile trajectory:

$$P_{\text{adiabatic}}^{(3\text{D})} = \exp\left(- \int_{-\infty}^{\infty} \Gamma^{(3\text{D})}(\mathbf{R}(t))\, \mathrm{d}t \right) , \tag{7.8}$$

where $\Gamma^{(3\text{D})}$ for a two–dimensional surface continuum consists of equal rates for the decay in x and y direction,

$$\Gamma^{(3D)}(d) = \Gamma_x(d) + \Gamma_y(d) + \Gamma_z(d) \approx 2\Gamma^{(2D)}(d). \tag{7.9}$$

assuming Γ_z for the RCT along the surface normal is negligible, as long as the projected band gap is blocking the transfer for zero parallel momentum[Borisov 98,Borisov 99]. It follows that the decay rates in the exponent of (7.8) are doubled for a two–dimensional over the one–dimensional surface continuum; thus the full three–dimensional ion survival rate can be approximated by[Chakraborty 04a]:

$$P_{\text{surv.}}^{(3\text{D})}(E, \Theta) \approx \left| P_{\text{surv.}}^{(2\text{D})}(E, \Theta) \right|^2 . \tag{7.10}$$

7.5 Results

The ion survival probability $P_{\text{surv.}}^{(3D)}(E, \Theta)$ for 1 keV H^- ions as a function of the scattering angle Θ with the surface is shown in Figure 7.4. Comparing with the measured ion–fractions by Guillemot and Esaulov[Guillemot 99] for the scattering on Ag(111) surfaces, the present calculation agrees well with the experimental data for scattering angles $\Theta \geq 30°$, thereby confirming the estimate (7.9) of doubling the decay rates into the surface continuum to account for the reduced dimensionality. For smaller angles, this is when the velocity parallel to the surface and the time of the projectile spend within the vicinity of the surface increases, the agreement worsens. This might be addressed to the following limiting assumptions in the calculations:

- As the model potential (7.1), adjusted to the band structure of the metal at $k_{\text{par.}} = 0$, shows corrugation only along the surface normal, the motion of the electron in parallel direction follows the free dispersion curves shown in Figure 7.2. In reality, the dynamics along the surface is not translational invariant, and the motion of the electron parallel to the surface is associated with an effective mass[Borisov 99].

- The distance of closest approach and the time the ion spends near the surface strongly depends on the shape of the Biersack–Ziegler potential[Biersack 82] used to model the ion trajectory, and deviations are most pronounced for small angles Θ.

Comparing the survival probabilities for the H^- ion scattered from the Ag(111), Pd(111) and Pd(100) surfaces with an incident kinetic energy of 1 keV at normal incidence $\Theta = 90°$ shows: Both (111) surfaces, sharing a similar localized surface state within the band gap, have almost identical survival probabilities for the ion, despite their dissimilar image states. By contrast, the Pd(100) surface with the surface states embedded in the valence band exhibits a much larger neutralization (cf. Figure 7.2). Thus, for short interaction times with the surface, neutralization is dominated by the character of the surface state. In particular, as the surface state of Pd(100) overlaps with bulk states, it provides an

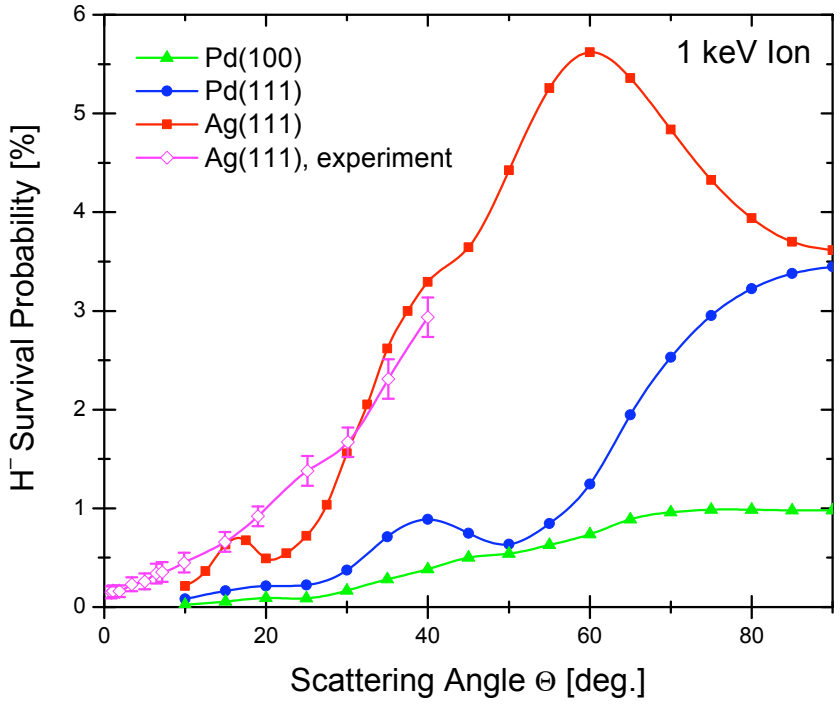

Figure 7.4: *Ion survival probability $P^{(3D)}_{surv.}(E,\Theta)$ for* 1 keV *H^- projectiles scattered from Pd(100), Pd(111) and Ag(111) surfaces with varying angles of incidence Θ with respect to the surface plane. For comparison, measured values on Ag(111) by Guillemot and Esaulov*[Guillemot 99] *are shown as well.*

effective decay channel through the valence band of the metal. This is in contrast to the localized surface states for the (111) metal, that can only decay in parallel direction to the surface[Borisov 99,Chakraborty 04a,Chakraborty 04b].

Clearly, as the angle with the surface plane Θ decreases, the interaction time of the ion with the surface increases and so does the charge neutralization probability of the ion, leading to a reduced survival of the ion on all three considered surfaces. For incident angles $\Theta < 60°$, both of the palladium surfaces show a similar ion survival probability, while for Ag(111) the ion survival is largely enhanced and features a pronounced maximum at $\Theta = 60°$. Since Ag(111) and Pd(111) share similar surface states, the differences can be attributed to their distinct image states. The palladium surfaces possess the same Rydberg series of localized image states, that reach far into the vacuum region and thus evolve away from the surface when populated[Chulkov 99,Chakraborty 04a]. In contrast, the image states of the

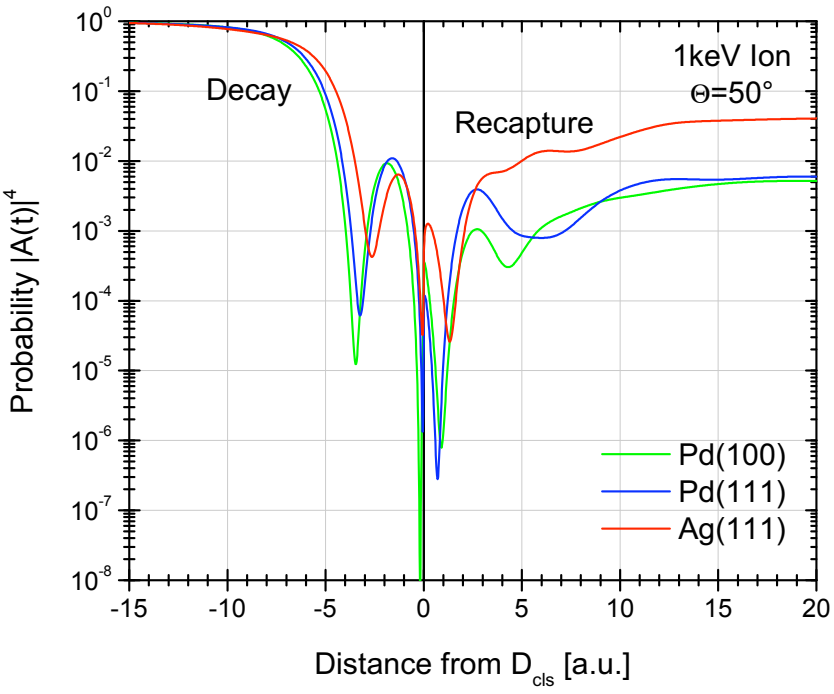

Figure 7.5: *Time evolution of the population of the ion $|A(t)|^4$ as a function of the distance from the turning point D_{cls} of the trajectory in logarithmic scale. The incident beam energy is* 1 keV *at an scattering angle of* 50° *with the surface. For negative distances, corresponding to an incident ion, the population of the ion state readily decreases due to the decay in image and surface states. For the outgoing ion trajectory at positive distances, the interaction with the image states leads to recapture.*

silver surface (except for the lowest one) are coupled with the conduction band and remain localized at the surface. Consequently, the probability for recapture of the electron on the outgoing trajectory of the ion is enhanced for the Ag(111) surface, due to the proximity of the electron cloud at the on the vaccum–surface interface.

This is further illustrated in Figure 7.5, where the evolution of the population $|A(t)|^4$ (c.f. (7.10)) of the ionic state is shown at an incident trajectory of $\Theta = 50°$. At "negative" distances, as the ion approaches the turning point D_{cls}, the ionic state firstly interacts with the surface states of the metal, and, at around 4 a.u. before the distance of closest approach also with the image states. A signature of the latter is seen in the oscillations of $|A(t)|^4$, as the populated image states reach into the vacuum side and spatially overlap with the projectile. Until the turning point of the trajectory, the resulting decay through surface and

image state interactions is very similar for all three surfaces. Therefore, after the turning point, it becomes evident that the final charge state of the projectile is determined by the degree of electron recapture from the surface on the outgoing part of the trajectory.

Lastly, the dynamics of the electron recapture can be seen directly in the evolution of the wave function probability density shown in Figure 7.6 for both (111) surfaces. For the scattering at $\Theta = 50°$, the three snapshots illustrate the three processes of surface state interaction, image state interaction and recapture. At a time of 100 a.u. before the turning point of the trajectory, the ion interacts and populates the localized surface state. At 50 a.u. after the turning point, the surface states are seen to only decay laterally due to the band gap of the metal, while a diffuse electron probability cloud is forming on the vacuum side. The difference in the image state interaction becomes obvious at 200 a.u. after the closest encounter with the surface: For the palladium surface the electron probability density has moved to large distances into the vacuum away from the surface due to the population of image states that are energetically degenerate with the band gap and thus prohibit the decay along the surface normal. For Ag(111) on the other hand, the coupling between the image states and the unoccupied part of the conduction band has left the surface interface near the projectile with a much larger density distribution, thereby again suggesting a larger rate of electron recapture on the outgoing part of the trajectory.

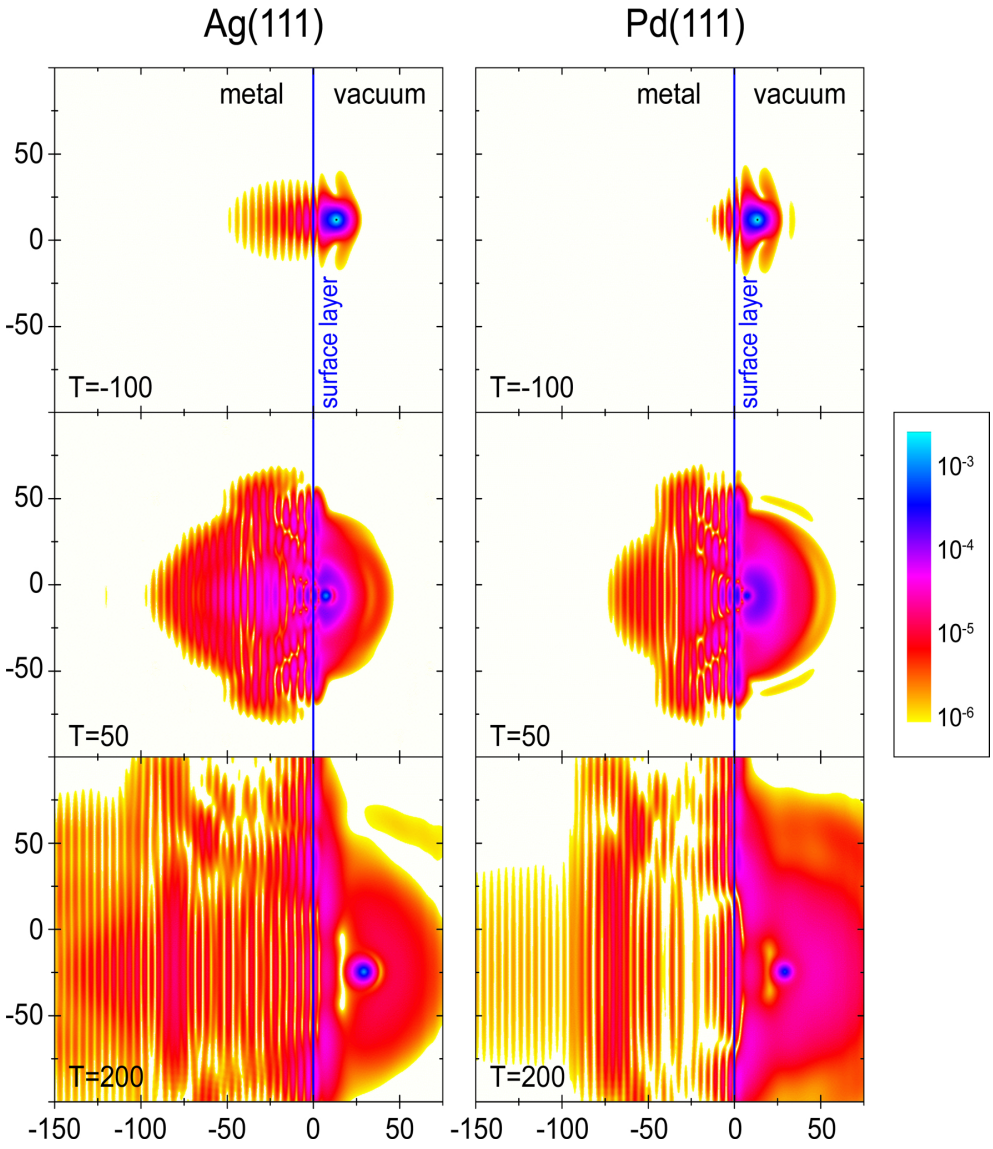

Figure 7.6: *Time evolution of the electronic probability density in logarithmic scale. Three snapshots are shown for the Ag(111) and Pd(111) surface at a time* 100 a.u. *before,* 50 a.u. *after, and* 200 a.u. *after the projectile reaches the turning point. The ion approaches at an energy of* 1 keV *and an angle of* 50° *to the surface.*

Chapter 8

Conclusions and Outlook

«Thou, nature, art my goddess; to thy laws my services are bound. »
Carl F. Gauss

In the first application of the wave packet propagation scheme the effects of a strong laser field on the dynamics of electron capture and emission in ion–atom collisions has been investigated within a reduced dimensionality model of the scattering system (Chapter 3). In order to reduce the numerical effort, the motion of the active electron and the laser electric field vector were confined to the scattering plane. Significant differences in the electron capture probabilities $P_{cap}^{\pm}(b)$ in co– and counterrotating laser–assisted collisions between a proton and a hydrogen atom were observed. They are caused by a resonant transition at matching energy levels between the laser field distorted Coulomb potentials of the projectile and target nuclei at relevant impact parameters. Both, capture and electron emission probabilities differ significantly from results for laser–unassisted collisions. A strong circular dichroism is predicted, i.e., the capture (and to a lesser extent ionization) probabilities are different for parallel and anti–parallel laser helicity and projectile angular momenta.

Laser pulses with lengths of a few nano–seconds and intensities of about 5×10^{12} W/cm^2 and higher should allow for the experimental verification of the predicted dichroism in the capture probability. In addition, evidence was seen for the charge resonant enhanced ionization mechanism in laser–assisted ionization. This effect may be used in angle–differential laser–assisted collision experiments in order to select a specific orientation of the laser electric field at the time of closest approach between projectile and target and would allow for

the detection of the laser collision phase.

Following the encouraging predictions from the reduced–dimensionality model for laser–assisted collisions, an *ab initio* calculation, solving the full three–dimensional time–dependent Schrödinger equation was presented in Chapter 4. Both, electron capture and ionization probabilities for ion–atom collisions were computed for a laser intensity of 5×10^{13} W/cm^2 where the most promising dichroism was found in the reduced dimensionality model. For circularly polarized laser fields and an impact energy of 1.2 keV/amu, a substantial modification of the electronic dynamics in the proton–hydrogen collision system as compared to field–free collisions was observed. In particular, the strong dependence on the initial laser collision phase and the impact parameter for both capture and ionization, which can be explained using semi–classical arguments, agrees well with the predictions of the previous two–dimensional model calculations, for the special case that the collision and the laser plane coincide. The inclusion of "off–plane" contributions allows for the calculation of total cross sections for electron capture and shows a noticeable dichroism remaining in the integrated capture probabilities after averaging over the laser collision phase.

Although other theoretical studies on laser–assisted heavy particle collisions have suggested novel pathways to steer the charge transfer reaction, the experimental verification remains rather challenging. Yet, the presently available technology would in general permit such experiments. Research on pico–pulsed particle beams and strong lasers has emerged in AMO facilities. More theoretical work is necessary to find more suitable collision reactions, where less intense lasers fields and experimentally better available reactants are involved. In particular, the atomic hydrogen target has been appealing for the theoretical investigation, but is fairly delicate to handle in experimental studies.

Another course to control the quantum dynamical motion has been shown in Chapter 5 for the nuclear vibration of small homonuclear molecules. Good agreement was found within the present model that treats the nuclear motion on two coupled one–dimensional Born-Oppenheimer potential curves, for the position of the wave function revival times. The

addition of one or more short and intense control laser pulses can significantly alter the vibrational state composition of nuclear wave packets in D_2^+ by coherent Raman transitions. For the particular parameters in the current model calculations it is shown that a single control pulse can populate a selected lower vibrational state of D_2^+ with 50% probability, and that this fraction can be increased to over 75% by adding a second identical control pulse at an appropriate delay time. Such almost stationary nuclear wave packets should be observable in experimental studies by subsequent fragmentation of the molecular ion with a probe pulse, providing an unique tool of assessing the degree at which the nuclear motion in small molecules can be controlled. Creating a quantum system in a particular state with the help of precisely timed control laser pulses can possibly be applied to store information or to prepare a well defined initial state by "freezing" the nuclear motion for subsequent studies.

So far only two equal control pulses have been considered. The number of adjustable parameters (pulse length, intensity, wavelength) grows quickly with every additional control pulse, and a general control scheme for arbitrarily shaped control pulses is desirable. Genetic algorithms could offer a way to optimize such a multi–control design and exhibit exceptional control over the vibrational dynamics in a "time–gating" procedure.

In Chapter 6 the time–dependent nuclear motion of vibrational wave packets in the D_2^+ molecular ion induced by the ionization of its neutral D_2 parent molecule by an intense pump laser pulse is investigated by harmonic time series analysis. The Fourier-transform resembles the shape of the binding adiabatic nuclear potential curve and exhibits clear lines corresponding to difference energies between the vibrational states. This is due to pair-wise interference exhibiting the nodal structure of both vibrational wave functions. As an application for the R–dependent Fourier–time analysis, the initial vibrational distribution has been reconstructed and the progression of decoherence, induced by weak random interactions with the environment has been discussed. Extensions of this method indicate the possibility visualize other molecular processes in the energy domain, such as temporary

bond formation or bond breaking.

First experimental evidence for the applicability of the imaging scheme shows clear interference lines between neighboring vibrational states and their energetic position agrees well with the current work. A significant increase in experimental resolution can be obtained by utilizing attosecond pump and probe pulses such that the vibrational motion can be studied in much greater detail in the frequency domain. In addition, it appears feasible to apply the technique to more complicated molecules such as trimers or poly–atomic complexes, including the study of field induced effects, thus providing a novel scheme to visualize reaction dynamics.

The final Chapter 7 considers the resonant charge transfer in ion – surface scattering of negative H^- anions at three different metal surfaces – Ag(111), Pd(111) and Pd(100) – exhibiting distinctly different surface electronic band structures. Several different mechanisms for the resonant charge transfers reactions were found:

- As the surface state is populated first by the incident projectile, the existence of an embedded surface state within the valence band (as for Pd(100) and other (100) surfaces) leads to an effective decay of the electron along the surface normal into the bulk of the metal.

- Isolated surface states inside the band gap of the metal only allow for the decay in lateral direction and tend to reduce the ion neutralization.

- For normal incidence of the projectile, i.e. when the interaction time is short, the charge transfer is governed by the surface state interaction

- Image state wave packets quickly evolve into the vacuum region when occupied, thereby preventing recapture of the ion on the outgoing trajectory, and

- Image states within the conduction band couple with the unoccupied bulk states, and are thereby obstructed from freely evolving into the vacuum. Since the image states retain their proximity to the surface, this facilitates the recapture by the projectile.

The inclusion of many–electron effects from the occupied valence band of the metal, corrugation of the potential in parallel direction of the surface as well as nano–structured surfaces are currently invesigated[Obreshkov 06,Obreshkov 07] within a different (density functional) framework.

In conclusion, the utilized grid method for solving an up to three–dimensional Schrödinger equation on a numerical grid has provided a suitable ground for investigating several distinct scenarios in atomic, molecular, optical and surface physics, reaching from laser–assisted heavy particle collisions, laser–molecule interactions to particle – surface scattering. With the still unsurpassed empirical Moore's law, stating that the number of transistors on a chip doubles in about every two years, computer performance and storage capacities kept growing exponentially over the past four decades, and it is anticipated that more complex scenarios in more than three dimensions can soon be performed with the presented grid method. Of course, as the dimensionality grows, new challenges such as visualizing the wave functions will emerge. Another utilization is the refinement of the presented calculations since the studies of exploring a larger part of the parameter range are prohibitively expensive on the current hardware generation.

Appendix A

Atomic Units

In atomic physics it is common to use *atomic units* which are based on the typical scales found in atomic systems and are chosen such that many constants become unity. Therefore the Coulomb–law or the Schrödinger equations become more simplified in atomic units. One notable feature is the convention, that all units in this system are given in *a.u.*, i.e. the length, velocity, mass or time all carry the same dimensions. Thus it is essential to clearly specify the type of a unit (length, velocity etc.).

Since the international SI unit system is based on four constants of nature, the *meter*, *kilogram*, *second* and the *ampère*, one can choose four other reference units to be unity:

$$\begin{aligned} \hbar &= 1 && \text{(atomic unit of action)} \\ m_e &= 1 && \text{(atomic unit of mass)} \\ e &= 1 && \text{(atomic unit of charge)} \\ \frac{1}{4\pi\epsilon_0} &= 1 && \text{(atomic unit of the Coulomb force constant)} \end{aligned}$$

Coulomb's inverse–square law giving the magnitude and direction of the electrostatic force between two electrically charged objects of charge Z_1 and Z_2 then reads as

$$\begin{aligned} \mathbf{F} &= \frac{e^2}{4\pi\epsilon_0}\frac{Z_1 Z_2}{|\mathbf{r_1}-\mathbf{r_2}|^2}\,\mathbf{e}_r \qquad \text{(SI system)} \\ \mathbf{F} &= \frac{Z_1 Z_2}{|\mathbf{r_1}-\mathbf{r_2}|^2}\,\mathbf{e}_r \qquad \text{(in atomic units)} \end{aligned}$$

With the mass of the electron and the unit of action being unity in atomic units, the time–dependent Schrödinger equation for the motion of an electron also simplifies to

$$i\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \Big(-\frac{\nabla^2}{2} + V\Big)\Psi(\mathbf{r},t) \tag{A.1}$$

Typical conversions between the SI system and atomic units[Bransden 03] are given below in Table A.1, and some useful natural constants are provided in Table A.2. The latest values of the physical constants can be found from the *Committee on Data for Science and Technology*[COD] .

unit	**relation**	**comment**
length	$a_0 = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} = 5.29177 \times 10^{-11}$ m	Bohr radius
velocity	$v_0 = \frac{\hbar}{4\pi\epsilon_0 e^2} = \alpha c = 2.18769 \times 10^{6}$ m s^{-1}	classical orbiting velocity on the first Bohr orbital
time	$t_0 = \frac{a_0}{v_0} = 2.41888 \times 10^{-17}$ s	length / velocity
frequency	$\omega_0 = \frac{2\pi}{t_0} = 4.13414 \times 10^{16}$ s^{-1}	derived from the time definition
mass	$m_e = 9.10938 \times 10^{-31}$ kg	mass of the electron
momentum	$p_0 = 1.99285 \times 10^{-24}$ kg m s^{-1}	mass · velocity
charge	$q = 1.60218 \times 10^{-19}$ C	elementary charge
energy	$E_0 = \frac{1}{(4\pi\epsilon_0)^2}\frac{m_e e^4}{\hbar^2} = 27.2114$ eV	Rydberg energy
electric field	$\mathcal{E} = \frac{E_0}{a_0 e} = 5.14221 \times 10^{11}$ V m^{-1}	
intensity	$I = \frac{1}{2}\epsilon_0 c\mathcal{E}^2 = 3.50945 \times 10^{16}$ W cm^{-2}	

Table A.1: *Units and conversion factors for calculating with atomic units.*

constant	**symbol**	**[SI]**	**[a.u.]**
speed of light	c	299 792 458 m s^{-1}	137.036
Planck constant	h	$6.6260755 \times 10^{-34}$ kg m^2 s^{-1}	2π
vacuum permittivity	ϵ_0	8.8544×10^{-12} A s V^{-1} m^{-1}	$1/(4\pi)$
vacuum permeability	μ_0	$4\pi \times 10^{-7}$ V s A^{-1} m^{-1}	$4\pi/c^2$
Bohr magneton	μ_B	9.274×10^{-24} A m^2	1/2
proton mass	m_p	1.67262×10^{-27} kg	1836.15

Table A.2: *Important physical constants in SI and atomic units.*

Appendix B

Alternative Derivation of the Short-Time Propagator

In Chapter 2 the short time propagation method was derived by expanding a wave function $\Psi(\mathbf{r}, t)$ in a Taylor–series around t and applying the time–dependent Schrödinger equation to express the first derivative of the wave function. As an alternative, the Schrödinger equation provides the starting point for the time evolution operator $\hat{U}(t, t_0)$:

$$i\frac{\partial}{\partial t}\hat{U}(t, t_0) = \hat{H}(t)\hat{U}(t, t_0), \tag{B.1}$$

Formally integrating (B.1) with the proper initial condition gives:

$$\hat{U}(t, t_0) = \hat{\mathbf{1}} - i\int_{t_0}^{t} \hat{H}(t_1)\hat{U}(t_1, t_0)\mathrm{d}t_1. \tag{B.2}$$

The solution for $\hat{U}(t, t_0)$ also appears on the right hand side and iteratively follows the *Dyson*-series[Nolting 97]:

$$\hat{U}(t, t_0) = \hat{\mathbf{1}} - i\int_{t_0}^{t} \hat{H}(t_1)\mathrm{d}t_1 - \int_{t_0}^{t} \hat{H}(t_1)\int_{t_0}^{t_1} \hat{H}(t_2)\hat{U}(t_2, t_0)\mathrm{d}t_2\mathrm{d}t_1 \tag{B.3}$$

$$= \sum_{n=0}^{\infty} \hat{U}^{(n)}(t, t_0), \tag{B.4}$$

where $\hat{U}^{(n)}(t, t0)$ is defined as

$$\hat{U}^{(n)}(t, t_0) = (-i)^n \int_{t_0}^{t}\int_{t_0}^{t_1} \dots \int_{t_0}^{t_{n-1}} \hat{H}(t_1)\hat{H}(t_2)\dots\hat{H}(t_n)\mathrm{d}t_n \dots \mathrm{d}t_2\mathrm{d}t_1, \tag{B.5}$$

with $t_0 \leq t_n \leq \ldots \leq t_2 \leq t_1 \leq t$ and $\hat{U}^{(0)}(t,t_0) = \hat{\mathbf{1}}$. With the definition of the *Dyson*–time ordering operator,

$$\hat{\mathscr{T}}(\hat{A}(t_1)\hat{B}(t_2)) = \begin{cases} \hat{A}(t_1)\hat{B}(t_2) & \text{for } t_1 > t_2 \\ \hat{B}(t_2)\hat{A}(t_1) & \text{for } t_1 \leq t_2 \end{cases}, \tag{B.6}$$

which ensures the chronological time order for the product of two non–commuting operators, a relationship is obtained for a change of the integration limits:

$$\int_{t_0}^{t}\int_{t_0}^{t_1} \hat{H}(t_1)\hat{H}(t_2)\mathrm{d}t_2\mathrm{d}t_1 = \frac{1}{2}\iint_{t_0}^{t} \hat{\mathscr{T}}\left(\hat{H}(t_1)\hat{H}(t_2)\right)\mathrm{d}t_2\mathrm{d}t_1, \tag{B.7}$$

such that the time evolution operator $\hat{U}(t,t_0)$ can be written in the following form:

$$\hat{U}(t,t_0) = \sum_{n=0}^{\infty}\frac{(-i)^n}{n!}\int_{t_0}^{t}\ldots\int_{t_0}^{t}\hat{\mathscr{T}}\left(\hat{H}(t_1)\ldots\hat{H}(t_n)\right)\mathrm{d}t_n\ldots\mathrm{d}t_1 \tag{B.8}$$

$$= \hat{\mathscr{T}}\exp\left(-i\int_{t_0}^{t}\hat{H}(t')\mathrm{d}t'\right). \tag{B.9}$$

For the special case that $\left[\hat{H}(t_1),\hat{H}(t_2)\right] = 0$ for all t_1 and t_2, the action of $\hat{\mathscr{T}} = \hat{\mathbf{1}}$ becomes the identity, and (B.9) can be evaluated for any given time. This is the case, if the potential V is time independent, such that

$$\hat{U}(t,t_0) = \exp\left(-i\hat{H}\cdot(t-t_0)\right). \tag{B.10}$$

In general, however, time–dependent potentials are of interest, such that the assumption (B.10) is not valid. Therefore a time interval Δt is now considered which is sufficiently short, such that $\hat{V}(t+\tau)$ can be assumed as constant for all $\tau \in [0,\Delta t]$ and for any starting time t. Evaluating the Dyson-series (B.3) and omitting all terms of order Δt^3 or higher gives:

$$\hat{U}(t+\Delta t,t) = \hat{\mathbf{1}} - i\int_{t}^{t+\Delta t}\hat{H}(t_1)\mathrm{d}t_1 - \int_{t}^{t+\Delta t}\hat{H}(t_1)\int_{t}^{t_1}\hat{H}(t_2)\mathrm{d}t_2\mathrm{d}t_1 \tag{B.11}$$

$$= \hat{\mathbf{1}} - i\int_{0}^{\Delta t}\hat{H}(t+\tau_1)\mathrm{d}\tau_1 - \int_{0}^{\Delta t}\hat{H}(t+\tau_1)\int_{0}^{\tau_1}\hat{H}(t+\tau_2)\mathrm{d}\tau_2\mathrm{d}\tau_1. \tag{B.12}$$

For a small time τ, the Hamiltonian can be expanded in a Taylor–series:

$$\hat{H}(t+\tau) = \hat{H}(t) + \frac{\partial \hat{H}(t)}{\partial t}\tau + \frac{1}{2}\frac{\partial^2 \hat{H}(t)}{\partial t^2}\tau^2 + \hat{\mathscr{O}}(\tau^3). \tag{B.13}$$

With this expansion, the first integral in (B.12) becomes

$$\int_0^{\Delta t} \hat{H}(t+\tau_1)\mathrm{d}\tau_1 = \int_0^{\Delta t} \left(\hat{H}(t) + \frac{\partial \hat{H}(t)}{\partial t}\tau_1 + \frac{1}{2}\frac{\partial^2 \hat{H}(t)}{\partial t^2}\tau_1^2 \right) \mathrm{d}\tau_1 + \hat{\mathscr{O}}(\Delta\tau^3) \tag{B.14}$$

$$= \hat{H}(t)\Delta t + \frac{1}{2}\frac{\partial \hat{H}(t)}{\partial t}\Delta t^2 + \hat{\mathscr{O}}(\Delta t^3). \tag{B.15}$$

The second integral in (B.12) gives

$$\int_0^{\Delta t} \hat{H}(t+\tau_1) \int_0^{\tau_1} \hat{H}(t+\tau_2)\mathrm{d}\tau_2\mathrm{d}\tau_1 \tag{B.16}$$

$$= \int_0^{\Delta t} \hat{H}(t+\tau_1) \int_0^{\tau_1} \left(\hat{H}(t) + \frac{\partial \hat{H}(t)}{\partial t}\tau_2 + \frac{1}{2}\frac{\partial^2 \hat{H}(t)}{\partial t^2}\tau_2^2 \right) \mathrm{d}\tau_2\mathrm{d}\tau_1 + \hat{\mathscr{O}}(\Delta t^3) \tag{B.17}$$

$$= \int_0^{\Delta t} \left(\hat{H}(t) + \frac{\partial \hat{H}(t)}{\partial t}\tau_1 + \frac{1}{2}\frac{\partial^2 \hat{H}(t)}{\partial t^2}\tau_1^2 \right) \left(\hat{H}(t)\tau_1 + \frac{1}{2}\frac{\partial \hat{H}(t)}{\partial t}\tau_1^2 \right) \mathrm{d}\tau_1 + \hat{\mathscr{O}}(\Delta t^3) \tag{B.18}$$

$$= \int_0^{\Delta t} \left(\hat{H}^2(t)\tau_1 + \frac{1}{2}\hat{H}(t)\frac{\partial \hat{H}(t)}{\partial t}\tau_1^2 + \frac{\partial \hat{H}(t)}{\partial t}\hat{H}(t)\tau_1^2 \right) \mathrm{d}\tau_1 + \hat{\mathscr{O}}(\Delta t^3) \tag{B.19}$$

$$= \frac{1}{2}\hat{H}^2(t)\Delta t^2 + \hat{\mathscr{O}}(\Delta t^3) \tag{B.20}$$

Inserting (B.15) and (B.20) into (B.12) gives the short-time propagator

$$\hat{U}(t+\Delta t, t) = \hat{\mathbf{1}} - i\hat{H}(t)\Delta t - \frac{1}{2}\left(\hat{H}^2(t) + i\frac{\partial \hat{H}(t)}{\partial t} \right) \Delta t^2 + \hat{\mathscr{O}}(\Delta t^3), \tag{B.21}$$

which has an numerical error of Δt^3 and is identical with the previously obtained short time propagator (2.9).

Appendix C

«GOD is REAL unless declared INTEGER. »

Source Code Listings

C.1 Overview

This appendix gives more precise details on the numerical implementation and contains portions of the program source code that has been developed for the work described in this dissertation. The programs and subroutines have been mostly written in *Fortran 90* [For] as this programming language, despite its age and lack of features seen in modern languages, provides optimal performance and is still a widely used standard for large scale scientific computations.

The program has been divided in several logical blocks, *modules*, that provide certain functions (see Table C.1). Most modules are written in different versions for one–, two– and three–dimensions, such that all application specific calculations can be embedded in the main program module, *main.f90*, while all other general functions are defined in a proper module file.

Beside the main program file, the propagator module is the most important part of the numerical approach of solving the Schrödinger equation, as most of the wall clock time is spend for the Crank–Nicholson propagation. Therefore, stringent optimizations in the propagation loop is essential for efficient computations. One way of archiving fast running code is by using highly optimized mathematical libraries such as *LAPACK* [LAP] for the

Module	Description
main.f90	The main program. Initializes the different modules and contains the time loop of the program.
prop.f90	Implementation of the Crank–Nicholson propagation scheme (Section 2.2) for solving the time–dependent Schrödinger equation. Since a large fraction of the wall–time is spend for propagation, fine–grain parallelization has been used for the two– and three–dimensional version of this module.
pot.f90	Calculation of the potential array from an analytical form for (model-) potentials or reading in the values from an external file.
wfmath.f90	General routines for manipulating and accessing wave functions. (overlap integral between two wave functions, energy, saving and restoring from file, and Gaussian wave packets etc.)
vdetect.f90	Virtual detector, based on the work of B. Feuerstein[Feuerstein 03c] to obtain the kinetic energy and momentum distribution from the unbound flux of the wave function passing through a surface line close to the grid end in front of the absorber.
diag.f90	One–dimensional diagonalization routine for the tridiagonal Hamiltonian matrix[Press 92] providing energies and eigenfunctions of the bound states.
tools.f90	Obtaining the ground and first excited state wave functions for two– and three–dimensional potentials, using the imaginary time propagation (Section 2.6).
adk.f90	Used for calculating the ADK[Ammosov 86]–depleted ground state[Saenz 00] of a one–dimensional nuclear wave packet due to tunnel ionization in a laser field[Brichta 06] in order to obtain the nuclear wave function of the nuclear ion.
system.f90	Contains general string manipulation functions as well as routines to inquire the command line arguments for overwriting default input parameters of the program.
graphics.f90	Provides interactive visualization of the wave function and potential during the propagation for debugging purposes and to create animated films of the collision scenario using the *PGPlot*[PGP] or *PLPlot*[PLP] libraries.

Table C.1: *Description of the different program modules that have been developed and used.*

solution of the general tridiagonal linear equation set (see Section 2.5). Our approach, however, takes the special symmetry of the equation set into account where the upper and lower off–diagonal elements are a constant such that only the diagonal elements need to be described as an array. Therefore we can reduce the memory read operations significantly over the general tridiagonal solver routines.

Several crucial subroutines, especially for the three–dimensional calculations in Chapter 4 have been additionally optimized by using the *OpenMP* interface[Ope] for developing multithreading applications, which offers a fast and effective way to convert serial programs to parallel execution. Since the current trend of the processor industry is to increase the computing power by including more processing cores into one chip, adopting to shared–memory multi–threaded routines allows to gain the highest possible performance. Modern compilers such as the *Interl Fortran Compiler 9.1*[For], provide already automatic parallelization of simple loops and array operations, with OpenMP instructions, thereby allowing for a higher level of fine–grain (loop-level) and large–grain (function-level) optimizations. Since the split–operator Crank–Nicholson scheme reduces to a series of independent one–dimensional propagations along each coordinate direction, these propagations can be easily assigned to to the number of available processors within the inner propagation loop.

As an example, a selection a few most important modules and subroutines is provided below for the one–dimensional coupled–channel nuclear wave packet propagation in Chapter 5.

C.2 Main Program (`main.f90`)

```
!
!-----------------------------------------
! Main program
!-----------------------------------------
!
! Wave Propagation Method in 1 dimension for 4 coupled potential curves
! Source: CN1D program from Uwe Thumm
! Changes: Thomas Niederhausen 8/22/06
! Bernold Feuerstein 7/27/06
! Last change: Thomas 11/02/06
!
! NOTES:
! - The Gaussian laser pulse routine does not work for very short pulses
! since the time integral does not give a zero. Use more sophisticated
! routine to FFT a sin pulse, phase shift and back-FFT the pulse.
!-----------------------------------------
```

```
!

module progvars
  use vdetect
  implicit none

  ! input parameters
  real*8 :: deltaz          ! the grid spacing in a.u. for the electron
  real*8 :: widthz          ! the size of the grid in a.u.
  real*8 :: minz            ! the minumum z-value
  real*8 :: deltat          ! the time spacing for the ELECTRONIC propagation
  real*8 :: maxt            ! the total propagation time

  real*8 :: fadewidth       ! the width of the absorber for the electrons
  real*8 :: fadestrength    ! the strength of the absorbing border potential

  ! Parameters for the 1st "Pump" pulse
  ! Note: these pulse parameters only appear, if ADK based transition from the initial wave function of H2
  ! are used rather than a Franck-Condon transition for the initial vibrationl wave function.
  real*8 :: pumpomega       ! most likely the same as for the control pulse, but we are variable here
  real*8 :: pumpphase       ! the carrier envelope phase of the pump pulse
  real*8 :: pumpint         ! pulse intensity of the pump pulse
  real*8 :: pumplength      ! pulse length of the pump pulse

  ! Parameters for the 2nd "Control" pulse
  real*8 :: laseromega      ! the frequency of the laser
  real*8 :: laserphase      ! the carrier envelope phase
  real*8 :: laserint        ! intensity of the laser in a.u.
  real*8 :: laserlength     ! the pulse length of the laser pulse
  real*8 :: laserdelay      ! the time when the maximum laser intensity occurs
  real*8 :: laserramp       ! ramping time of the laser - only used for the CW laser pulse

  ! Parameters for 3rd, Coulomp explosion "Probe" pulse
  ! Note: some parameters don't make much sense though, like frequency or ce phase, since we currently
  ! only treat this pulse either via ADK such that only the intensity, the pulse lenght and the delay matter,
  ! or via a second Franck-Condon transition, such that only the delay is used. However, it could technically
  ! be treated just like a second control pulse as well.
  real*8 :: probeomega      ! most likely the same as for the control pulse, but we are variable here
  real*8 :: probephase      ! the carrier envelope phase of the probe pulse
  real*8 :: probeint        ! pulse intensity of the proble pulse
  real*8 :: probelength     ! pulse length of the probe pulse
  real*8 :: probedelay      ! the total delay (1st pulse starts the clock) for probe pulse

  type(Tdetector) :: vdetgerade     ! the virtual detector on the H2+ gerade curve
  type(Tdetector) :: vdetungerade   ! the virtual detector on the H2+ ungerade curve
  type(Tdetector) :: vdetground     ! the virtual detector on the H2 ground state curve
  type(Tdetector) :: vdetion        ! the virtual detector on the H2++ Coulomb explosion curve
  type(Tdetector) :: vdettotal      ! the virtual detector for the total wave function

  real*8 :: detectorwidth   ! the width of the virtual detector
  real*8 :: detectorgap     ! the gap between the virtual detector end and the absorbing boundary
  real*8 :: detectormaxp    ! maximum momentum resolved with the virtual detector
  real*8 :: detectorminp    ! and the minimum momentum
  real*8 :: detectormaxe    ! maximum energy resolved with the virtual detector
  real*8 :: detectormine    ! and the minimum energy
  integer :: detectorbins   ! number of grid points for the momentum binning

  integer :: ntprint        ! number of propagation steps between print outputs
  integer :: ntplot         ! number of propagation steps between plot outputs

  ! from program calculated parameters
  integer :: nzstep         ! the number of gridpoints for the electrons
  integer :: ntstep         ! the total number of timesteps for the propagation
  real*8 :: lasercycle      ! the length of one laser cycle
  real*8 :: mass            ! the mass of the particle
  real*8 :: detectorpos     ! the center position of the virtual detector

  ! dynamical variables for the propagation
  real*8 :: time            ! the current time
  real*8 :: laserenv        ! the laser envelope
  real*8 :: lasercphase     ! the current phase of the laser
  real*8 :: zfield          ! the laser electric field in z direction

  real*8 :: norm            ! the total norm inside the grid
  real*8 :: ionization      ! the ionization propability

  ! Declare additional wave functions that we will need. The PROPAGATOR defines the H2 ground wave function
  complex*16, allocatable :: psigerade(:)      ! two more wave function arrays for the H2+
  complex*16, allocatable :: psiungerade(:)    ! potential curves (gerade/ungerade)
  complex*16, allocatable :: psiion(:)         ! potential curve for the Coulomb explosion (p+p)
  complex*16, allocatable :: psitotal(:)       ! the total nuclear wave function
  real*8, allocatable, target :: coupling(:)   ! the coupling betweemaxn the H2+ gerade and ungerade states

  ! Declare ADK static variables to speed up the calculation of ADK rates
  real*8, allocatable :: adk_const_g(:)   ! lump all the R-dependent factors in here - H2+ gerade potential
  real*8, allocatable :: adk_kappa_g(:)   ! sqrt(2*I_p) - wave vector to ionize from gerade potential curve
  real*8, allocatable :: adk_const_u(:)   ! same for H2+ ungerade potential curve
  real*8, allocatable :: adk_kappa_u(:)   ! ditto
```

```
105
      ! other program related variables
      integer :: programstart    ! the start time of the program

      ! and now a few switches...
110   logical, parameter :: wfoutput = .false.          ! are we interested in the nuclear wave function?
      logical, parameter :: analyticalpulse = .true.    ! if false, the file "SPIDER.DAT" provides the laser pulse instead
      logical, parameter :: detector = .true.           ! do we need the dissociation energy and momentum spectra using virtual
          detector?
      logical, parameter :: adkrate = .true.            ! include ADK rates from the two singly ionized potential curves?
      logical, parameter :: looprun = .true.            ! loop through a series of delays or just a single delay run?
115   logical, parameter :: probepulse = .true.         ! are we having a Coulomb explosion probe pulse?
      logical, parameter :: adkpump = .true.            ! Use ADK rate based first ionization step or Franck-Condon principle instead?
      logical, parameter :: fcprobe = .false.           ! Use Franck-Condon principle for probe pulse or model the pulse analytically?
      logical, parameter :: normalizestates = .false.   ! for the vibrational state occupation - should we normalize to the bound states?

120 end module progvars

    !-----------------------------------------------------------------------

    program main
125
      implicit none

      call init                                 ! do all the initialization (array allocation etc.)
      call run                                  ! runs the program, single or loop mode
130   call done                                 ! That's it - we done! Now clear everything...

    end program main

    !-----------------------------------------------------------------------
135
    subroutine init
      ! allocate and initialize arrays etc.
      use debug
      use potentials
140   use propagator
      use wfmath
      use tools
      use vdetect
      use fft
145   use diag
      use progvars
      use graphics

      implicit none
150
      integer :: iAllocStatus

      call system_clock(programstart)                   ! save the starting time of the program

155   call setdebug(5,0)                                ! set the appropriate debug output level

      call init_parameters                              ! load some initial parameters

      nzstep  = nint(widthz / deltaz) + 1               ! calculate the number of gridpoints needed
160   ntstep  = nint(maxt / deltat) + 1

      ! initialize the potentials

      call  potentials_init (nzstep)                    ! allocate the potential array
165   call  potentials_setscale (minz,minz+widthz)      ! set the dimensions of the potential
      call  potentials_zero                             ! make sure that the initial potential is zero

      ! create absorbing walls (imaginary potential)

170   call  potentials_setabsorber_right (fadewidth, fadestrength / deltat)

      ! initialize the propagator

      call  propagator_init (nzstep)                    ! allocate the wavefunction arrays
175   call  propagator_setscale (minz,minz+widthz)      ! set the dimensions of the grid

      ! initialize the math, tools and virtual detector routines

      call wfmath_init(nzstep,minz,minz+widthz)         ! initialize the mathematics package
180   call  tools_init (nzstep,minz,minz+widthz)        ! initialize the tools package
      if (detector) call  vdetect_init (nzstep,minz,minz+widthz)

      ! adjust for the heavier nuclei
      ! SOURCE: E. R. Cohen and B. N. Taylor, Phys. Today 53 - 9 (2000)
185 ! mass = 918.0763505d0 ! reduced mass for HYDROGEN
      mass = 1835.241507d0                              ! reduced mass for DEUTERIUM
    ! mass = 2748.46079d0 ! reduced mass for TRITIUM

      call propagator_setmass(mass)
190   call wfmath_setmass(mass)
      if (detector) call vdetect_setmass(mass)
```

```
    ! initialize the FFT and diagonalization routines

    call fft_init (nzstep, deltaz, mass)          ! parameters needed for momentum spectrum of wave function
    call diag_init (nzstep, deltaz, mass)         ! needed for diagonalization of the potential

    ! we will also need two additional wave function arrays because of the nuclear couplings

    allocate(psigerade(nzstep), psiungerade(nzstep), psiion(nzstep), psitotal(nzstep), stat=iAllocStatus)
    if (iAllocStatus /= 0) then
      print *,"ERROR: can't allocate 1D wavefunction PSI"
      stop
    endif
    call debugmsg (1,1,"INIT: all 1D H2 wavefunctions allocated")

    allocate(coupling(nzstep), stat=iAllocStatus)
    if (iAllocStatus /= 0) then
      print *,"ERROR: can't allocate 1D couling array"
      stop
    endif
    call debugmsg (1,1,"INIT: 1D coupling array between H2+ gerade/ungerade curves allocated")

    ! output to the X-Window or into a separate file

    call graphics_init ("192.168.2.156:0.0/xw",4,2)

  end subroutine init

  !------------------------------------------------------------------------------

  subroutine done
    use debug
    use potentials
    use propagator
    use wfmath
    use tools
    use vdetect
    use fft
    use diag
    use progvars
    use graphics

    implicit none

    integer :: count, countrate, countmax
    integer :: iDeallocStatus

    call graphics_done

    ! deallocate all the used arrays
    call potentials_done
    call propagator_done
    call wfmath_done
    call tools_done
    if (detector) call vdetect_done
    call fft_done
    call diag_done

    deallocate(psigerade, psiungerade, psiion, psitotal, stat=iDeallocStatus)
    if (iDeallocStatus /= 0) then
      print *,"ERROR: can't deallocate wavefunction arrays"
      stop
    endif
    call debugmsg (1,1,"DONE: all 1D H2 wavefunctions destroyed")

    deallocate(coupling, stat=iDeallocStatus)
    if (iDeallocStatus /= 0) then
      print *,"ERROR: can't deallocate 1D coupling array"
      stop
    endif
    call debugmsg (1,1,"DONE: 1D coupling array between H2+ gerade/ungerade curves destroyed")

    call system_clock(count, countrate, countmax)   ! read the time for the program termination time
    count = count - programstart                    ! calculate the total running time
    print *,"TIME: ",real(count/countrate),"seconds [", count, countmax,"]"

  end subroutine done

  !------------------------------------------------------------------------------

  subroutine init_parameters
    use debug
    use system
    use progvars

    implicit none

    real*8, parameter :: pi = 3.14159265358979323846264 3d0
```

```
280
      deltaz   = system_realvariable("deltaz",0.05d0)              ! grid spacing in a.u. in z direction
      widthz   = system_realvariable("widthz",29.95d0)             ! length of the grid in direction parallel to internuclear axis
      minz     = system_realvariable("minz", deltaz)               ! the starting point of the grid
      maxt     = system_realvariable("maxt",15000.d0)              ! the maximum propagation time
285
      deltat   = system_realvariable("deltat",1.d0)                ! time spacing for the propagation

      ntprint  = nint( system_realvariable("ntprint",41.d0))       ! number of propagation steps between consecutive file outputs
      ntplot   = nint( system_realvariable("ntplot",128.d0))       ! number of propagation steps between consecutive screen plots
290
      fadewidth     = system_realvariable("fadewidth",10.0d0)      ! the width of the absorber in a.u.
      fadestrength  = system_realvariable("fadestrength",0.01d0)   ! the maximum heigth of the negative imaginary potential

      ! Control pulse parameters
295   ! laseromega = system_realvariable("laseromega",0.044d0) ! the frequency of the laser for 1064 nm
      laseromega     = system_realvariable("laseromega",0.057d0)     ! the frequency of the laser for 800 nm
      laserphase     = system_realvariable("phase",0.d0) * pi / 180  ! the carrier envelope phase in degrees
      laserint       = system_realvariable("int",0.00285d0)          ! laser intensity, default is 1E14 W/cm^2
      laserlength    = system_realvariable("pulselength",248.049d0)  ! laser pulse length, default is 6fs
300   laserdelay     = system_realvariable("laserdelay",1500.d0)     ! time of the maximum of the laser pulse
      laserramp      = system_realvariable("laserramp",100.d0)       ! time of the maximum of the laser pulse

      lasercycle  = 2*pi / laseromega                                ! this is the time for one laser cycle

305   ! Pump pulse parameters. Standard values assumed, wavelength by default same as for control pulse
      pumpomega      = system_realvariable("pumpomega",laseromega)       ! the frequency
      pumpphase      = system_realvariable("pumpphase",0.d0) * pi / 180  ! CE-phase, default is zero
      pumpint        = system_realvariable("pumpint",0.00285d0)          ! intensity, default is 1E14 W/cm^2
      pumplength     = system_realvariable("pumplength",248.049d0)       ! pulse length, default is 6fs
310
      ! Probe pulse parameters. Standard values assumed, wavelength by default same as for control pulse
      probeomega     = system_realvariable("probeomega",laseromega)      ! the frequency
      probephase     = system_realvariable("probephase",0.d0) * pi / 180 ! CE-phase, default is zero
      probeint       = system_realvariable("probeint",0.014247d0)        ! intensity, default is 5E14 W/cm^2
315   probelength    = system_realvariable("probelength",248.049d0)      ! pulse length, default is 6fs
      probedelay     = system_realvariable("probedelay", laserdelay+1000.d0) ! time of the probe pulse, default is 24fs after control

      detectorwidth  = system_realvariable("detectorwidth",5.d0) ! the width of the virtual detector
      detectorgap    = system_realvariable("detectorgap",1.d0)   ! gap between virtual detector and absorbing boundary
320   detectorminp   = system_realvariable("detectorminp",-10.d0)       ! minimum momentum
      detectormaxp   = system_realvariable("detectormaxp",50.d0)        ! maximum momentum
      detectormine   = system_realvariable("detectormine",0.d0)         ! minimum momentum
      detectormaxe   = system_realvariable("detectormaxe",0.5d0)        ! maximum momentum
      detectorbins   = nint( system_realvariable("detectorbins",1000.d0)) ! number of detector bins
325
      detectorpos    = minz + widthz - fadewidth - detectorgap - detectorwidth/2.d0

    end subroutine init_parameters

330 !------------------------------------------------------------------------------

    subroutine show_laserparameters
      ! just print the current laser parameters to the screen

335   use progvars

      implicit none

      real*8, parameter :: pi = 3.14159265358979323846264 3d0
340
      print '(A36,ES12.3,A7)',"␣Laser-Parameters:␣Intensity␣␣␣␣␣␣=␣", laserint * 3.50945d16,"W/cm^2"
      print '(A36,F12.3,A3)',"␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Wavelength␣␣␣␣␣=␣",2.d0*pi*137.036d0*5.29177d-2/laseromega,"nm"
      print '(A36,F12.3,A4)',"␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣CE-Phase␣␣␣␣␣␣␣=␣", laserphase * 180.d0 / pi,"DEG"
      print '(A36,F12.3,A3)',"␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Pulse␣duration␣=␣", laserlength * 2.41888d-2,"fs"
345   print '(A36,F12.3,A3)',"␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Pulse␣delay␣␣␣␣=␣", laserdelay * 2.41888d-2,"fs"

    end subroutine show_laserparameters

    !------------------------------------------------------------------------------
350
    subroutine init_detectors
      ! initialize the virtual detector objects, i.e. allocate the momentum and energy binning arrays
      use vdetect
      use progvars
355
      implicit none

      if (.not. detector) return

360   ! ensure proper disassociation of the new pointer objects
      nullify(vdetgerade%momentum)
      nullify(vdetgerade%energy)
      nullify(vdetgerade%rho)
      nullify(vdetgerade%flux)
365
      nullify(vdetungerade%momentum)
```

```
    nullify(vdetungerade%energy)
    nullify(vdetungerade%rho)
    nullify(vdetungerade%flux)

    nullify(vdetground%momentum)
    nullify(vdetground%energy)
    nullify(vdetground%rho)
    nullify(vdetground%flux)

    nullify(vdetion%momentum)
    nullify(vdetion%energy)
    nullify(vdetion%rho)
    nullify(vdetion%flux)

    nullify( vdettotal%momentum)
    nullify( vdettotal%energy)
    nullify( vdettotal%rho)
    nullify( vdettotal%flux)

    ! the positions for all the 4 detectors are identical
    call  vdetect_setposition (vdetgerade,   detectorpos, detectorwidth)
    call  vdetect_setposition (vdetungerade, detectorpos, detectorwidth)
    call  vdetect_setposition (vdetground,   detectorpos, detectorwidth)
    call  vdetect_setposition (vdetion,      detectorpos, detectorwidth)
    call  vdetect_setposition ( vdettotal,   detectorpos, detectorwidth)

    ! the momentum resolution is also identical for all detectors
    call vdetect_setmomentumgrid(vdetgerade,   detectorminp,detectormaxp, detectorbins)
    call vdetect_setmomentumgrid(vdetungerade,detectorminp,detectormaxp, detectorbins)
    call vdetect_setmomentumgrid(vdetground,   detectorminp,detectormaxp, detectorbins)
    call vdetect_setmomentumgrid(vdetion,      detectorminp,detectormaxp, detectorbins)
    call vdetect_setmomentumgrid(vdettotal,    detectorminp,detectormaxp, detectorbins)

    ! and so is the energy resolution
    call  vdetect_setenergygrid (vdetgerade,   detectormine,detectormaxe, detectorbins)
    call  vdetect_setenergygrid (vdetungerade,detectormine,detectormaxe, detectorbins)
    call  vdetect_setenergygrid (vdetground,   detectormine,detectormaxe, detectorbins)
    call  vdetect_setenergygrid (vdetion,      detectormine,detectormaxe, detectorbins)
    call  vdetect_setenergygrid ( vdettotal,   detectormine,detectormaxe, detectorbins)

  end subroutine init_detectors

  !-----------------------------------------------------------------------

  subroutine done_detectors
    ! destroy the virtual detector objects, i.e. deallocate the momentum and energy binning arrays
    use vdetect
    use progvars

    implicit none

    if (.not. detector) return

    call  vdetect_destroy (vdetgerade)
    call  vdetect_destroy (vdetungerade)
    call  vdetect_destroy (vdetground)
    call  vdetect_destroy (vdetion)
    call  vdetect_destroy ( vdettotal)

  end subroutine done_detectors

  !-----------------------------------------------------------------------

  subroutine init_adk
    ! initialize the ADK rate arrays and precompute the static variables
    ! IMPORTANT: it is assumed that the GERADE curve is POTENTIAL(2) and the
    ! UNGERADE curve is POTENTIAL(3)!!!

    ! The molecular ADK rates are taken from eq. (5) of J. P. Brichta et al., J. Phys. B 39, 3769 (2006)

    use progvars
    use potentials
    use debug

    implicit none

    integer :: iAllocStatus          ! for I/O operation
    real*8  :: e, pi, z              ! some general variables
    integer :: nR
    real*8  :: rpos, adkg, adku      ! position and ADK rates
    real*8  :: maxfield              ! the maximum electric field strenth of the laser

    allocate(adk_const_g(nzstep), adk_const_u(nzstep), adk_kappa_g(nzstep), adk_kappa_u(nzstep), stat=iAllocStatus)
    if (iAllocStatus /= 0) then
      print *,"ERROR: can't allocate ADK arrays"
      stop
    endif
```

```
455     ! now procompute the static part of the ADK rates
        e  = dexp(1.d0)          ! the Euler constant
        pi = dacos(-1.d0)        ! well guess - what could this be...
        z  = 2.d0                ! effective charge of the two nucleii together - obviously this is 2 for H2 and D2.

460     adk_kappa_g = dsqrt(2.d0 * (pot_curve4 - pot_curve2))     ! sqrt(2*I_p) with I_p the ionization potential
        adk_kappa_u = dsqrt(2.d0 * (pot_curve4 - pot_curve3))     ! same for the ungerade curve

        adk_const_g = (3.d0*e/pi)**1.5d0 &
                      *(adk_kappa_g**4.5d0)/(z**2.5d0) &
465                   *(4.d0*e*(adk_kappa_g**4)/z)**(2.d0*z/adk_kappa_g-1.5d0)

        adk_const_u = (3.d0*e/pi)**1.5d0 &
                      *(adk_kappa_u**4.5d0)/(z**2.5d0) &
                      *(4.d0*e*(adk_kappa_u**4)/z)**(2.d0*z/adk_kappa_u-1.5d0)
470
        maxfield = dsqrt( laserint )                              ! the maximum electric field strength of the laser
        open(unit=131,file="test_adk_max.dat",status="replace",access="sequential",recl=1024)
        write(131,'(A6,2A12)') "R","ADK_G","ADK_UG"
        do nR=1, nzstep
475       rpos = (nR-1)*deltaz + minz                             ! grid position of the nucleii
          adkg = adk_const_g(nR) * (maxfield**(1.5d0 - 2.d0*z/adk_kappa_g(nR))) &
                 *dexp(-2.d0*adk_kappa_g(nR)**3/(3.d0*maxfield)) ! the maximum ADK rate from the gerade curve
          adku = adk_const_u(nR) * (maxfield**(1.5d0 - 2.d0*z/adk_kappa_u(nR))) &
                 *dexp(-2.d0*adk_kappa_u(nR)**3/(3.d0*maxfield)) ! maximum ADK rate from the ungerade curve
480
          write(131,'(F6.2,2E12.4)') rpos, adkg, adku
        enddo
        write(131,*)
        close(131)
485
        call debugmsg (5,1,"INIT_ADK: ADK arrays allocated, precomputed and plotted")

      end subroutine init_adk

490   !-------------------------------------------------------------------------

      subroutine done_adk
        ! destroy the ADK arrays
        use progvars
495     use debug

        implicit none

        integer :: iDeallocStatus          ! for I/O operation
500
        deallocate(adk_const_g,adk_const_u,adk_kappa_g,adk_kappa_u, stat=iDeallocStatus)
        if ( iDeallocStatus /= 0) then
          print *,"ERROR: can't deallocate ADK arrays"
          stop
505     endif

        call debugmsg (5,1,"DONE_ADK: ADK arrays destroyed")

      end subroutine done_adk
510
      !-------------------------------------------------------------------------

      subroutine readcoupling(filename)
        ! read in a coupling array between the given two potential curves POT1 and POT2
515
        use debug
        use potentials
        use progvars

520     implicit none

        character(len=*), intent(in) :: filename          ! potential file to read
        real*8, pointer              :: tmpptr(:)         ! a temporary pointer to save the content of the potential

525     tmpptr => potential                               ! save the current potential pointer (Note POTENTIAL is a pointer!)

        ! now find the right coupling array where we temporarily redirect POTENTIAL to
          potential => coupling

530     ! after we assigned POTENTIAL with the coupling array, we can now use the READFROMFILE routine from POTENTIALS
        ! just like reading in a potential array.

        call  potentials_readfromfile (filename)

535     potential => tmpptr                               ! now restore the previous potential we had

        call debugmsg(5,1,"READCOUPLING: couplings read from file")

      end subroutine readcoupling
540
      !-------------------------------------------------------------------------
```

```
      subroutine coupleH2Plus(timestep, fieldstrength)
        ! this routine does the dipole coupling between the H2+ gerade and ungerade curves
545
        use progvars

        implicit none

550     real*8, intent(in) :: timestep
        real*8, intent(in) :: fieldstrength
        integer            :: nR
        complex*16         :: tmp
        real*8             :: costmp, sintmp, tmpvalue
555
        ! The couplings are pointwise, i.e. PSIGERADE(R) is only coupled to PSIUNGERADE(R) with the same R.
        ! We go pointwise through the wave functions, therefore we can have this loop running in parallel using OpenMP
        ! Since the routine will be called quite often, it has been optimized for speed.

560   !$OMP PARALLEL DO SCHEDULE(STATIC,1) PRIVATE(nR, tmp, costmp, sintmp, tmpvalue)
        do nR=1, nzstep

          tmpvalue = coupling(nR)*timestep*fieldstrength   ! this is the argument of the trigonometric functions
          costmp   = dcos(tmpvalue)                        ! essentially how much original wave function is left after coupling
565       sintmp   = dsin(tmpvalue)                        ! the amount of the other wave function that mixes in

          tmp              = psigerade(nR)                 ! save the value since we overwrite it below
          psigerade(nR)    = psigerade(nR)*costmp   - psiungerade(nR)*dcmplx(0,sintmp)     ! the new gerade wave function
          psiungerade(nR) = psiungerade(nR)*costmp - tmp*dcmplx(0,sintmp)                 ! we need TMP since GERADE was just
              overwritten
570
        enddo
      !$OMP END PARALLEL DO

      end subroutine coupleH2Plus
575
      !-------------------------------------------------------------------------

      subroutine adktransition(timestep, fieldstrength)
        ! this routine does the ADK rate depletion of both H2+ curves to the p+p Coulomb explosion curve.
580     ! The ADK rate acts like a optical potential damping the H2+ wave functions but cannot coherently
        ! build up the correct wave function on the 2H+ curve due to the lost phase information.
        ! Instead we poppulate the upper potential curve constructively and do not propagate this wave function,
        ! such that the 1/R mapping of the potential curve resembles the wave function on the detector.

585     use progvars

        implicit none

        real*8, intent(in) :: timestep
590     real*8, intent(in) :: fieldstrength
        real*8             :: fieldabs
        integer            :: nR
        complex*16         :: tmpg, tmpu
        real*8             :: adkg, adku
595     real*8, parameter :: z = 2.d0

        fieldabs = dabs(fieldstrength)
        if (fieldabs < 1.d-4) return                                  ! certainly nothing to do here, if there is no field

600     ! The ADK rate transitions are pointwise for every R.
        ! We go pointwise through the wave functions, therefore we can have this loop running in parallel using OpenMP
        ! Since the routine will be called quite often, it has been optimized for speed.

      !$OMP PARALLEL DO SCHEDULE(STATIC,1) PRIVATE(nR, tmpg, tmpu, adkg, adku)
605     do nR=1, nzstep
          adkg = adk_const_g(nR) * (fieldabs**(1.5d0 - 2.d0*z/adk_kappa_g(nR))) &
                 *dexp(-2.d0*adk_kappa_g(nR)**3/(3.d0*fieldabs))      ! the current ADK rate from the gerade curve
          adku = adk_const_u(nR) * (fieldabs**(1.5d0 - 2.d0*z/adk_kappa_u(nR))) &
                 *dexp(-2.d0*adk_kappa_u(nR)**3/(3.d0*fieldabs))      ! the current ADK rate from the ungerade curve
610
          tmpg             = dsqrt(dabs(adkg*timestep)) * psigerade(nR)     ! the ADK transition amplitude
          psigerade(nR)    = psigerade(nR) - tmpg                           ! depletion of the gerade wave function

          tmpu             = dsqrt(dabs(adkg*timestep)) * psiungerade(nR)   ! the ADK transition amplitude
615       psiungerade(nR) = psiungerade(nR) - tmpu                         ! the depletion of the ungerade wave function

          ! the ionization now is problematic, since we do not have any phase information from the ADK rates
      ! psiion(nR) = psiion(nR) + tmpg + tmpu ! the ionization curve
      ! psiion(nR) = psiion(nR) + sign(fieldstrength,1.d0)*(tmpg + tmpu) ! phase adjusted variation
620       psiion(nR)       = psiion(nR) + cdabs(tmpg) + cdabs(tmpu)         ! only add magnitudes
        enddo
      !$OMP END PARALLEL DO

      end subroutine adktransition
625
      !-------------------------------------------------------------------------

      function analyticalfield(lasertime) result(fieldstrength)
        ! return the field strength of the laser pulse specified in the PROGVAR block at the given LASERTIME
```

```
630     ! for an analytical form of the pulse, i.e. a Gaussian pulse.

        use progvars

        implicit none
635
        real*8, intent(in) :: lasertime
        real*8             :: fieldstrength
        real*8, parameter :: pi = 3.14159265358979323846264 3d0
        real*8             :: envelope, phase
640
        envelope      = dsqrt( laserint ) * dexp(- 2.d0 * dlog(2.d0) * (( lasertime - laserdelay)**2) / ( laserlength **2) )
        phase         = dmod(laseromega*(time-laserdelay)+laserphase,2*pi)  ! the current laser phase with respect to the maximum of
            the pulse
         fieldstrength  = envelope * dcos(phase)

645     if (( probepulse).and.(.not. fcprobe)) then                          ! add the probe pulse if given
          envelope      = dsqrt( probeint ) * dexp(- 2.d0 * dlog(2.d0) * (( lasertime - probedelay)**2) / ( probelength **2) )
          phase         = dmod(probeomega*(time-probedelay)+probephase,2*pi) ! the current probe laser phase
           fieldstrength  = fieldstrength  + envelope * dcos(phase)           ! add the probe field to the pulse
        endif
650
        return

      end function analyticalfield

655   !-----------------------------------------------------------------------

      function spiderfield( lasertime ) result( fieldstrength )
        ! return the field strength of the laser pulse just as in LASERFIELD, but using an experimental
        ! pulse from the file SPIDER.DAT (refers to the Heidelberg group using SPIDER to analyze the pulses).
660     ! The file contains three columns: time, field amplitude and phase. The peak should be at time = 0 and
        ! the amplitude normalized to 1.

        use progvars

665     implicit none

        real*8, intent(in)          :: lasertime
        real*8                      :: fieldstrength
        real*8, parameter           :: pi = 3.14159265358979323846264 3d0
670     logical, save               :: initialized = .false.          ! first time of the call - we need to read in the spider file
        real*8, save                :: mintime, maxtime, deltatime    ! the time grid in the file
        real*8, allocatable, save :: spideramplitude(:)               ! the amplitude of the pulse envelope
        real*8, allocatable, save :: spiderphase(:)                   ! the phase of the field
        integer, save               :: maxcount                       ! largest array value
675     integer                     :: counter, i
        real*8                      :: loctime, amplitude, phase
        integer                     :: iostatus

        if (.not. initialized ) then                                  ! if this is the first time this routine is called...
680       open(unit=130,file="spider.dat",access="sequential",recl=1024,status="old") ! open file

          ! count the number of records in the file and assign min and max values
          counter = 0
          read(130,*,iostat=iostatus) loctime, amplitude, phase       ! read the first record
685       mintime = loctime
          do while (iostatus == 0)
            maxtime = loctime
            counter = counter + 1
            read(130,*,iostat=iostatus) loctime, amplitude, phase     ! read the next record
690       end do
          deltatime = (maxtime - mintime) / (counter - 1)             ! the equidistant time step from the file
          maxcount = counter

          ! now allocate the arrays
695       allocate(spideramplitude(counter), spiderphase(counter), stat=iostatus)
          if ( iostatus /= 0) then
            print *,"ERROR: can't allocate Spider pulse array"
            stop
          endif
700       rewind(130)                                                 ! back to the start of the file

          do i=1, counter                                             ! now read the file to the arrays
            read(130,*) loctime, amplitude, phase
            spideramplitude(i) = amplitude
705         spiderphase(i) = phase
          enddo

          close(130)
           initialized = .true.
710     end if ! *first time initialization*

        ! find the correct value from the memory array for the Spider pulse

        loctime = lasertime - laserdelay                              ! the shift of the pulse
715     if (loctime > maxtime) then
          counter = maxcount
```

```
    else if (loctime < mintime) then
      counter = 1
    else
720   counter = nint((loctime - mintime) / deltatime) + 1
    endif

    amplitude     = dsqrt( laserint ) * spideramplitude(counter)
    phase         = spiderphase(counter)
725  fieldstrength  = amplitude * dcos(phase)

    if ((probepulse).and.(.not. fcprobe)) then                    ! add an ANALYTICAL probe pulse if given
      amplitude     = dsqrt( probeint ) * dexp(- 2.d0 * dlog(2.d0) * ((lasertime - probedelay)**2) / (probelength**2) )
      phase         = dmod(probeomega*(time-probedelay)+probephase,2*pi) ! the current probe laser phase
730    fieldstrength  = fieldstrength + amplitude * dcos(phase)          ! add the probe field to the pulse
    endif

    return

735 end function spiderfield

    !-------------------------------------------------------------------------

    function laserfield(lasertime) result( fieldstrength )
740   ! return the field strength of the laser pulse specified in the PROGVAR block at the given LASERTIME

      use progvars

      implicit none
745
      real*8, intent(in) :: lasertime
      real*8             :: fieldstrength
      real*8             :: analyticalfield , spiderfield

750   if ( analyticalpulse ) then
        fieldstrength = analyticalfield (lasertime)       ! use the analytical form for the pulse
      else
        fieldstrength = spiderfield (lasertime)           ! use the experimental pulse shape
      endif
755
    end function laserfield

    !-------------------------------------------------------------------------

760 subroutine run
      ! Run the program. Loop or not is specified in the variable LOOPRUN

      use potentials
      use propagator
765   use wfmath
      use tools
      use vdetect
      use diag
      use adk
770   use progvars
      use graphics
      use system
      use debug

775   implicit none

      integer           :: nt              ! that is our time counter
      character(len=80) :: line            ! for output on the screen
      real*8            :: laserfield      ! the function that gives us the electric field strenth of the laser
780   complex*16        :: autocorrelation ! the auto-correlation of the H2+ gerade wave function
      real*8            :: boundnorm       ! the norm of the H2+ gerade bound states
      type(TStates)     :: states          ! object with wave functions and energies from diagonalization
      integer           :: nstate          ! number of the H2+ eigenstate we currently consider
      integer           :: ndelay, nmax    ! the delay counter
785   real*8            :: delay           ! the current delay value

      ! Note: initialization of the grid parameters and variables has been done already during INIT.
      ! Here we only do the real work.

790   print *," ---------------------------- "
      print *,"Wave Packet Propagation in 1D"
      print *," ---------------------------- "
      print *,""

795   print *,"STEP 1: Reading in potential curves and couplings from file"

      call potentials_setactive (1)                       ! activate the H2 ground state curve
      call potentials_readfromfile ("H2_pot_ground")      ! and read the curve

800   call potentials_setactive (2)                       ! activate the H2+ gerade curve
      call potentials_readfromfile ("H2+pot_gerade")      ! and read the curve too

      call potentials_setactive (3)                       ! activate the H2+ ungerade curve
      call potentials_readfromfile ("H2+pot_ungerade")    ! and read the curve as well
```

```
805
      call potentials_setactive (4)                          ! activate the H2++ CE curve
      call potentials_readfromfile ("H2++ce")                ! and read this guy too

      call readcoupling("H2+dpcouplings")                    ! and get the couplings between H2+ states (gerade/ungerade)
810
      ! And now nicely plot all the curves on the screen

      call potentials_setactive (1)                          ! H2 ground state curve
      call graphics_lineplot ( potential ,1, minz,minz+widthz,-1.2d0,-1.d0,label="Potential H2 (ground state)")
815   call potentials_setactive (2)                          ! H2+ gerade curve
      call graphics_lineplot ( potential ,2, minz,minz+widthz,-0.12d0,0.d0,label="Potential H2+ (gerade)")
      call potentials_setactive (3)                          ! H2+ ungerade curve
      call graphics_lineplot ( potential ,3, minz,minz+widthz,-0.0d0,1.d0,label="Potential H2+ (ungerade)")
      call potentials_setactive (4)                          ! H2+ ungerade curve
820   call graphics_lineplot ( potential ,4, minz,minz+widthz,-0.0d0,1.d0,label="Potential p+p (CE)")

      print *," ------------------------------------ "
      print *,""
      print *,"STEP 2: Diagonalizing the potential of H2+(gerade)"
825
      call potentials_setactive (2)                          ! switch to the H2+ gerade potential

      nullify( states%energyarr)                             ! ensure proper disassociation of the new pointers
      nullify( states%wavefunctions)
830
      call diag_diagonalize ( potential , states )           ! find the bound states of the potential - wave functions too
      print *,"Number of bound states: ", states%numbound    ! number of bound states of the potential
      call diag_plot ( states , "test")                      ! write the energies and wave functionsin a file

835   print *," ------------------------------------ "
      print *,""
      print *,"STEP 3: Obtaining the ground state wave function on the H2 curve"
      call wfmath_gaussian(psi,2.0 d0,1. d0,0. d0)           ! some arbitrary trial wave function
    ! psi = sum(states%wavefunctions,1) ! flat superposition of all vibrational states
840 ! psi(:) = states%wavefunctions(states%numbound-1,:) ! or one particular vibrational wave function
      call wfmath_normalize(psigerade)                             ! and normalize this guy of course
      call graphics_lineplot ( psi ,5, minz,minz+widthz,0.d0,1.d0, label ="H2 nuclear wf probability")

      ! now use Imaginary Time Propagation on curve 1 to find the ground state
845   call potentials_setactive (1)                          ! H2 ground state curve
      call tools_groundstate ( psi , deltat )                ! now we like to know the ground state of the potential using ITP
      call graphics_lineplot ( psi ,5, minz,minz+widthz,0.d0,1.d0, label ="H2 nuclear wf probability")
      print *,"Energy of groundstate: ", dreal(wfmath_energy(psi))*27.2114d0," eV"
      print *,"<R> expectation value: ", dreal(wfmath_dipole( psi , psi)),"a.u."
850
      print *," ------------------------------------ "
      print *,""
      print *,"STEP 4: Obtaining initial wave function after first ionization"

855   ! One can in principle use Franck-Condon transition and just move the H2 wave function on the H2+ gerade curve.
      ! According to Urbain et al., Phys. Rev. Lett. 92, 163004 (2004) this is incorrect and overestimates the higher
      ! vibrational levels due to the R-dependence of the ionization step. A better method therefore account for
      ! this dependence by using ADK rates, but still assume an instantaneous (short pulse!!!) first ionization step.

860   if (adkpump) then
        print *,"---> Method: ADK rate dependent instantaneous ionization"
        psigerade = psi                                      ! first we need a copy of the original ground state wave function
        call adk_deplete( psi, pot_curve1, pot_curve2, 1.d0, pumpint, pumplength, pumpomega, pumpphase) ! ADK deplete this state
        psigerade = psigerade - psi                          ! subtract the final ground state from the original - the new excited
            state
865     call wfmath_normalize(psigerade)                     ! of course we want to start with normalized H2+
      else
        print *,"---> Method: simple Franck-Condon transition"
        psigerade = psi                                      ! just copy the wave function to the H2+ gerade potential curve
    ! call wfmathzero(psi) ! and the ground state wave function is gone
870   endif
      psi = psigerade                                        ! now backup the initial H2+ wave function for the loop run

      call wfmath_zero(psiungerade)                          ! and let us have no wave function in the H2+ ungerade state yet
      call wfmath_zero(psiion)                               ! same with the Coulomb explosion wave function. Nothing there yet.
875
      call potentials_setactive (2)                          ! switch to the H2+ gerade potential
      print *,"Norm of H2+gerade state: ", wfmath_norm(psigerade)
      print *,"Energy of H2+gerade state: ", dreal(wfmath_energy(psigerade))*27.2114d0," eV"
      print *,"<R> expectation value: ", dreal(wfmath_dipole( psigerade , psigerade)),"a.u."
880
      call potentials_setactive (3)                          ! switch to the H2+ ungerade potential
      print *,"Norm of H2+ungerade state: ", wfmath_norm(psiungerade)
      print *,"Energy of H2+ungerade state: ", dreal(wfmath_energy(psiungerade))*27.2114d0," eV"
      print *,"<R> expectation value: ", dreal(wfmath_dipole(psiungerade,psiungerade)),"a.u."
885
      call graphics_lineplot ( psi ,5, minz,minz+widthz,0.d0,1.d0, label ="H2 nuclear wf probability")
      call graphics_lineplot ( psigerade ,6, minz,minz+widthz,0.d0,1.d0, label ="H2+ nuclear wf probability")
      call graphics_lineplot ( psiungerade ,7, minz,minz+widthz,0.d0,1.d0, label ="H2+ nuclear wf probability")
      call graphics_lineplot ( psiion ,8, minz,minz+widthz,0.d0,1.d0, label ="H2++ nuclear wf probability")
890
      open(unit=111,file="test_FC_factors.dat",status="replace",access="sequential",recl=1024)
```

```
      write(111,'(A6,2A12)') "State","Energy","FC-Factor"
      do nstate=1, states%numbound                          ! output the overlap with the eigenfunctions of H2+gerade
        write(111,'(I6,2E12.4)') nstate, states%energyarr(nstate), &
895       cdabs(wfmath_overlap(psigerade(:), states%wavefunctions(nstate,:)))**2
      enddo
      write(111,*)
      close(111)

900   print *," ------------------------------------ "
      print *,""
      print *,"STEP 5: Propagating wave packet in laser field"
      call show_laserparameters                             ! Lets just see what we are calculating today.

905   call propagator_startpropagation                      ! does absolutely nothing, but a good idea to call anyhow
      if (detector) call init_detectors                     ! create a couple virtual detectors
      if (adkrate) call init_adk                            ! initialize ADK rate arrays and precompute static variables

      ! open a file if we are interested in the wave function output
910
      if (looprun) then                                     ! if looping, we want final states and some matrix output
        open(unit=120,file="test_stateoccupation.dat",status="replace",access="sequential",recl=1024)
        open(unit=121,file="test_Rexpval.dat",status="replace",access="sequential",recl=1024)
        open(unit=122,file="test_Rwidth.dat",status="replace",access="sequential",recl=1024)
915     open(unit=123,file="test_norm.dat",status="replace",access="sequential",recl=1024)

        write(123,'(4A12)') "Delay","Delay_fs","Norm_G","Norm_UG"
        nmax = 3000                                         ! maximum loop counter
      else                                                  ! only one run - we are interested in time evolution
920     open(unit=100,file="test_coupled.dat",status="replace",access="sequential",recl=1024)
        write(100,'(8A12)') "TIME","Time_fs","Field","NormG","<R_G>","NormU","<R_U>","Correlation"

        open(unit=110,file="test_correlation.dat",status="replace",access="sequential",recl=1024)

925     if (wfoutput) then                                  ! wave function makes only sense in single run mode
          open(unit=101,file="coupled_wf_h2pgerade.dat",status="replace",access="sequential",recl=1024)
          open(unit=102,file="coupled_wf_h2pungerade.dat",status="replace",access="sequential",recl=1024)
          open(unit=105,file="coupled_wf_total.dat",status="replace",access="sequential",recl=1024)
        endif
930
        nmax = 0                                            ! only one run through the following loop
      endif

      if (looprun) then
935     print *,"*** Entering Delay Loop Mode ***"
      else
        print *,"*** Single Time Loop Run ***"
      endif

940   do ndelay=0, nmax
        if (looprun) then
          delay = real(ndelay)/(2.41888d-2) !+ laserdelay
      ! laserdelay = delay ! override the value for the laser delay
          probedelay = laserdelay + delay                                ! override the value for the probe laser delay
945   ! ntstep = nint((delay + 5.d0*laserlength + 3000.d0) / deltat) + 1 ! adjust maximum propagation time
      ! ntstep = nint((laserdelay + delay + 5.d0*laserlength + 3000.d0) / deltat) + 1 ! adjust maximum propagation time
          ntstep = nint((delay + 100000.d0) / deltat) + 1
        endif

950     psigerade = psi                                     ! use the backup initial H2+ wave function
        call wfmath_zero(psiungerade)                       ! and let us have no wave function in the H2+ ungerade state yet

        call propagator_startpropagation                    ! does absolutely nothing, but a good idea to call anyhow

955     if (looprun) then                                   ! if looping, write out the looping parameters
          write(121,'(E12.4,$)') delay, delay*2.41888d-2
          write(122,'(E12.4,$)') delay, delay*2.41888d-2
        endif

960     if (detector) then                                  ! reset the virtual detector if used
          call vdetect_reset (vdetgerade)                   ! this is important especially in loop mode to avoid
          call vdetect_reset (vdetungerade)                 ! and inherited counter from the previous loop run
          call vdetect_reset (vdettotal)
          call vdetect_reset (vdetion)
965     endif

        do nt=0, ntstep                                     ! here comes the time loop
          time = nt*deltat                                  ! the current time

970       ! === PROBE PULSE ===

          if ((probepulse).and.(fcprobe)) then              ! enter this loop only if we have Franck-Condon probe pulse
            ! The Coulomb explosion step is assumed to be instantaneous for Franck-Condon. Thus we need to find the
            ! propagation time step in the time loop where we insert the CE routine.
975         if ((time >= probedelay) .and. (time-probedelay < deltat)) then ! only to be executed ONCE - at the probe time
              print *,"*** Applying Instantaneous Franck-Condon Probe Laser Pulse ***"
              psiion = psitotal                             ! Franck-Condon transition
              call wfmath_zero(psigerade)                   ! big closing sale!!!
              call wfmath_zero(psiungerade)                 ! everything has to go...
```

```
980             call wfmath_zero( psitotal )                    ! and gone too
                print *
              endif ! (* Ensure only one run of the routine *)
            endif ! (* Existenze of probe pulse? *)

985         ! === END PROBE PULSE

            zfield = laserfield (time)                          ! calculate the electric field for the laser including any probe if given

            ! === PROPAGATION AND COUPLING ===
990
            if (adkrate) call adktransition ( deltat /2.d0, zfield )    ! couple H2+ wave functions to the p+p curve
            call coupleH2Plus( deltat /2.d0, zfield )           ! do one coupling between H2+ gerade/ungerade curves for half
                 timestep

            call potentials_setactive (2)                       ! propagate in the H2+ gerade potential
995         call propagator_cnprop( psigerade, dcmplx( deltat ,0. d0), potential , absorber)    ! and propagate a time step

            call potentials_setactive (3)                       ! propagate in the H2+ ungerade potential
            call propagator_cnprop(psiungerade, dcmplx( deltat ,0. d0), potential , absorber)   ! and propagate a time step

1000        if ((probepulse).and.(fcprobe)) then                ! propagate on H2++ curve if we use Franck-Condon for the probe pulse
              call potentials_setactive (4)                     ! propagate in the H2++ Coulomb explosion potential
              call propagator_cnprop( psiion , dcmplx( deltat ,0. d0), potential , absorber)    ! and propagate a time step
            endif

1005        call coupleH2Plus( deltat /2.d0, zfield )           ! do one coupling between H2+ gerade/ungerade curves for half
                 timestep
            if (adkrate) call adktransition ( deltat /2.d0, zfield )    ! couple H2+ wave functions to the p+p curve

            psitotal = psigerade + psiungerade                  ! calculate the total wave function (this time without the ground state)

1010        ! === END OF PROPAGATION AND COUPLING ===

            if ( detector) then                                 ! apply the virtual detector to the wave functions
              call vdetect_detector (vdetgerade, psigerade, deltat)
              call vdetect_detector (vdetungerade,psiungerade, deltat)
1015          call vdetect_detector ( vdettotal , psitotal , deltat)
              call vdetect_detector (vdetion, psitotal , deltat)
            endif

            if ((mod(nt,ntprint).eq.0).or.(nt.eq.ntstep-1)) then ! check if we plot this round
1020          write(line,'(I5,A,I5,A,I5,A,F8.2,A,F8.2)') ndelay,"␣Iterations:",nt,"/",ntstep-1,"␣Time:",time,"/",maxt
              line(len(line):len(line)) = char(13)              ! position the cursor back on the beginning of the line
              print '(A$)', line                                ! overwrite the current line on the screen

              if (looprun) then                                 ! output data - different in loop and single run mode
1025            write(121,'(E12.4,$)') dreal(wfmath_dipole( psigerade , psigerade))
                write(122,'(E12.4,$)') wfmath_width(psigerade)
              else                                              ! in single run mode we want time evolution, that means more work here
                autocorrelation = wfmath_overlap(psigerade, psi)    ! calculate the auto-correlation

1030            write(100,'(F12.2,F12.4,6E12.4)') time, time*2.41888d-2, zfield, &
                 wfmath_norm(psigerade), dreal(wfmath_dipole( psigerade , psigerade)), &
                 wfmath_norm(psiungerade), dreal(wfmath_dipole(psiungerade,psiungerade)), &
                 cdabs( autocorrelation )**2

1035            write(110,'(E12.4,$)') time, time*2.41888d-2
                boundnorm = wfmath_norm(psigerade)              ! calculate the norm of the remaining bound H2+ states
                do nstate=1, states%numbound                    ! output the overlap with the eigenfunctions of H2+gerade
                  if ( normalizestates) then                    ! we normalize to the remaining bound state
                    write(110,'(E12.4,$)') cdabs(wfmath_overlap(psigerade(:) , states%wavefunctions(nstate,:)))**2/boundnorm
1040              else                                          ! otherwise give the remaining TOTAL vibration state occupation
                    write(110,'(E12.4,$)') cdabs(wfmath_overlap(psigerade(:) , states%wavefunctions(nstate,:)))**2
                  endif
                enddo
                write(110,*)
1045
                if (wfoutput) then                              ! write wave function to disk - if feature is activated
                  write(101,'(E12.4,$)') cdabs( psigerade)**2
                  write(101,*)

1050              write(102,'(E12.4,$)') cdabs(psiungerade)**2
                  write(102,*)

                  write(105,'(E12.4,$)') cdabs( psigerade+psiungerade)**2
                  write(105,*)
1055            endif
              endif   ! (* looprun test *)

            endif ! (* data output condition *)

1060        if ((mod(nt,ntplot).eq.0).or.(nt.eq.ntstep-1)) then  ! check if we plot graphics this round
              call graphics_lineplot (psigerade,6, minz,minz+widthz,0.d0,1.d0, label="H2+␣nuclear␣wf␣probability")
              call graphics_lineplot (psiungerade,7, minz,minz+widthz,0.d0,1.d0, label="H2+␣nuclear␣wf␣probability")
              call graphics_lineplot (psiion ,8, minz,minz+widthz,0.d0,1.d0, label="H2++␣nuclear␣wf␣probability")
            endif
1065
```

```
      enddo ! (* timeloop *)

      if ((detector).and.(probepulse).and.(.not. fcprobe)) then ! special treatment, if no propagation occured on H2++ curve
        ! the virtual detector on the H2++ curve can only detect if we propagate the wave function. For the case of
        ! ADK depletion of the single ionized states, we loose the phase information and thus only collect wave function
        ! probability on the H2++ curve without further propagation. Therefore we need to map the R-dependence of the
        ! wave function probability to a detector count in energy space.
        call vdetect_reset (vdetion)                          ! first delete what we collected - it doesn't count anyway
        call vdetect_reverseimage(vdetion, psiion, pot_curve4) ! get detector counts from reversing wave function on potential curve
      endif

      if (looprun) then                                       ! loop mode output is after the time evolution here
        autocorrelation = wfmath_overlap(psigerade, psi)      ! calculate the auto-correlation
        write(120,'(E12.4,$)') delay, delay*2.41888d-2
        boundnorm = wfmath_norm(psigerade)
        do nstate=1, states%numbound                          ! output the overlap with the eigenfunctions of H2+gerade
          if (normalizestates) then
            write(120,'(E12.4,$)') cdabs(wfmath_overlap(psigerade(:), states%wavefunctions(nstate,:)))**2 / boundnorm
          else
            write(120,'(E12.4,$)') cdabs(wfmath_overlap(psigerade(:), states%wavefunctions(nstate,:)))**2
          endif
        enddo
        write(120,*)
        write(121,*)
        write(122,*)
        write(123,'(4E12.4)') delay, delay*2.41888d-2,wfmath_norm(psigerade),wfmath_norm(psiungerade)

        ! Plot the data from the virtual detectors
        if (detector) then
          call vdetect_plot_variant (vdetgerade,"H2+gerade")
          call vdetect_plot_variant (vdetungerade,"H2+ungerade")
          call vdetect_plot_variant (vdettotal,"H2_total")
          call vdetect_plot_variant (vdetion,"H2++CE")
        endif
      endif ! (* looprun test *)

    enddo ! (* delay loop *)

    if (looprun) then
      close(120)
      close(121)
      close(122)
      close(123)
    else
      call potentials_setactive (2)                           ! switch to the H2+ gerade potential
      print *,"Energy of H2+gerade state :", dreal(wfmath_energy(psigerade))*27.2114d0," eV"
      print *,"<R> expectation value :", dreal(wfmath_dipole(psigerade,psigerade)),"a.u."

      call potentials_setactive (3)                           ! switch to the H2+ ungerade potential
      print *,"Energy of H2+ungerade state :", dreal(wfmath_energy(psiungerade))*27.2114d0," eV"
      print *,"<R> expectation value :", dreal(wfmath_dipole(psiungerade,psiungerade)),"a.u."

      ! Plot the data from the virtual detectors
      if (detector) then
        call vdetect_plot (vdetgerade,"H2+gerade")
        call vdetect_plot (vdetungerade,"H2+ungerade")
        call vdetect_plot (vdettotal,"H2_total")
        call vdetect_plot (vdetion,"H2++CE")
      endif

      close(100)
      close(110)
      close(112)

      if (wfoutput) then
        close(101)                                            ! close the wave function file
        close(102)                                            ! close the wave function file
        close(105)                                            ! close the wave function file
      endif
    endif ! (* looprun test *)

    if (detector) call done_detectors                         ! release all the virtual detectors
    if (adkrate)  call done_adk                               ! release the ADK arrays

    call diag_release (states)                                ! release the diagonalized wave function array

  end subroutine run
```

C.3 Propagator Module (prop1D.f90)

```
!-----------------------------------------
! DEFINITION OF CRANK-NICHOLSON PROPAGATOR MODULE
!-----------------------------------------
! Author : Thomas Niederhausen
```

```
! Sources : Uwe Thumm and Bernold Feuerstein
! Date : 6 Jun 2006
! contains Crank-Nicholson propagator algorithm
!-----------------------------------------------

! usage:
!
! PROPAGATOR_INIT(zdim)
! allocates the 1D storage arrays for the wavefunctions
!
! PROPAGATOR_DONE
! kills the 1D storage arrays for the wavefunctions
!
! PROPAGATOR_SETSCALE(zmin,zmax)
! sets the left (zmin) and right (zmax) grid boundaries in atomic units and thus
! defines the grid spacing in z-direction
!
! PROPAGATOR_SETMASS(zmass)
! sets the mass of the particle for the propagation (heavier particles move
! slower).
!
! PROPAGATOR_GETZMIN
! PROPAGATOR_GETZMAX
! PROPAGATOR_GETDELTAZ
! PROPAGATOR_GETMASS
! functions do return specific information about the numerical grid to the caller
!
! PROPAGATOR_CNPROP(psiin, timestep, potential, absorber)
! propagates the given wavefunction with Crank-Nicholson Algorithm for on timestep
! If optional absorber is given, the wavefunction is masked with this array to reduce
! boudary reflections. Note: timestep must be COMPLEX*16, to use the routine for
! imaginary time propagation as well
!
! PROPAGATOR_CNITPROP(psiin, timestep, potential, absorber)
! same as PROPAGATOR_CNPROP, but now timestep is REAL*8 and the propagation transforms
! to imaginary time propagation to find ground states for the given potential.


module propagator
  implicit none

  complex*16, allocatable :: psi(:)                    ! the 1D wavefunction input

  complex*16, allocatable, private :: psiout(:)        ! local wavefunction for propagation
  complex*16, allocatable, private :: gamma(:)         ! used for CN-propagation algorithm
  complex*16, allocatable, private :: diagarr(:)       ! diagonal elements used for CN-propagation algorithm

  integer, private         :: nzstep                   ! the dimenstion of the 1D potential
  real*8, private          :: massz=1.0                ! the masses of particles in a.u.
  real*8, private          :: deltaz                   ! the grid spacing
  real*8, private          :: zmin                     ! defines the origin (in a.u.) of the grid

  contains

!-----------------------------------------------------------------------
! INITIALIZATION
!-----------------------------------------------------------------------

  subroutine propagator_init(zdim)
    use debug
    integer, intent(in) :: zdim              ! the dimension of the arrays
    integer             :: iAllocStatus      ! return value

    ! allocate 1D wavefunction
    allocate(psi(zdim),stat=iAllocStatus)
    if ( iAllocStatus /= 0) then
      print *,"ERROR: can't allocate 1D wavefunction PSI"
      stop
    endif
    call debugmsg (10,1,"PROPAGATOR_INIT: 1D wavefunction PSI allocated")

    ! allocate 1D local wavefunctions
    allocate(psiout(zdim),stat=iAllocStatus)
    if ( iAllocStatus /= 0) then
      print *,"ERROR: can't allocate 1D local wavefunction array"
      stop
    endif
    call debugmsg (10,1,"PROPAGATOR_INIT: 1D local wavefunction array allocated")

    ! allocate temporary CN-propagator array
    allocate(gamma(zdim),stat=iAllocStatus)
    if ( iAllocStatus /= 0) then
      print *,"ERROR: can't allocate temporary propagator array GAMMA"
      stop
    endif
    call debugmsg (10,1,"PROPAGATOR_INIT: temporary propagator array GAMMA allocated")

    ! allocate CN diagonal elements array
```

```
      allocate(diagarr(zdim),stat=iAllocStatus)
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't allocate DIAGARR array"
        stop
      endif
      call debugmsg (10,1,"PROPAGATOR_INIT: DIAGARR allocated")

      ! set the dimensions of the wavefunction
      nzstep = zdim

      call debugmsg(5,1,"PROPAGATOR_INIT: propagation arrays allocated")
    end subroutine propagator_init
!-----------------------------------------------------------------------

    subroutine propagator_done
      use debug
      integer              :: iDeallocStatus      ! return value

      ! deallocate 1D wavefunction
      deallocate(psi,stat=iDeallocStatus)
      if ( iDeallocStatus /= 0) then
        print *,"ERROR: can't deallocate wavefunction arrays"
        stop
      endif
      call debugmsg (10,1,"PROPAGATOR_DONE: wavefunction arrays destroyed")

      ! deallocate 1D sub-wavefunctions
      deallocate(psiout,stat=iDeallocStatus)
      if ( iDeallocStatus /= 0) then
        print *,"ERROR: can't deallocate 1D local wavefunction array"
        stop
      endif
      call debugmsg (10,1,"PROPAGATOR_DONE: 1D local wavefunction array destroyed")

      deallocate(gamma,stat=iDeallocStatus)
      if ( iDeallocStatus /= 0) then
        print *,"ERROR: can't deallocate temporary propagator array GAMMA"
        stop
      endif
      call debugmsg (10,1,"PROPAGATOR_DONE: temporary propagator array GAMMA destroyed")

      deallocate(diagarr,stat=iDeallocStatus)
      if ( iDeallocStatus /= 0) then
        print *,"ERROR: can't deallocate DIAGARR"
        stop
      endif
      call debugmsg (10,1,"PROPAGATOR_DONE: DIAGARR destroyed")

      call debugmsg(5,1,"PROPAGATOR_DONE: propagation arrays destroyed")
    end subroutine propagator_done

!-----------------------------------------------------------------------

    subroutine propagator_startpropagation
      ! For compatibility with variable time step routine
    end subroutine propagator_startpropagation

!-----------------------------------------------------------------------

    subroutine propagator_setscale(min, max)
      ! sets the grid boundaries in atomic units in z-direction
      real*8, intent(in) :: min, max

      zmin   = min
      deltaz = (max - min) / ( nzstep - 1)
    end subroutine propagator_setscale

!-----------------------------------------------------------------------

    subroutine propagator_setmass(zmass)
      ! sets the masses of the particles
      real*8, intent(in)              :: zmass

      massz = zmass  ! sets the first mass parameter

    end subroutine propagator_setmass

!-----------------------------------------------------------------------
! GRID INQUIRY FUNCTIONS
!-----------------------------------------------------------------------

    function propagator_getzmin() result(min)
      real*8 :: min
      min = zmin
    end function propagator_getzmin

!-----------------------------------------------------------------------
```

```
    function propagator_getzmax() result(max)
      real*8 :: max
      max = zmin + nzstep*deltaz
    end function propagator_getzmax

!-----------------------------------------------------------------------

    function propagator_getdeltaz() result(delta)
      real*8 :: delta
      delta = deltaz
    end function propagator_getdeltaz

!-----------------------------------------------------------------------

    function propagator_getmass() result(mass)
      real*8 :: mass
      mass = massz
    end function propagator_getmass

!-----------------------------------------------------------------------
! PROPAGATION
!-----------------------------------------------------------------------

    subroutine propagator_cnprop(psiin, timestep, potential, absorber)
      ! uses Crank-Nicholson propagation scheme to propagate wavefunction for small timestep
      complex*16, intent(inout), dimension(:) :: psiin
      real*8, intent(in), dimension(:)        :: potential
      real*8, intent(in), dimension(:), optional :: absorber
      complex*16, intent(in)                  :: timestep          ! complex for imaginary time propagation
      complex*16, parameter                   :: iu = (0.d0,1.d0)  ! imaginary unit
      complex*16                              :: diagz             ! the diagonal elements without potential term
      complex*16                              :: subdiagz          ! the constant off-diagonal elements
      integer                                 :: nz

      ! 1.) Calculate constant tridiagonal elements of propagation matrix (Hamiltonean)

      diagz = 0.5 + iu * timestep / (massz * 4.0 * deltaz**2)     ! Note: potential is added in propagation loop

      subdiagz = -iu * timestep / (massz * 8.0 * deltaz**2)


      ! 2.) Propagate a full potential and kinetic energy step at once

      diagarr = potential * iu * timestep / 4.d0 + diagz       ! the diagonal elements including potential
      call cntridag_nonconst(psiin, psiout, diagarr, subdiagz)      ! propagate in z-direction


      ! 3.) Propagate half a potential step

! psiin = psiin * (1.d0 - iu * potential * timestep / 4.d0) / (1.d0 + iu * potential * timestep / 4.d0)


      ! 3.) Propagate a full timestep T_z in z direction

! call cntridag_const(psiin,psiout,diagz,subdiagz) ! propagate in z-direction
! psiin = psiout ! write back the result into the wavefunction


      ! 4.) Propagate another half a potential step

! psiin = psiin * (1.d0 - iu * potential * timestep / 4.d0) / (1.d0 + iu * potential * timestep / 4.d0)


      ! 5.) Apply the absorber to reduce grid boundary reflections (if absorber array is given)
      if (present(absorber)) then
        psiin = psiout*absorber
      else
        psiin = psiout
      endif

    end subroutine propagator_cnprop

!-----------------------------------------------------------------------

    subroutine propagator_cnitprop(psiin, timestep, potential, absorber)
      ! uses Crank-Nicholson propagation scheme for small timestep imaginary time propagation (finding ground state)
      complex*16, intent(inout), dimension(:) :: psiin
      real*8, intent(in), dimension(:)        :: potential
      real*8, intent(in), dimension(:), optional :: absorber
      real*8, intent(in)                      :: timestep
      complex*16, parameter                   :: iu = (0.d0,1.d0)    ! imaginary unit

      if (present(absorber)) then
        call propagator_cnprop(psiin, -iu*timestep, potential, absorber)
      else
        call propagator_cnprop(psiin, -iu*timestep, potential)
      endif
```

```
  end subroutine propagator_cnitprop
!-----------------------------------------------------------------------

  subroutine cntridag_const(psiin,psiout,diag,subdiag)
    ! modified NUMERICAL RECIPES TRIDAG routine for constant diagonal and super/subdiagonal elements
    ! Ref.: Num. Recipes, p.24, sec. 2.4, U. Thumm ITAMP, 6/15/99
    complex*16, intent(in) :: psiin(:)
    complex*16, intent(out) :: psiout(:)
    complex*16, intent(in) :: diag, subdiag
    complex*16             :: beta
    integer                :: j

    beta     = diag
    psiout(1) = psiin(1) / beta

    do j=2,size(psiin)
      gamma(j) = subdiag / beta
      beta     = diag - subdiag * gamma(j)
      if (beta.eq.0) then
        pause "ERROR: CNTRIDAG_NONCONST failed (beta=0)"
        stop
      endif
      psiout(j) = (psiin(j) - subdiag * psiout(j-1)) / beta
    enddo

    do j=size(psiin)-1,1,-1
      psiout(j) = psiout(j) - gamma(j+1) * psiout(j+1)     ! this loop clearly cries for optimization!!!
    enddo

    psiout = psiout - psiin      ! subtract initial vector
  end subroutine cntridag_const

!-----------------------------------------------------------------------

  subroutine cntridag_nonconst(psiin,psiout,diag,subdiag)
    ! modified NUMERICAL RECIPES TRIDAG routine for constant diagonal and super/subdiagonal elements
    ! Ref.: Num. Recipes, p.24, sec. 2.4, U. Thumm ITAMP, 6/15/99
    complex*16, intent(in) :: psiin(:), diag(:)
    complex*16, intent(out) :: psiout(:)
    complex*16, intent(in) :: subdiag
    complex*16             :: beta
    integer                :: j

    beta     = diag(1)
    psiout(1) = psiin(1) / beta

    do j=2,size(psiin)
      gamma(j) = subdiag / beta
      beta     = diag(j) - subdiag * gamma(j)
      if (beta.eq.0) then
        pause "ERROR: CNTRIDAG_CONST failed (beta=0)"
        stop
      endif
      psiout(j) = (psiin(j) - subdiag * psiout(j-1)) / beta
    enddo

    do j=size(psiin)-1,1,-1
      psiout(j) = psiout(j) - gamma(j+1) * psiout(j+1)     ! this loop clearly cries for optimization!!!
    enddo

    psiout = psiout - psiin      ! subtract initial vector
  end subroutine cntridag_nonconst

!-----------------------------------------------------------------------

end module propagator
```

C.4 Potential Module (pot1D.f90)

```
!-----------------------------------------------
! DEFINITION OF POTENTIAL MODULE
!-----------------------------------------------
! Author : Thomas Niederhausen
! Date : 24 Aug 2006
! provides 1D potential routines for 3 coupled hydrogenic potential curves
!-----------------------------------------------

! usage:
!
! POTENTIALS_INIT(zdim)
! allocates the 1D storage arrays for potential and absorber, but does NOT fill
! them with useful values
!
```

```
! POTENTIALS_DONE
! kills the 1D storage arrays for potential and absorber
!
! POTENTIALS_BACKUP
! creates an array to store a backup of the current potential
! useful to store the time independent part of the potential
!
! POTENTIALS_RESTORE
! loads the backup potential into the current potential array
! useful to add the time dependent part afterwards
!
! POTENTIALS_REALZERO
! sets the real potential to be a flat, zero potential
!
! POTENTIALS_OPTICALZERO
! since the optical potential is implemented as a masking function, this sets the
! masking array to 1 everywhere in space, equivalent to a zero optical potential
!
! POTENTIALS_ZERO
! calls the two functions above and thus sets the effective complex potential
! to a flat homeogenious space
!
! POTENTIALS_SETSCALE(zmin,zmax)
! sets the left (zmin) and right (zmax) grid boundaries in atomic units and thus
! defines the grid spacing in z-direction
!
! POTENTIALS_SETABSORBER(width,strength)
! defines the strength and width of the aborbing boundaries on both sides
!
! POTENTIALS_PLOT
! plots the current potential (real and imaginary part) in separate files
! in ASCII format
!
! POTENTIALS_SOFTHYDROGEN(zpos, softpar [optional], charge [optional])
! adds a soft-core hydrogen ion at the given position to the potential
! the softening parameter is optional - otherwise a default value 0.641 is assumed
! the charge gives the nuclear charge in case it is not hydrogen atom.
!
! POTENTIALS_SOFTHELIUM(zpos, softpar [optional], effcharge [optional])
! adds a soft-core helium-atom at the given position to the potential. The
! potential is an active 1-electron potential for the outermost electron,
! which includes the screening from the inner electron.
! The softening parameter and the effective charge are optional and are adjusted
! to reproduce the ionization potential of the second electron for the ground state
! 1s2s and the first excited state 1s2s. They NEED TO GET ADJUSTED!!!!
!
! POTENTIALS_EFIELD(zint, xint)
! adds a electric field gradient of the given intensity in x and z direction
!
! POTENTIALS_READFROMFILE(filename)
! read the potential data from the given filename in the current POTENTIAL
! array.

module potentials
  implicit none

  real*8, allocatable :: absorber(:)          ! bumper potential
  real*8, pointer     :: potential(:)         ! the current potetial
  real*8, allocatable :: potentialbackup(:)   ! a copy of the potential to store the static part

  real*8, allocatable, target :: pot_curve1(:), pot_curve2(:), pot_curve3(:), pot_curve4(:)  ! these are our 4 potential curves

  integer, private    :: nzstep               ! the dimenstion of the 1D potentials
  real*8, private     :: deltaz               ! the grid spacing
  real*8, private     :: zmin                 ! defines the origin (in a.u.) of the grid

  contains

!------------------------------------------------------------------------
! I N I T I A L I Z A T I O N
!------------------------------------------------------------------------

  subroutine potentials_init(zdim)
    use debug
    integer, intent(in) :: zdim        ! the dimension of the arrays
    integer             :: iAllocStatus       ! return value

    ! allocate potential arrays
    allocate(pot_curve1(zdim),pot_curve2(zdim),pot_curve3(zdim),pot_curve4(zdim),stat=iAllocStatus)
    if (iAllocStatus /= 0) then
      print *,"ERROR: can't allocate POTENTIAL arrays"
      stop
    endif
    call debugmsg(10,1,"POTENTIALS_INIT: POTENTIAL arrays allocated")

    ! allocate bumper potential
    allocate(absorber(zdim),stat=iAllocStatus)
    if (iAllocStatus /= 0) then
```

```
        print *,"ERROR: can't allocate ABSORBER array"
        stop
      endif
      call debugmsg (10,1,"POTENTIALS_INIT: ABSORBER allocated")

      ! set the dimensions of the potential
      nzstep = zdim

      ! and make sure we have an active potential assigned - initialized with the lowest curve.
      call potentials_setactive (1)

      call debugmsg(5,1,"POTENTIALS_INIT: 1D potentials allocated")

    end subroutine potentials_init

  !------------------------------------------------------------------------

    subroutine potentials_done
      use debug
      integer :: iDeallocStatus       ! return value

      ! check and deallocate the potential backup copy if allocated
      if ( allocated ( potentialbackup )) then
        deallocate(potentialbackup,stat=iDeallocStatus)
        if (iDeAllocStatus /= 0) then
          print *,"ERROR: can't deallocate POTENTIALBACKUP array"
          stop
        endif
        call debugmsg(10,1,"POTENTIALS_DONE: POTENTIALBACKUP destroyed")
      endif

      ! deallocate potential arrays
      deallocate(pot_curve1, pot_curve2, pot_curve3, pot_curve4,stat=iDeallocStatus)
      if (iDeAllocStatus /= 0) then
        print *,"ERROR: can't deallocate POTENTIAL arrays"
        stop
      endif
      call debugmsg(10,1,"POTENTIALS_DONE: POTENTIAL arrays destroyed")

      ! deallocate bumper potential
      deallocate(absorber,stat=iDeallocStatus)
      if ( iDeallocStatus /= 0) then
        print *,"ERROR: can't deallocate ABSORBER array"
        stop
      endif
      call debugmsg (10,1,"POTENTIALS_DONE: ABSORBER destroyed")

      call debugmsg(5,1,"POTENTIALS_DONE: 1D potentials destroyed")

    end subroutine potentials_done

  !------------------------------------------------------------------------

    subroutine potentials_setscale (min, max)
      ! sets the grid boundaries in atomic units in z-direction
      real*8, intent(in) :: min, max

      zmin  = min
      deltaz = (max - min) / ( nzstep - 1)
    end subroutine potentials_setscale

  !------------------------------------------------------------------------

    subroutine potentials_setactive (potnr)
      ! chooses which potential curve is the currently active potential curve
      integer, intent(in) :: potnr

      select case(potnr)
        case(1)
          potential => pot_curve1
        case(2)
          potential => pot_curve2
        case(3)
          potential => pot_curve3
        case(4)
          potential => pot_curve4
        case default
          print *," POTENTIALS_SETACTIVE: ERROR- potential number ",potnr,"is not within [1,2,3] range"
          stop
      end select
    end subroutine potentials_setactive

  !------------------------------------------------------------------------

    subroutine potentials_backup
      ! creates a backup copy of the current potential
      use debug
      integer :: iAllocStatus
```

```
    ! check if the potential backup copy already exists and create the array if needed
    if (.not. allocated(potentialbackup)) then
      allocate(potentialbackup(nzstep),stat=iAllocStatus)
      if (iAllocStatus /= 0) then
        print *,"ERROR: can't allocate POTENTIAL array"
        stop
      endif
      call debugmsg(5,1,"POTENTIALS_BACKUP: POTENTIALBACKUP allocated")
    endif

    potentialbackup = potential    ! copy the current potential to the backup copy

    call debugmsg(8,1,"POTENTIALS_BACKUP: current potential stored")
  end subroutine potentials_backup

!-----------------------------------------------------------------------

  subroutine potentials_restore
    ! copies the backup copy of the potential in the current potential array

    potential = potentialbackup

  end subroutine potentials_restore

!-----------------------------------------------------------------------
! P O T E N T I A L S
!-----------------------------------------------------------------------

  subroutine potentials_realzero
    ! sets real part of potential to 0 everywhere in space

    potential = 0.d0

  end subroutine potentials_realzero

!-----------------------------------------------------------------------

  subroutine potentials_opticalzero
    ! sets masking array to 1 everywhere in space

    absorber = 1.d0
  end subroutine potentials_opticalzero

!-----------------------------------------------------------------------

  subroutine potentials_zero
    ! sets optical and real potential to 0 everywhere in space

    call potentials_realzero
    call potentials_opticalzero
  end subroutine potentials_zero

!-----------------------------------------------------------------------

  subroutine potentials_softhydrogen(zpos, softpar, charge)
    ! adds a soft-core hydrogen ion at the given position to the potential
    ! the softening parameter is optional - otherwise a default value is assumed
    ! the charge is also optional - if note given we assume Z=1
    real*8, intent(in)             :: zpos
    real*8, intent(in), optional :: softpar
    real*8, intent(in), optional :: charge
    integer                        :: nz
    real*8                         :: z, rsq
    real*8                         :: softparameter
    real*8                         :: corecharge

    ! check for softenting parameter. default is 0.641 which makes I_p=0.5 a.u. in 2D

    ! *** ADJUST the softening parameter!!!

    if (present(softpar)) then
      softparameter = softpar
    else
      softparameter = 0.641d0
    endif

    ! check for nuclear charge parameter. default is Z=1 for hydrogen
    if (present(charge)) then
      corecharge = charge
    else
      corecharge = 1.d0
    endif

    z = zmin
    do nz=1, nzstep

      rsq = (z - zpos)**2
```

```
        potential(nz) = potential(nz) - corecharge / dsqrt(rsq + softparameter)

        z = z + deltaz                ! next grid position in z-direction
      enddo

    end subroutine potentials_softhydrogen

!-------------------------------------------------------------------------------

    subroutine potentials_softmolecule( rdist , apar, charge)
      ! Improved molecular soft-core 1D potential, based on B. Feuerstein
      ! (see Phys. Rev. A 67 - 043405 (2003)) using a softening function
      ! a(R) in order to reproduce the 1_sigma_g potential curve in 1D
      ! correctly. The potential of one nucleus is given by
      ! V(z) = - Q / ( sqrt(z^2 + (a/b)^2) + 1/a - a/b ) with a=a(R), and b=5
      !
      ! This version uses a natural cubic spline interpolation

      use mathtools

      real*8, intent(in)            :: rdist
      real*8, intent(in), optional :: apar
      real*8, intent(in), optional :: charge
      integer                       :: nz, nR
      real*8                        :: z, q, a, rsqa, rsqb
      ! Table of precomputed fitting values for interpolation
      real*8, parameter             :: rtab(50) = (/ 0.d0, 0.2d0, 0.4d0, 0.6d0, 0.8d0, 1.d0, 1.2d0, 1.4d0, 1.6d0, 1.8d0, 2.d0, &
                                                     2.2d0, 2.4d0, 2.6d0, 2.8d0, 3.d0, 3.2d0, 3.4d0, 3.6d0, 3.8d0, 4.d0, 4.2d0, &
                                                     4.4d0, 4.6d0, 4.8d0, 5.d0, 5.5d0, 6.d0, 6.5d0, 7.d0, 7.5d0, 8.d0, 8.5d0, &
                                                     9.d0, 10.d0, 11.d0, 12.d0, 13.d0, 14.d0, 15.d0, 16.d0, 17.d0, 18.d0, 19.d0, &
                                                     20.d0, 21.d0, 22.d0, 23.d0, 24.d0, 25.d0 /)
      real*8, parameter             :: atab(50) = (/ 1.565108776d0, 1.520073891d0, 1.443648815d0, 1.370487690d0, 1.306910992d0, &
                                                     1.252897263d0, 1.206868172d0, 1.167484999d0, 1.133425713d0, 1.103911877d0, &
                                                     1.078041077d0, 1.055446148d0, 1.035488129d0, 1.018023968d0, 1.002538681d0, &
                                                     0.9890131950d0, 0.9770517349d0, 0.9666829109d0, 0.9575562477d0, 0.9497456551d0, &
                                                     0.9428887367d0, 0.9371356964d0, 0.9321336746d0, 0.9280090332d0, 0.9244518280d0, &
                                                     0.9216003418d0, 0.9163241386d0, 0.9130887985d0, 0.9111528397d0, 0.9098558426d0, &
                                                     0.9087901115d0, 0.9078197479d0, 0.9069590569d0, 0.9061222076d0, 0.9048431900d0, &
                                                     0.9031679200d0, 0.9015463800d0, 0.9002052300d0, 0.8991304400d0, 0.8982425700d0, &
                                                     0.8975135800d0, 0.8969014200d0, 0.8963780400d0, 0.8959341000d0, 0.8955509200d0, &
                                                     0.8952191400d0, 0.8949247400d0, 0.8946677200d0, 0.8944317300d0, 0.8942307900d0 &
                                                     /)

      real*8, save                  :: dtab(50)
      logical                       :: dtabinit = .false.

      ! for the spline interpolation we need the table of second derivatives to be initialized first.
      if ( dtabinit == .false.) then
        call math_spline(rtab, atab, dtab)          ! compute the second derivatives of the interpolating function
        dtabinit = .true.                           ! okay - next time we do not have to compute this table again
      endif

      ! If no charge is given, we assume hydrogen. Softening is only adjusted to 1D H2+ molecule!!!
      if (present(charge)) then
        q = charge
      else
        q = 1.d0
      endif

      ! Use the interpolated fitting value a if not specified otherwise
      if (present(apar)) then
        a = apar
      else
! ! now we have to do some linear interpolation
! do nR=lbound(rtab,1), ubound(rtab,1) ! loop through the distance table
! if (rtab(nR) ¿= rdist) exit ! and find the next bigger distance from the given value
! enddo ! rdist is now in the interval specified by [nR-1]...[nR]
        if ( rdist >= rtab(ubound(rtab,1))) then              ! check if we are out of bound, i.e. distance larger than table allows
          a = atab(ubound(rtab,1))                  ! to big value for RDIST - return the largest tabulated value for a
        else                                        ! everything clear - we can interpolate
          a = math_spline_interpolation(rtab, atab, dtab, rdist)   ! do the cubic spline interpolation
! a = atab(nR-1) + (atab(nR) - atab(nR-1)) * (rdist - rtab(nR-1)) / (rtab(nR) - rtab(nR-1)) ! linear interpolation
        endif
      endif

      z = zmin
      do nz=1, nzstep

        rsqa = (z - rdist/2.d0)**2 + a**2 / 25.d0
        rsqb = (z + rdist/2.d0)**2 + a**2 / 25.d0
```

```
        potential(nz) = potential(nz) - q / ( dsqrt(rsqa) + 1.d0 / a - a / 5.d0) &
                                      - q / ( dsqrt(rsqb) + 1.d0 / a - a / 5.d0)

        z = z + deltaz              ! next grid position in z-direction
      enddo

    end subroutine potentials_softmolecule

!-----------------------------------------------------------------------

    subroutine potentials_softmolecule_fitfunction ( rdist , apar, charge)
      ! Improved molecular soft-core 1D potential, based on B. Feuerstein
      ! (see Phys. Rev. A 67 - 043405 (2003)) using a softening function
      ! a(R) in order to reproduce the 1_sigma_g potential curve in 1D
      ! correctly. The potential of one nucleus is given by
      ! V(z) = - Q / ( sqrt(z^2 + (a/b)^2) + 1/a - a/b ) with a=a(R), and b=5
      !
      ! This version uses a fitting function

      real*8, intent(in)             :: rdist
      real*8, intent(in), optional :: apar
      real*8, intent(in), optional :: charge
      integer                        :: nz
      real*8                         :: z, q, a, rsqa, rsqb

      ! If no charge is given, we assume hydrogen. Softening is only adjusted to 1D H2+ molecule!!!
      if (present(charge)) then
        q = charge
      else
        q = 1.d0
      endif

      ! Use the analytical fitting function to obtain the softening parameter, unless specified directly
      if (present(apar)) then
        a = apar
      else
! a = 0.69293d0 * dexp(-rdist/1.47909d0) + 0.89989d0 ! our a(R) fitting function - original reproducing better for R=0..9
        a = 0.69356d0 * dexp(-rdist/1.48948d0) + 0.8984d0     ! our a(R) fitting function - better interpolation for larger distances
      endif

      z = zmin
      do nz=1, nzstep

        rsqa = (z - rdist/2.d0)**2 + a**2 / 25.d0
        rsqb = (z + rdist/2.d0)**2 + a**2 / 25.d0

        potential(nz) = potential(nz) - q / ( dsqrt(rsqa) + 1.d0 / a - a / 5.d0) &
                                      - q / ( dsqrt(rsqb) + 1.d0 / a - a / 5.d0)

        z = z + deltaz              ! next grid position in z-direction
      enddo

    end subroutine potentials_softmolecule_fitfunction

!-----------------------------------------------------------------------

    subroutine potentials_soft_diatomic_molecule( rdist , apar, bpar, charge)
      ! Improved molecular soft-core 1D potential, based on B. Feuerstein
      ! (see Phys. Rev. A 67 - 043405 (2003)) using the softening functions
      ! a(R) in order to reproduce the 1_sigma_g and b(R) for the 1_sigma_u
      ! potential curves in 1D correctly. The potential of one nucleus is given by
      ! V(z) = - Q / ( sqrt(z^2 + (a/b)^2) + 1/a - a/b ) with a=a(R), and b=b(R)
      real*8, intent(in) :: rdist
      real*8, intent(in) :: apar, bpar
      real*8, intent(in) :: charge
      integer            :: nz
      real*8             :: z, rsqa, rsqb

      z = zmin
      do nz=1, nzstep

        rsqa = (z - rdist/2.d0)**2 + apar**2 / bpar**2
        rsqb = (z + rdist/2.d0)**2 + apar**2 / bpar**2
        potential(nz) = potential(nz) - charge / ( dsqrt(rsqa) + 1.d0 / apar - apar / bpar) &
                                      - charge / ( dsqrt(rsqb) + 1.d0 / apar - apar / bpar)

        z = z + deltaz              ! next grid position in z-direction
      enddo

    end subroutine potentials_soft_diatomic_molecule

!-----------------------------------------------------------------------

    subroutine potentials_softhelium(zpos, softpar, effcharge)
      ! adds a soft-core helium-atom at the given position to the potential. The
      ! potential is an active 1-electron potential for the outermost electron,
      ! which includes the screening from the inner electron.
```

```
      ! The softening parameter and the effective charge are optional and are adjusted
      ! to reproduce the ionization potential of the second electron for the ground state
      ! 1s2 and the first excited state 1s2s. They NEED TO GET ADJUSTED!!!!
      !!! the softening parameter is optional - the default value is 0.1635 for the ground state (E0 =2 a.u.)
      !!! use softpar = 0.2195 to adjust for the first excited state (E1 = 0.5 a.u.)
      real*8, intent(in)              :: zpos
      real*8, intent(in), optional :: softpar
      real*8, intent(in), optional :: effcharge
      integer                         :: nz
      real*8                          :: z, rsq
      real*8                          :: softparameter
      real*8                          :: charge

      ! softening parameter - in this 2D model ti reproduces first 2 states of excited electron correctly
      if (present(softpar)) then
        softparameter = softpar
      else
        softparameter = 0.0913640992d0
      endif

      ! effective charge - in this 2D model ti reproduces first 2 states of excited electron correctly
      if (present(effcharge)) then
        charge = effcharge
      else
        charge = 1.04365323d0
      endif

      z = zmin
      do nz=1, nzstep

        rsq = (z - zpos)**2
        potential(nz) = potential(nz) - charge / dsqrt(rsq + softparameter)     ! using effective charge for helium
        z = z + deltaz                ! next grid position in z-direction

      enddo

    end subroutine potentials_softhelium

!-----------------------------------------------------------------------

    subroutine potentials_efield(zint)
      ! adds a electric field gradient of the given intensity in x and z direction
      ! and might be used for example to model the effect of a laser field
      real*8, intent(in) :: zint
      integer            :: nz
      real*8             :: z

      ! add the electric field in z-direction
      if (zint.ne.0.d0) then
        z = zmin
        do nz=1, nzstep
          potential(nz) = potential(nz) + z*zint
          z = z + deltaz        ! next grid position in z
        enddo
      endif

    end subroutine potentials_efield

!-----------------------------------------------------------------------
! A B S O R B E R
!-----------------------------------------------------------------------

! 2005/11/21 - changed the absorber corners from the MIN function to multiplicative behaviour, since the old absorber
! type had a discontinuity in the derivative. Therefore the absorber is now stronger in the grid
! corners, and thus reflecting more, but therefore smooth.

    subroutine potentials_setabsorber_left (width, strength)
      ! sets the absorber (optical potential) on the left side of the grid (min z-coordinate)
      real*8, intent(in) :: width, strength
      integer            :: n
      real*8             :: maskvalue

      do n=1, nint(width / deltaz)
        maskvalue = ((width - (n-1)*deltaz) / width)**2 * strength
        maskvalue = exp(-maskvalue)
        absorber(n) = absorber(n) * maskvalue
      enddo
    end subroutine potentials_setabsorber_left

!-----------------------------------------------------------------------

    subroutine potentials_setabsorber_right (width, strength)
      ! sets the absorber (optical potential) on the right side of the grid (max z-coordinate)
      real*8, intent(in) :: width, strength
      integer            :: n
      real*8             :: maskvalue

      do n=1, nint(width / deltaz)
```

```
        maskvalue = ((width - (n-1)*deltaz) / width)**2 * strength
        maskvalue = exp(-maskvalue)
        absorber(nzstep+1 - n) = absorber(nzstep+1 - n) * maskvalue
      enddo
    end subroutine potentials_setabsorber_right

!------------------------------------------------------------------------

    subroutine potentials_setabsorber(width, strength)
      ! sets the absorber (optical potential) on both sides of the potential
      real*8, intent(in) :: width, strength

      call potentials_setabsorber_left (width, strength)
      call potentials_setabsorber_right (width, strength)

    end subroutine potentials_setabsorber

!------------------------------------------------------------------------
! P L O T T I N G
!------------------------------------------------------------------------

    subroutine potentials_plot
      use debug

      open(unit=500,file="potential.dat",status="replace",access="sequential",recl=1024)

      write(500,'(3e12.4,$)') potential(:), -log(absorber(:)), absorber(:)

      write(500,*)

      close(500)

      call debugmsg(5,2,"POTENTIALS_PLOT: current potential plotted to file")
    end subroutine potentials_plot

!------------------------------------------------------------------------

    subroutine potentials_readfromfile(filename)
      ! read the potential from a file. In the first column is the position while the second column gives
      ! the potential value.

      use debug

      character(len=*), intent(in) :: filename          ! potential file to read
      integer                      :: ios               ! the error for the I/O operation
      real*8                       :: pos, pot          ! position and potential information from the file
      integer                      :: nzpos             ! the grid point where we store the potential value
      logical                      :: warning           ! the general warning flag
      real*8                       :: gridspacing       ! the grid spacing in the file
      real*8                       :: lastpos           ! the previous grid position to find grid spacing
      real*8                       :: minpos, maxpos    ! the minimum and maximum positions of the potential in the file
      real*8                       :: minpv, maxpv      ! minimum/maximum position potential value

      call potentials_realzero                          ! lets start out with a clean sheet of potential array
      open(unit=501,file=trim(filename),access="sequential",recl=1024,status="old") ! open file in binary form to read
           wavefunction

      warning     = .false.                             ! no warning consition
      gridspacing = 0.d0                                ! and initialize this debugging value
      lastpos     = 0.d0                                ! same here
      minpos      = 1.e20                               ! something ridiculously big
      maxpos      = -1.e20                              ! and the same on the other side of the grid

      do
        read(501,*,IOSTAT=ios) pos, pot                 ! read one line
        if (ios.ne.0) exit                              ! exit the do loop if an error occured, i.e. end of file

        ! now we have the values from the file and can start assigning them to our internal array

        if (pos > maxpos) then                          ! check for the largest potential position
          maxpv = pot                                   ! we got a new maximum position potential value
          maxpos = pos                                  ! remember the largest potential position
        endif

        if (pos < minpos) then                          ! check for the smallest potential position
          minpv = pot                                   ! we got a new minimum position potential value
          minpos = pos                                  ! remember the minimum potential position
        endif

        if (pos < zmin) cycle                           ! next do cycle if out of lower bound
        if (pos > zmin + nzstep*deltaz) cycle           ! same if out of upper array bound

        nzpos = nint((pos - zmin) / deltaz + 1.d0)      ! find the right grid point
        if (abs(zmin + (nzpos-1)*deltaz - pos) < 1.d-6) potential(nzpos) = pot ! assign value, if we are close enough to a grid point

        if (lastpos.ne.0.d0) then                       ! do some simple checks
          if (gridspacing.eq.0.d0) gridspacing = pos - lastpos     ! calculate the used grid spacing
          if (abs(pos - lastpos - gridspacing) > 1.d-6) warning = .true.
```

```
        if (abs(gridspacing - deltaz) > 1.d-6) warning = .true.
      endif

      lastpos = pos
    enddo

    close(501)

    do nzpos=1, nzstep                          ! final loop through potential to adjust the outer values
      pos = zmin + (nzpos-1)*deltaz             ! current position

      if (pos < minpos) potential(nzpos) = minpv   ! adjust if position is out if file boundaries
      if (pos > maxpos) potential(nzpos) = maxpv   ! adjust if position is out if file boundaries

    enddo

    if (ios < 0) then
      if (warning) then
        call debugmsg(5,2,"POTENTIALS_READ: WARNING - potential read, but file does not match grid")
        pause
      else
        call debugmsg(5,2,"POTENTIALS_READ: data successfully read from file")
      endif
    else
      call debugmsg(5,2,"POTENTIALS_READ: ERROR while reading file")
      print *,"Error number: ", ios
      pause
    endif

  end subroutine potentials_readfromfile

!-------------------------------------------------------------------------

end module potentials
```

C.5 Wave Function Module (wfmath1D.f90)

```
!-----------------------------------------
! DEFINITION OF 2D WAVEFUNCTION ANALYSIS AND MATHEMATICS
!-----------------------------------------
! Author : Thomas Niederhausen
! Date : 6 Jun 2006
!-----------------------------------------

! usage:
!
! WFMATH_INIT(zdim,minz,maxz)
! This provides the module witht the information about the used grid.
!
! WFMATH_DONE
! This routine does nothing but should be called for compatibility
!
! WFMATH_SETMASS(zmass)
! sets the masses of the particles for the energy (heavier particles have more
! energy).
!
! WFMATH_OVERLAP(wf1,wf2)
! calculates the quantum mechanical overlap of the given two wavefunctions.
! The result of the FUNCTION is a complex number.
!
! WFMATH_DIPOLE(wf1,wf2)
! calculates the dipole matrix element ¡wf1—z—wf2¿
!
! WFMATH_ENERGY(wf)
! calculates the energ corresponding to the given wavefunction: E=¡Psi—H—Psi¿
!
! WFMATH_DIPOLEACC(wf)
! calculates the dipole acceleration of the given wavefunction
!
! WFMATH_NORMALIZE(wf)
! Calculates the norm of the given wavefunction and renormalizes it to 1.
!
! WFMATH_ZERO(psiin)
! Sets the whole wavefunction to constant 0.
!
! WFMATH_GAUSSIAN(psiin,centerz,widthz,pz,norm)
! Adds a gaussian wavepacket to the given wavefunction psiin at the specified
! center with the given width and initial momentum. The parameter norm is
! optinal and if not given a value of 1 is assumed.
!
! WFMATH_SAVETOFILE(psi,filename)
! saves the specified wavefunction in binary form to disk
!
! WFMATH_READFROMFILE(psi,filename)
```

```
! reads wavefunction from the given binary file in the wavefunction array.
!
! WFMATH_PLOT(psiin)
! plots for the given 1D wavefucntion psi the probability density in a ASCII file
!


module wfmath
  implicit none

  integer, private      :: nzstep              ! the dimenstion of the 1D potentials/wavefunctions
  real*8, private       :: deltaz              ! the grid spacing
  real*8, private       :: zmin                ! defines the origin (in a.u.) of the grid
  real*8, private       :: massz=1.0           ! the masses of particles in a.u.

  contains

!-------------------------------------------------------------------------------
! INITIALIZATION
!-------------------------------------------------------------------------------

  subroutine wfmath_init(zdim,minz,maxz)
    integer, intent(in) :: zdim              ! the dimension of the arrays
    real*8, intent(in)  :: minz, maxz        ! the size of the "playground"

    nzstep = zdim

    zmin   = minz
    deltaz = (maxz - minz) / (nzstep -1)
  end subroutine wfmath_init

!-------------------------------------------------------------------------------

  subroutine wfmath_done
    ! a do-nothing-subroutine for later compatibility
  end subroutine wfmath_done

!-------------------------------------------------------------------------------

  subroutine wfmath_allocate(psi)
    ! allocates the given wavefunction array
    use debug

    complex*16, dimension(:), pointer :: psi
    integer                           :: iAllocStatus

    allocate(psi(nzstep),stat=iAllocStatus)
      if (iAllocStatus /= 0) then
        print *,"ERROR: can't allocate wavefunction"
        stop
      endif

    call wfmath_zero(psi)
    call debugmsg(5,1,"WFMATH_ALLOCATE: wavefunction array allocated")
  end subroutine wfmath_allocate

!-------------------------------------------------------------------------------

  subroutine wfmath_deallocate(psi)
    ! allocates the given wavefunction array
    use debug

    complex*16, dimension(:), pointer :: psi
    integer                           :: iDeallocStatus

    deallocate(psi,stat=iDeallocStatus)
      if (iDeallocStatus /= 0) then
        print *,"ERROR: can't deallocate wavefunction"
        stop
      endif

    call debugmsg(5,1,"WFMATH_DEALLOCATE: wavefunction array destroyed")
  end subroutine wfmath_deallocate

!-------------------------------------------------------------------------------

  subroutine wfmath_setmass(zmass)
    ! sets the masses of the particles
    real*8, intent(in)              :: zmass

    massz = zmass  ! sets the first mass parameter

  end subroutine wfmath_setmass

!-------------------------------------------------------------------------------
! WAVEFUNCTION ANALYSIS
!-------------------------------------------------------------------------------
```

```
      function wfmath_overlap(wf1,wf2,centerz,widthz) result(overlap)
        ! calculating the overlap of the two given wf's inside the optional given integration areas
        ! algorithm can be improved by a better integration method!
        complex*16, intent(in), dimension(:) :: wf1, wf2
        real*8, intent(in), optional         :: centerz          ! the center of the integration window
        real*8, intent(in), optional         :: widthz           ! the full length of the integration window
        complex*16                           :: overlap
        integer                              :: nzstart, nzend   ! the integration grid boundary

        overlap = cmplx(0.d0,0.d0)                               ! initial function result in case of out-of-integration-area

        if (present(centerz).and.present(widthz)) then
          nzstart = int((centerz - zmin - widthz/2) / deltaz) + 1   ! the integration boundary in z-grid space
          nzend   = int((centerz - zmin + widthz/2) / deltaz) + 1
          if (nzstart.gt.nzstep) return                          ! get out - we run out on the right side
          if (nzend.lt.1) return                                 ! get out - we run out on the left side
          if (nzstart.lt.1) nzstart = 1                          ! correct for left side out of integration area
          if (nzend.gt.nzstep) nzend = nzstep                    ! correct for right side
        else
          nzstart = 1
          nzend   = nzstep
        endif

        ! calculate the quantum mechanical overlap of the two wavefunctions
        overlap = sum(conjg(wf1(nzstart:nzend))*wf2(nzstart:nzend))*deltaz
        return
      end function wfmath_overlap

    !-------------------------------------------------------------------------

      function wfmath_dipole(wf1,wf2) result(dipole)
        ! calculate the dipole element ¡wf1—z—wf2¿
        complex*16, intent(in), dimension(:) :: wf1, wf2
        complex*16                           :: dipole
        integer                              :: nz
        real*8                               :: zpos

        dipole = cmplx(0.d0, 0.d0)
        do nz=1, nzstep

          zpos = zmin + (nz-1)*deltaz
          dipole = dipole + conjg(wf1(nz))*zpos*wf2(nz)*deltaz

        end do

        return
      end function wfmath_dipole

    !-------------------------------------------------------------------------

      function wfmath_squarepole(wf1,wf2) result(squaredipole)
        ! calculate the squared dipole element ¡wf1—z^2—wf2¿
        complex*16, intent(in), dimension(:) :: wf1, wf2
        complex*16                           :: squaredipole
        integer                              :: nz
        real*8                               :: zpos

        squaredipole = cmplx(0.d0, 0.d0)
        do nz=1, nzstep

          zpos = zmin + (nz-1)*deltaz
          squaredipole = squaredipole + conjg(wf1(nz))*(zpos**2)*wf2(nz)*deltaz

        end do

        return
      end function wfmath_squarepole

    !-------------------------------------------------------------------------

      function wfmath_width(wf) result(width)
        ! calculate the width of a wave packet, i.e. Delta R = sqrt(¡R¿^2 - ¡R^2¿)
        complex*16, intent(in), dimension(:) :: wf
        real*8                               :: width

        width = dsqrt( dreal(wfmath_squarepole(wf,wf)) - dreal(wfmath_dipole(wf,wf))**2 )
        return
      end function wfmath_width

    !-------------------------------------------------------------------------

      function wfmath_energy(wf) result(energy)
        ! calculates the energy corresponding to the given wavefunction: E=¡Psi—H—Psi¿
        ! using 5-point formula for H=-1/(2*mass) * (d^2/dx^2 + d^2/dz^2) + V
        ! 5-point formula: f" = 1/(12 Delta^2) * (- f_-2 + 16f_-1 - 30f_0 + 16f_1 - f_2)
        use potentials
        complex*16, intent(in), dimension(:) :: wf                 ! the input wavefunction we calculate the energy from
```

```
      complex*16                          :: energy              ! the functional result for the energy
      complex*16                          :: epot, ekinz         ! contributions to the total energy

      ! splitting operator into sum: E=-1/2m ¡Psi—pz2̂—Psi¿ + ¡Psi—V—Psi¿
      ! First step: Potential energy Epot = ¡Psi—V—Psi¿

      epot = sum(conjg(wf)*wf*potential)*deltaz

      ! Second: kinetic energy

      ekinz = - 1 / (24*massz) / deltaz * &
        sum(conjg(wf(3:nzstep-2)) * &
        ( -wf(1:nzstep-4)+16*wf(2:nzstep-3)-30*wf(3:nzstep-2)+16*wf(4:nzstep-1)-wf(5:nzstep) ) )


      ! Third: return the total (sum) energy
      energy = epot + ekinz

      return
    end function wfmath_energy

  !-----------------------------------------------------------------------

    function wfmath_momentum(wf) result(momentum)
      ! calculates the momentum corresponding to the given wavefunction ¡p¿
      ! using 5-point formula for the p operator
      ! 5-point formula: f' = 1/(12 Delta) * (f_-2 - 8f_-1 + 8_1 - f_2)
      use potentials
      complex*16, intent(in), dimension(:) :: wf              ! the input wavefunction we calculate the energy from
      complex*16                           :: momentum        ! the functional result for the energy
      complex*16, parameter                :: iu = (0.d0,1.d0)  ! imaginary unit

      momentum = 1.d0 / (12.d0*iu*deltaz) &
        *sum(conjg(wf(3:nzstep-2)) * &
        ( wf(1:nzstep-4)-8*wf(2:nzstep-3)+8*wf(4:nzstep-1)-wf(5:nzstep) ) )

      return
    end function wfmath_momentum

  !-----------------------------------------------------------------------

    function wfmath_dipoleacc(wf) result( dipoleacceleration )
      ! calculates the dipole acceleration of the given wavefunction in z direction
      ! using Ehrenfest's theorem: a = d2̂/dt2̂ ¡r¿ = - ¡psi—grad V—psi¿
      ! and 5-pont formula: f' = 1/(12*Delta) * (f_-2 - 8f_-1 + 8f_1 - f_2) *** this is correct - no f_0!!!
      use potentials
      complex*16, intent(in), dimension(:) :: wf              ! the input wavefunction
      complex*16                           :: dipoleacceleration  ! the calculated dipole moment

      dipoleacceleration = - 1 / (12*massz) * &
        sum(conjg(wf(3:nzstep-2)) * &
        ( potential(1:nzstep-4)-8*potential(2:nzstep-3)+8*potential(4:nzstep-1)-potential(5:nzstep) ) * &
        wf(3:nzstep-2) )

      return
    end function wfmath_dipoleacc

  !-----------------------------------------------------------------------

    function wfmath_dipoleacc_test(wf) result( dipoleacceleration )
      ! calculates the dipole acceleration of the given wavefunction in z direction
      ! using Ehrenfest's theorem: a = d2̂/dt2̂ ¡r¿ = - ¡psi—grad V—psi¿
      ! and 5-pont formula: f' = 1/(12*Delta) * (f_-2 - 8f_-1 + 8f_1 - f_2) *** this is correct - no f_0!!!
      use potentials
      complex*16, intent(in), dimension(:) :: wf              ! the input wavefunction
      complex*16                           :: dipoleacceleration  ! the calculated dipole moment
      real*8                               :: temp
      integer                              :: nz

      dipoleacceleration = 0

      do nz=3, nzstep-2
        temp = potential(nz-2) - 8*potential(nz-1) + 8*potential(nz+1) - potential(nz+2)
        dipoleacceleration = dipoleacceleration - temp * (cdabs(wf(nz))**2) / 12.d0
      enddo

      return
    end function wfmath_dipoleacc_test

  !-----------------------------------------------------------------------

    subroutine wfmath_normalize(wf)
      ! nomalizing the given wavefunction
      complex*16, intent(inout), dimension(:) :: wf
      complex*16                              :: wfnorm

      wfnorm = wfmath_overlap(wf,wf)      ! calculate the norm of the whole wavefunction
```

```
310       wf = wf / sqrt(wfnorm)                ! and divide everything by the square root of the norm -¿ normalize...
        end subroutine wfmath_normalize

      !-------------------------------------------------------------------------

315     function wfmath_norm(wf,centerz,widthz) result(norm)
          ! calculates the norm of the given wavefunction. The given optional parameters for center and width of
          ! the integration window allow to look only for the norm at a specified position, i.e. to find the norm
          ! at the target for example
          complex*16, intent(in), dimension(:) :: wf
320       real*8, intent(in), optional         :: centerz    ! the center of the integration window
          real*8, intent(in), optional         :: widthz     ! the full length of the integration window
          real*8                               :: norm       ! thats our function result

          norm = abs(wfmath_overlap(wf,wf,centerz,widthz))   ! calculate the norm
325       return
        end function wfmath_norm

      !-------------------------------------------------------------------------
      ! INITIAL WAVEFUNCTIONS
330   !-------------------------------------------------------------------------

        subroutine wfmath_zero(psiin)
          ! clears a wavefunction and sets it zero everywhere in space
          complex*16, intent(out), dimension(:) :: psiin
335
          psiin = cmplx(0.d0,0.d0)
        end subroutine wfmath_zero

      !-------------------------------------------------------------------------
340
        subroutine wfmath_gaussian(psiin,centerz,widthz,pz,norm)
          ! places a gaussian wavepacket to the given array
          complex*16, intent(inout), dimension(:) :: psiin
          real*8, intent(in)                      :: centerz          ! the center of the wavepacket
345       real*8, intent(in)                      :: widthz           ! the width of the wavepacket
          real*8, intent(in)                      :: pz               ! momentum
          real*8, intent(in), optional            :: norm             ! the norm of the wavefunction, otherwise 1
          real*8                                  :: z
          integer                                 :: nz
350       real*8                                  :: rvalue
          complex*16                              :: cvalue
          real*8, parameter                       :: pi = 3.14159265358979323846264 3d0

          z=zmin
355       do nz=1, nzstep

            rvalue = exp( -((z-centerz)/widthz)**2 /2 ) / (2*pi*widthz)
            cvalue = rvalue * cdexp( cmplx(0.0,1.0)*(pz*z) )
            psiin(nz) = psiin(nz) + cvalue
360
            z = z + deltaz                ! next grid position in z-direction
          enddo

      ! call wfmath_normalize(psiin) ! normalize first
365
          if (present(norm)) then
            call wfmath_normalize(psiin)        ! normalize first
            psiin = psiin * sqrt(norm)    ! in case somebody wants a different normalization
          endif
370
        end subroutine wfmath_gaussian

      !-------------------------------------------------------------------------
      ! FILE OPERATIONS
375   !-------------------------------------------------------------------------

        subroutine wfmath_savetofile(psi,filename)
          use debug
          complex*16, intent(in), dimension(:) :: psi
380       character(len=*), intent(in)         :: filename

          open(unit=601,file=filename,form="unformatted",status="replace",access="sequential") ! open file in binary form to write
               output
          write(601) psi                                                ! write the wavefunction to disk
          close(601)                                                    ! good bye, file
385
          call debugmsg(5,4,"WFMATH_SAVETOFILE: Wavefunction saved to disk")

        end subroutine wfmath_savetofile

390
      !-------------------------------------------------------------------------

        subroutine wfmath_readfromfile(psi,filename, zshift )
          use debug
395       complex*16, intent(out), dimension(:) :: psi
          character(len=*), intent(in)          :: filename
```

```
    real*8, intent(in), optional          :: zshift
    integer                               :: shift

    open(unit=602,file=filename,form='unformatted',access='direct',recl=16,status="old") ! open file in binary form to read
        wavefunction
    read(602) psi                                                ! read the wavefunction from the disk
    close(602)                                                   ! good bye, file

    if (present( zshift )) then                                  ! shift the matrix in z-direction
      shift = nint( zshift / deltaz )                            ! shift in gridpoints for the wavefunction
      if (( deltaz * shift ) .ne. zshift ) print *,"WARNING: WFMATH_READFROMFILE performed a mis-quantized wavefunction shift!"
      psi=eoshift( psi,- shift )                                 ! shift the wavefunction
    endif

    call debugmsg(5,4,"WFMATH_READFROMFILE: Wavefunction successfully read from file")

  end subroutine wfmath_readfromfile

!-----------------------------------------------------------------------

  subroutine wfmath_loadfromfile(psi,filename,szmin,lzstep , shiftz )
    ! this routine reads a wavefunction from the given file and truncates or extends this wavefunction to the given
    ! wavefucntion storage array.
    use debug
    complex*16, intent(out), dimension(:) :: psi
    character(len=*), intent(in)          :: filename
    real*8, intent(in)                    :: szmin               ! the starting point in the file
    integer, intent(in)                   :: lzstep              ! number of gridpoints in the file
    real*8, intent(in), optional          :: shiftz              ! shift of the wavefunction in a.u.

    complex*16, dimension(:), allocatable :: locpsi              ! stores the wavefuntion read from file
    integer                               :: iAllocStatus        ! status of file operations
    integer                               :: lszmin              ! the source starting point for copying
    integer                               :: ltzmin              ! the target starting point for copying
    integer                               :: lznum               ! the number of points to copy
    real*8                                :: temp                ! guess what...
    integer                               :: itemp               ! guess again...

    ! 1.) calculate the grid area that we have to copy
    temp = szmin - zmin
    if (present( shiftz )) temp = temp + shiftz
    itemp = nint(temp / deltaz)                                  ! then target grid coordinate in z where the origin of source is
    if (itemp.lt.0) then
      ltzmin = 1
      lszmin = - itemp + 1
      lznum = min(nzstep,( lzstep + itemp))                      ! the number of points to copy
    else
      lszmin = 1
      ltzmin = itemp + 1
      lznum = min(lzstep,( nzstep - itemp))                      ! the number of points to copy
    endif

    ! 2.) clear the output wavefunction and check if parameters in 1.) let us do some work...
    call wfmath_zero(psi)                                        ! clear the wavefunction

    if (lznum.le.0) return                                       ! done - we have nothing to do today...

    ! 3.) allocate array for storing the file wavefunction
    allocate( locpsi( lzstep ),stat=iAllocStatus)
    if ( iAllocStatus /= 0) then
      print *,"ERROR: can't allocate local wavefunction array LOCPSI"
      stop
    endif
    call debugmsg(10,4,"WFMATH_LOADFROMFILE: LOCPSI allocated")

    ! 4.) read wavefucntion from disk
    open(unit=603,file=filename,form="binary",recl=16,status="old")  ! open file in binary form to read wavefunction
    read(603) locpsi                                             ! read the wavefunction from the disk
    close(603)                                                   ! good bye, file

    ! 5.) copy the given area into the target (output) wavefunction
    psi(ltzmin:ltzmin+lznum-1) = locpsi(lszmin:lszmin+lznum-1)

    ! 6.) deallocate array for storing the file wavefunction
    deallocate(locpsi,stat=iAllocStatus)
    if ( iAllocStatus /= 0) then
      print *,"ERROR: can't deallocate local wavefunction array LOCPSI"
      stop
    endif
    call debugmsg(10,4,"WFMATH_LOADFROMFILE: LOCPSI destroyed")

  end subroutine wfmath_loadfromfile

!-----------------------------------------------------------------------

  subroutine wfmath_loadfromfile2(psi,filename,szmin,lzstep , shiftz )
    ! this routine reads a wavefunction from the given file and truncates or extends this wavefunction to the given
    ! wavefucntion storage array.
```

```
      ! Note: Slightly modified routine due to compiler upgrade incompatibility of the binary file
485   use debug
      complex*16, intent(out), dimension(:) :: psi
      character(len=*), intent(in)          :: filename
      real*8, intent(in)                    :: szmin          ! the starting point in the file
      integer, intent(in)                   :: lzstep         ! number of gridpoints in the file
490   real*8, intent(in), optional          :: shiftz         ! shift of the wavefunction in a.u.

      complex*16, dimension(:), allocatable :: locpsi         ! stores the wavefuntion read from file
      integer                               :: iAllocStatus   ! status of file operations
      integer                               :: lszmin         ! the source starting point for copying
495   integer                               :: ltzmin         ! the target starting point for copying
      integer                               :: lznum          ! the number of points to copy
      real*8                                :: temp           ! guess what...
      integer                               :: itemp          ! guess again...

500   ! 1.) calculate the grid area that we have to copy
      temp = szmin - zmin
      if (present(shiftz)) temp = temp + shiftz
      itemp = nint(temp / deltaz)                             ! then target grid coordinate in z where the origin of source is
      if (itemp.lt.0) then
505     ltzmin = 1
        lszmin = - itemp + 1
        lznum = min(nzstep,(lzstep + itemp))                  ! the number of points to copy
      else
        lszmin = 1
510     ltzmin = itemp + 1
        lznum = min(lzstep,(nzstep - itemp))                  ! the number of points to copy
      endif

      ! 2.) clear the output wavefunction and check if parameters in 1.) let us do some work...
515   call wfmath_zero(psi)                                   ! clear the wavefunction

      if (lznum.le.0) return                                  ! done - we have nothing to do today...

      ! 3.) allocate array for storing the file wavefunction
520   allocate(locpsi(lzstep),stat=iAllocStatus)
      if (iAllocStatus /= 0) then
        print *,"ERROR: can't allocate local wavefunction array LOCPSI"
        stop
      endif
525   call debugmsg(10,4,"WFMATH_LOADFROMFILE: LOCPSI allocated")

      ! 4.) read wavefucntion from disk
      open(unit=603,file=filename,form="unformatted",status="old",access="sequential") ! open file in binary form to write input
      read(603) locpsi                                        ! read the wavefunction from the disk
530   close(603)                                              ! good bye, file

      ! 5.) copy the given area into the target (output) wavefunction
      psi(ltzmin:ltzmin+lznum-1) = locpsi(lszmin:lszmin+lznum-1)

535
      ! 6.) deallocate array for storing the file wavefunction
      deallocate(locpsi,stat=iAllocStatus)
      if (iAllocStatus /= 0) then
        print *,"ERROR: can't deallocate local wavefunction array LOCPSI"
540     stop
      endif
      call debugmsg(10,4,"WFMATH_LOADFROMFILE: LOCPSI destroyed")

    end subroutine wfmath_loadfromfile2
545
  !-------------------------------------------------------------------------
  ! PLOTTING
  !-------------------------------------------------------------------------

550   subroutine wfmath_plot(psi)
        use debug
        complex*16, intent(in), dimension(:) :: psi

        open(unit=600,file="wavefunction.dat",status="replace",access="sequential",recl=1024)
555
        write(600,'(e12.4,$)') abs(psi(:))**2
        write(600,*)

        close(600)
560
        call debugmsg(5,2,"WFMATH_PLOT: given wavefunction plotted to file")
      end subroutine wfmath_plot

  !-------------------------------------------------------------------------
565
  end module wfmath
```

C.6 Diagonalization Module (diag.f90)

```
!------------------------------------------
! DEFINITION OF 1D DIAGONALIZATION ROUTINE
!------------------------------------------
! Author : Thomas Niederhausen
! Sources : Numerical Recipies (DTQLI) for tridiagonal matrix solver
! Date : 06 Oct 2006
!------------------------------------------

! usage:
!
! DIAG_INIT(zdim,dz,particlemass)
! This provides the module with the information about the used grid.
!
! DIAG_DONE
! Does nothing but is here for compatibility and should be called anyhow.
!
! DIAG_RELEASE(TObj)
! Releases either a TEnergies or TStates object if allocated
!
! DIAG_DIAGONALIZE(potential, TObject(TStates or TEnergies))
! diagonalizes the given potential and provides the energies of the bound states
! in TEnergies, or TStates for the corresponding normalized eigenstates of the potential,
! whichever type of variable is given.

module diag
  implicit none

  type :: TEnergies                                    ! declare a type array for bound state energies
    integer                :: numbound                 ! number of bound states
    real*8, pointer        :: energyarr(:)             ! the array with the bound energies
  end type

  type :: TStates                                      ! declare a type array for bound wave functions
    integer                :: numbound                 ! number of bound states, identical to TEnergies
    real*8, pointer        :: energyarr(:)             ! the array with the bound energies
    complex*16, pointer :: wavefunctions(:,:)          ! the array with the wave functions
  end type

  integer, private         :: nzstep                   ! the dimenstion of the 1D potentials/wavefunctions
  real*8, private          :: deltaz                   ! the grid spacing
  real*8, private          :: mass                     ! the mass of particles in a.u.

  interface diag_release                               ! release the objects TEnergies or TStates if allocated
    module procedure diag_release_TEnergies, diag_release_TStates   ! define different procedures for the types
  end interface

  interface diag_diagonalize                           ! the working horse. diagonalize the potential array.
    module procedure diag_diagonalize_TEnergies, diag_diagonalize_TStates   ! define different procedures for the types
  end interface

  interface diag_plot                                  ! Plot the results to file
    module procedure diag_plot_TEnergies, diag_plot_TStates ! define different procedures for the types
  end interface

  contains

!------------------------------------------------------------------------------
! I N I T I A L I Z A T I O N
!------------------------------------------------------------------------------

  subroutine diag_init(zdim,dz,particlemass)
    use debug

    integer, intent(in) :: zdim
    real*8, intent(in)  :: dz, particlemass

    nzstep = zdim
    deltaz = dz
    mass   = particlemass

    call debugmsg(5,1,"DIAG_INIT: diagonalization routine initialized.")

  end subroutine diag_init

!------------------------------------------------------------------------------

  subroutine diag_done
    use debug

    call debugmsg(5,1,"DIAG_DONE: diagonalization routine closed.")

  end subroutine diag_done

!------------------------------------------------------------------------------
```

```
85
      subroutine diag_release_TEnergies(energies)
        use debug
        type(TEnergies), intent(inout) :: energies          ! we want to release this object
        integer                        :: iDeallocStatus    ! return value
90
        energies%numbound = 0                               ! we have zero bound states from now on
        if ( associated( energies%energyarr)) then
          deallocate(energies%energyarr,stat=iDeallocStatus)
          if (iDeAllocStatus /= 0) then
95          print *,"ERROR: can't deallocate DIAGONALIZATION ENERGY array"
            stop
          endif
          call debugmsg(10,1,"DIAG_RELEASE: DIAGONALIZATION ENERGY array destroyed")
        endif
100     call debugmsg(5,1,"DIAG_RELEASE: object destroyed")

      end subroutine diag_release_TEnergies

    !-------------------------------------------------------------------------
105
      subroutine diag_release_TStates( states )
        use debug
        type(TStates), intent(inout) :: states              ! we want to release this object
        integer                      :: iDeallocStatus      ! return value
110
        states%numbound = 0                                 ! we have zero bound states from now on
        if ( associated( states%energyarr)) then
          deallocate(states%energyarr,stat=iDeallocStatus)
          if (iDeAllocStatus /= 0) then
115         print *,"ERROR: can't deallocate DIAGONALIZATION ENERGY array"
            stop
          endif
          call debugmsg(10,1,"DIAG_RELEASE: DIAGONALIZATION ENERGY array destroyed")
        endif
120
        if ( associated( states%wavefunctions)) then
          deallocate(states%wavefunctions,stat=iDeallocStatus)
          if (iDeAllocStatus /= 0) then
            print *,"ERROR: can't deallocate DIAGONALIZATION WAVE FUNCTION array"
125         stop
          endif
          call debugmsg(10,1,"DIAG_RELEASE: DIAGONALIZATION WAVE FUNCTION array destroyed")
        endif
        call debugmsg(5,1,"DIAG_RELEASE: object destroyed")
130
      end subroutine diag_release_TStates

    !-------------------------------------------------------------------------
    ! D I A G O N A L I Z A T I O N
135 !-------------------------------------------------------------------------

      subroutine diag_diagonalize_TEnergies(potential, energies)
      ! this routine diagonaalizes the 1d hamiltonean given only by it's potential
        use debug
140     real*8, pointer                :: potential(:)          ! the potential array
        type(TEnergies), intent(inout) :: energies              ! type for the returned energies
        integer                        :: iAllocStatus          ! return value
        real*8, allocatable            :: subsuper(:), diagarr(:) ! the diagonal, sub- and super-diagonals of the Hamiltonian
        real*8                         :: mine                  ! minimum bound energy
145
        if ( associated( energies%energyarr)) then
          deallocate(energies%energyarr,stat=iAllocStatus)
          if ( iAllocStatus /= 0) then
            print *,"ERROR: can't deallocate DIAGONALIZATION ENERGY array"
150         stop
          endif
          call debugmsg(10,1,"DIAG_DIAGONALIZE: DIAGONALIZATION ENERGY array destroyed")
        endif
        energies%numbound = 0                                   ! first we start with 0 bound states
155
        allocate(subsuper(nzstep),diagarr(nzstep),stat=iAllocStatus)
        if ( iAllocStatus /= 0) then
          print *,"ERROR: can't allocate tridiagonal Hamiltonian array"
          stop
160     endif
        call debugmsg(10,1,"DIAG_DIAGONALIZE: tridiagonal Hamiltonian array allocated")

        diagarr  = potential + 1.d0/(mass*deltaz**2)        ! the diagonal array of the Hamiltonian
        subsuper = -1.d0/(2.d0 * mass * deltaz**2)          ! the sub- and super-diagonals of H
165
        call debugmsg(5,1,"DIAG_DIAGONALIZE: diagonalizing potential - only energies...")
        call dtqli(diagarr,subsuper)                        ! call the routine from Numerical Recipes

        mine = min(potential(1), potential(nzstep))         ! find the minimum bound state energy
170     energies%numbound = count(diagarr < mine)           ! count the number of bound states

        allocate(energies%energyarr(energies%numbound),stat=iAllocStatus)
```

```
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't allocate DIAGONALIZATION ENERGY array"
        stop
      endif
      call debugmsg(10,1,"DIAG_DIAGONALIZE: DIAGONALIZATION ENERGY array allocated")

      energies%energyarr = pack(diagarr, diagarr<mine) ! copy only the corresponding energies into the output

      deallocate(subsuper,diagarr,stat=iAllocStatus)
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't deallocate tridiagonal Hamiltonian array"
        stop
      endif
      call debugmsg(10,1,"DIAG_DIAGONALIZE: tridiagonal Hamiltonian array released")

    end subroutine diag_diagonalize_TEnergies

!-----------------------------------------------------------------------

    subroutine diag_diagonalize_TStates( potential , states )
    ! this routine diagonaalizes the 1d hamiltonean given only by it's potential. This time WITH wave functions
      use debug
      use wfmath
      real*8, pointer                 :: potential (:)              ! the potential array
      type(TStates), intent(inout)    :: states                     ! type for the returned wave functions
      integer                         :: iAllocStatus               ! return value
      real*8, allocatable             :: subsuper(:) , diagarr (:)  ! the diagonal, sub- and super-diagonals of the Hamiltonian
      real*8, allocatable             :: wavefunctions (:,:)        ! temporary wave function array
      real*8                          :: mine                       ! minimum bound energy
      integer                         :: n

      if ( associated ( states%energyarr)) then
        deallocate(states%energyarr,stat=iAllocStatus)
        if ( iAllocStatus /= 0) then
          print *,"ERROR: can't deallocate DIAGONALIZATION ENERGY array", iAllocStatus
          stop
        endif
        call debugmsg(10,1,"DIAG_DIAGONALIZE: DIAGONALIZATION ENERGY array destroyed")
      endif

      if ( associated ( states%wavefunctions)) then
        deallocate(states%wavefunctions,stat=iAllocStatus)
        if ( iAllocStatus /= 0) then
          print *,"ERROR: can't deallocate DIAGONALIZATION WAVE FUNCTION array", iAllocStatus
          stop
        endif
        call debugmsg(10,1,"DIAG_DIAGONALIZE: DIAGONALIZATION WAVE FUNCTION array destroyed")
      endif

      states%numbound = 0                                 ! first we start with 0 bound states

      allocate(subsuper(nzstep) , diagarr( nzstep) ,stat=iAllocStatus)
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't allocate tridiagonal Hamiltonian array"
        stop
      endif
      call debugmsg(10,1,"DIAG_DIAGONALIZE: tridiagonal Hamiltonian array allocated")

      allocate(wavefunctions(nzstep, nzstep) ,stat=iAllocStatus)
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't allocate temporary wave function array"
        stop
      endif
      call debugmsg(10,1,"DIAG_DIAGONALIZE: temporary wave function array allocated")

      diagarr  = potential + 1.d0/(mass*deltaz**2)      ! the diagonal array of the Hamiltonian
      subsuper = -1.d0/(2.d0 * mass * deltaz**2)        ! the sub- and super-diagonals of H
      wavefunctions = 0.d0
      do n=1, nzstep
        wavefunctions(n,n) = 1.d0                       ! fill with identity matrix
      enddo

      call debugmsg(5,1,"DIAG_DIAGONALIZE: diagonalizing potential - energies and wave functions...")
      call  dtqli ( diagarr, subsuper, wavefunctions)   ! call the routine from Numerical Recipes

      mine = min(potential(1), potential ( nzstep))     ! find the minimum bound state energy
      states%numbound = count(diagarr < mine)           ! count the number of bound states

      allocate( states%energyarr(states%numbound),stat=iAllocStatus)
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't allocate DIAGONALIZATION ENERGY array"
        stop
      endif
      call debugmsg(10,1,"DIAG_DIAGONALIZE: DIAGONALIZATION ENERGY array allocated")

      allocate( states%wavefunctions(states%numbound,nzstep),stat=iAllocStatus)
      if ( iAllocStatus /= 0) then
        print *,"ERROR: can't allocate DIAGONALIZATION WAVE FUNCTION array"
```

```
      stop
    endif
    call debugmsg(10,1,"DIAG_DIAGONALIZE: DIAGONALIZATION WAVE FUNCTION array allocated")

    states%energyarr = pack(diagarr, diagarr<mine)    ! copy only the corresponding energies into the output
    do n=1, nzstep
      states%wavefunctions(:,n) = pack(wavefunctions(n,:), diagarr<mine)    ! and copy the wave functions
    enddo

    do n=1, states%numbound
      call wfmath_normalize(states%wavefunctions(n,:)) ! we like normalized wave functions
    enddo

    deallocate(subsuper,diagarr,wavefunctions,stat=iAllocStatus)
    if ( iAllocStatus /= 0) then
      print *,"ERROR: can't deallocate diagonalization arrays"
      stop
    endif
    call debugmsg(10,1,"DIAG_DIAGONALIZE: diagonalization arrays released")

  end subroutine diag_diagonalize_TStates

!-----------------------------------------------------------------------

  subroutine diag_plot_TEnergies(energies,filename)
    use debug
    use strings
    type(TEnergies), intent(in) :: energies          ! this is what we plot
    character(len=*), intent(in) :: filename
    integer                      :: n

    open(unit=900,file=trim(concat(filename,"_energies.dat")),status="replace",access="sequential")
    write(900,'(3A12)') "State","Energy_au","Energy_eV"

    do n=1,energies%numbound
      write(900,'(I12,2E12.4)') n, energies%energyarr(n),energies%energyarr(n)*27.2114d0
    enddo

    close(900)
    call debugmsg(5,2,"DIAG_PLOT: energies plotted to file")
  end subroutine diag_plot_TEnergies

!-----------------------------------------------------------------------

  subroutine diag_plot_TStates(states,filename)
    use debug
    use strings
    type(TStates), intent(in)    :: states          ! this is what we plot
    character(len=*), intent(in) :: filename
    integer                      :: n

    open(unit=901,file=trim(concat(filename,"_wavefunctions.dat")),status="replace",access="sequential")

    write(901,'(e12.4,$)') states%energyarr(:)
    do n=1,nzstep
      write(901,'(e12.4,$)') cdabs( states%wavefunctions(:,n))**2
      write(901,*)
    enddo

    close(901)
    call debugmsg(5,2,"DIAG_PLOT: wave functions plotted to file")
  end subroutine diag_plot_TStates

!-----------------------------------------------------------------------

  function dpythag(a, b) result(c)
  ! using Phytagoras a^2 + b^2 = c^2 for a triangle, calculate c in a numerically stable way.
    real*8, intent(in) :: a, b
    real*8             :: absa, absb
    real*8             :: c

    absa=dabs(a)
    absb=dabs(b)
    if(absa.gt.absb)then
      c=absa*dsqrt(1.d0+(absb/absa)**2)
    else
      if(absb.eq.0.d0)then
        c=0.d0
      else
        c=absb*dsqrt(1.d0+(absa/absb)**2)
      endif
    endif
    return
  end function dpythag

!-----------------------------------------------------------------------
```

```
      subroutine dtqli(d,e,z)
      ! solver routine from Numerical Recipes for a tridiagonal linear equation
350     real*8, dimension(:), intent(inout)              :: d, e ! the diagonal and sub- super-diagonals
        real*8, dimension(:,:), intent(inout), optional :: z     ! returns the wave functions. Must identity matrix at call
        integer                                          :: n, np, i, iter, k, l , m
        real*8                                           :: b, c, dd, f, g, p, r, s

355     np = size(d)
        n = np
        if (present(z)) then
          write (*,*) 'DTQLI: diagonalization with eigenvectors'
        else
360       write (*,*) 'DTQLI: diagonalization, only eigenvalues'
        endif
        write (*,*) '       matrix is',np,'x',np

        do i=2,n
365       e(i-1)=e(i)
        enddo
        e(n)=0.d0

        do 15 l=1,n
370       iter=0
    1     do m=l,n-1
            dd=dabs(d(m))+dabs(d(m+1))
            if (dabs(e(m))+dd.eq.dd) goto 2
          enddo
375       m=n
    2     if (m.ne.l) then
            if (iter.eq.30) pause 'too many iterations in tqli'
            iter=iter+1
            g=(d(l+1)-d(l))/(2.d0*e(l))
380         r=dpythag(g,1.d0)
            g=d(m)-d(l)+e(l)/(g+sign(r,g))
            s=1.d0
            c=1.d0
            p=0.d0
385         do 14 i=m-1,l,-1
              f=s*e(i)
              b=c*e(i)
              r=dpythag(f,g)
              e(i+1)=r
390           if(r.eq.0.d0)then
                d(i+1)=d(i+1)-p
                e(m)=0.d0
                goto 1
              endif
395           s=f/r
              c=g/r
              g=d(i+1)-p
              r=(d(i)-g)*s+2.d0*c*b
              p=s*r
400           d(i+1)=g+p
              g=c*r-b
              ! omit lines from here ...
              if (present(z)) then
                do k=1,n
405               f=z(k,i+1)
                  z(k,i+1)=s*z(k,i)+c*f
                  z(k,i)=c*z(k,i)-s*f
                enddo
              endif
410
              ! to here when finding only eigenvalues.
    14      continue
            d(l)=d(l)-p
            e(l)=g
415         e(m)=0.d0
            goto 1
          endif
    15  continue
        return
      end subroutine dtqli

    end module diag
```

Bibliography

[Abramowitz 72] M. Abramowitz, and I. A. Stegun. *Handbook of Mathematical Functions*. U.S. Government Printing Office, Washington, D.C., 10th edition, (1972).

[Agostini 04] Pierre Agostini, and Louis F. DiMauro. *The physics of attosecond light pulses*. Rep. Prog. Phys., vol. **67**, pages 813–855, (2004).

[Alnaser 03] A. S. Alnaser, T. Osipov, E. P. Benis, A. Wech, B. Shan, C. L. Cocke, X. M. Tong, and C. D. Lin. *Rescattering double ionization of D_2 and H_2 by intense laser pulses*. Phys. Rev. Lett., vol. **91**, page 163002, (2003).

[Alnaser 04] A. S. Alnaser, X. M. Tong, T. Osipov, S. Voss, C. M. Maharjan, P. Ranitovic, B. Ulrich, B. Shan, Z. Chang, C. D. Lin, and C. L. Cocke. *Routes to Control of H_2 Coulomb Explosion in Few-Cycle Laser Pulses*. Phys. Rev. Lett., vol. **93**, page 183202, (2004).

[Alnaser 05] A. S. Alnaser, B. Ulrich, X. M. Tong, I. V. Litvinyuk, C. M. Maharjan, P. Ranitovic, T. Osipov, R. Ali, S. Ghimire, Z. Chang, C. D. Lin, and C. L. Cocke. *Simultaneous real-time tracking of wave packets evolving on two different potential curves in H_2^+ and D_2^+*. Phys. Rev. A, vol. **72**, page 030702, (2005).

[Alvarellos 88] José Alvarellos, and Horia Metiu. *The evolution of the wave function in a curve crossing problem computed by a fast Fourier transform method*. J. Chem. Phys., vol. **88**, pages 4957–4966, (1988).

[Ammosov 86] M. V. Ammosov, N. B. Delone, and V. P. Krainov. *Tunnel ionization of complex atoms and of atomic ions in an alternating electromagnetic field*. Zh. Eksp. Teor. Fiz., vol. **91**, pages 2008–2013, (1986). [Sov. Phys. JETP **64**, 1191–1194 (1986)].

[Anderson 95] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell. *Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor*. Science, vol. **269**, pages 198–201, (1995).

[Anglin 02] James R. Anglin, and Wolfgang Ketterle. *Bose-Einstein condensation of atomic gases*. Nature, vol. **416**, pages 211–218, (2002).

[Anis 06] Fatima Anis, V. Roudnev, R. Cabrera-Trujillo, and B. D. Esry. *Laser-assisted charge transfer in $He^{2+}+H$ collisions*. Phys. Rev. A, vol. **73**, page 043414, (2006).

[ARP] *ARPACK, collection of FORTRAN subroutines for solving large scale (sparse) eigenproblems*.
`http://www.caam.rice.edu/software/ARPACK/`.

[Balakrishnan 99] N. Balakrishnan, B. D. Esry, H. R. Sadeghpour, S. T. Cornett, and M. J. Cavagnero. *Quantum wave-packet dynamics of the photodissociation of LiF*. Phys. Rev. A, vol. **60**, pages 1407–1413, (1999).

[Baltuška 03] A. Baltuška, Th. Udem, M. Uiberacker, M. Hentschel, E. Goulielmakis, Ch. Gohle, R. Holzwarth, V. S. Yakovlev, A. Scrinzi, T. W. Hnsch, and F. Krausz. *Attosecond control of electronic processes by intense light fields*. Nature, vol. **421**, pages 611–615, (2003).

[Bandrauk 81] André D. Bandrauk, and Michael L. Sink. *Photodissociation in intense laser fields: Predissociation analogy*. J. Chem. Phys., vol. **74**, pages 1110–1117, (1981).

[Bandrauk 99] A. D. Bandrauk, and J. Ruel. *Charge-resonance-enhanced ionization of molecular ions in intense laser pulses: geometric and orientation effects.* Phys. Rev. A, vol. **59**, pages 2153–2162, (1999).

[Barone 77] S. R. Barone, M. A. Narcowich, and F. J. Narcowich. *Floquet theory and applications.* Phys. Rev. A, vol. **15**, pages 1109–1125, (1977).

[Bates 53] D. R. Bates, Kathleen Ledsham, and A. L. Stewart. *Wave Functions of the Hydrogen Molecular Ion.* Philos. Trans. R. Soc. Lond. A, vol. **246**, pages 215–240, (1953).

[Biersack 82] J. P. Biersack, and J. F. Ziegler. *Refined universal potentials in atomic collisions.* Nucl. Instr. Meth. B, vol. **194**, pages 93–100, (1982).

[Borisov 98] A. G. Borisov, A. K. Kazansky, and J. P. Gauyacq. *Finite Time Effect in the Charge Transfer Process during an Ion-Metal Surface Collision.* Phys. Rev. Lett., vol. **80**, pages 1996–1999, (1998).

[Borisov 99] A. G. Borisov, A. K. Kazansky, and J. P. Gauyacq. *Resonant charge transfer in ion-metal surface collisions: Effect of a projected band gap in the* $H^- - Cu(111)$ *system.* Phys. Rev. B, vol. **59**, pages 10935–10949, (1999).

[Boyd 00] John P. Boyd. *Chebyshev and Fourier Spectral Methods.* Dover, Mineola, NY, 2nd edition, (2000).

[Boyd 02] Robert W. Boyd. *Nonlinear Optics.* Elsevier - Academic Press, 2nd edition, (2002).

[Brabec 00] Thomas Brabec, and Ferenc Krausz. *Intense few-cycle laser fields: Frontiers of nonlinear optics.* Rev. Mod. Phys., vol. **72**, pages 545–591, (2000).

[Bransden 92] B. H. Bransden, and M. R. C. McDowell. *Charge Exchange in the Theory of Ion-Atom Collisions*. Clarendon Press, Oxford, (1992).

[Bransden 03] B. H. Bransden, and C. J. Joachain. *Physics of Atoms and Molecules*. Prentice-Hall, New York, 2nd edition, (2003).

[Brichta 06] J. P. Brichta, W.-K. Liu, A. A. Zaidi, A. Trottier, and J. H. Sanderson. *Comparison of ADK ionization rates as a diagnostic for selective vibrational level population measurement*. J. Phys. B, vol. **39**, pages 3769–3779, (2006).

[Brixner 04] T. Brixner, G. Krampert, T. Pfeifer, R. Selle, G. Gerber, M. Wollenhaupt, O. Graefe, C. Horn, D. Liese, and T. Baumert. *Quantum Control by Ultrafast Polarization Shaping*. Phys. Rev. Lett., vol. **92**, page 208301, (2004).

[Bucksbaum 90] P. H. Bucksbaum, A. Zavriyev, H. G. Muller, and D. W. Schumacher. *Softening of the H_2^+ molecular bond in intense laser fields*. Phys. Rev. Lett., vol. **64**, pages 1883–1886, (1990).

[Burke 91] P. G. Burke, P. Francken, and C. J. Joachain. *R-Matrix-Floquet theory of multiphoton processes*. J. Phys. B, vol. **24**, pages 761–790, (1991).

[Carnes 06] K. D. Carnes, C. L. Cocke, Z. Chang, I. Ben-Itzhak, H. V. Needham, and A. Rankin. *Picosecond Ion Pulses from an EN Tandem Created by a Femtosecond Ti:Sapphire Laser*. In CAARI 2006: 19th International Conference on the Application of Accelerators in Research and Industry. Fort Worth, TX, (2006).

[Chakraborty 04a] H. S. Chakraborty, T. Niederhausen, and U. Thumm. *Effects of the surface Miller index on the resonant neutralization of hydrogen anions near Ag surfaces*. Phys. Rev. A, vol. **69**, page 052901, (2004).

[Chakraborty 04b] H. S. Chakraborty, T. Niederhausen, and U. Thumm. *Resonant neutralization of H^- near Cu surfaces: Effects of the surface symmetry and ion trajectory.* Phys. Rev. A, vol. **70**, page 052903, (2004).

[Chakraborty 05] H. S. Chakraborty, T. Niederhausen, and U. Thumm. *On the effect of image states on the resonant neutralization of hydrogen anions near metal surfaces.* Nucl. Instr. Meth. B, vol. **241**, pages 43–47, (2005).

[Chelkowski 96] Szczepan Chelkowski, and Andre D. Bandrauk. *Wave-function splitting technique for calculating above-threshold ionization electron spectra.* Int. J. Quantum Chem., vol. **60**, pages 1685–1689, (1996).

[Chulkov 99] E. V. Chulkov, V. M. Silkin, and P. M. Echenique. *Image potential states on metal surfaces: binding energies and wave functions.* Surf. Sci., vol. **437**, pages 330–352, (1999).

[Cionga 01] A. Cionga, F. Ehlotzky, and G. Zloh. *Coherent phase control in electron scattering by hydrogen atoms in a bichromatic laser field.* J. Phys. B, vol. **34**, pages 2057–2069, (2001).

[Cocke 91] C. L. Cocke, and R. E. Olson. *Recoil ions.* Phys. Rep., vol. **205**, pages 153–219, (1991).

[COD] *CODATA recommended fundamental constants of the International Council for Science: Committee on Data for Science and Technology.* `http://physics.nist.gov/cuu/Constants/`.

[Cohen 86] James S. Cohen, and Giovanni Fiorentini. *Stripping of H^- in low-energy collisions with antiprotons: Classical-trajectory Monte Carlo calculation.* Phys. Rev. A, vol. **33**, pages 1590–1594, (1986).

[De Raedt 87] Hans De Raedt. *Product formula algorithms for solving the time dependent Schrödinger equation.* Comp. Phys. Rep., vol. **7**, pages 1–72, (1987).

[Dèbarre 86] A. Dèbarre, and P. Cahuzac. *Interpretation of the experimental cross section profile of a laser-induced charge exchange process.* J. Phys. B, vol. **19**, pages 3965–3973, (1986).

[Deumens 94] E. Deumens, A. Diz, R. Longo, and Y. Öhrn. *Time-dependent theoretical treatments of the dynamics of electrons and nuclei in molecular systems.* Rev. Mod. Phys., vol. **66**, pages 917–983, (1994).

[Dooley 03] P. W. Dooley, I. V. Litvinyuk, Kevin F. Lee, D. M. Rayner, M. Spanner, D. M. Villeneuve, and P. B. Corkum. *Direct imaging of rotational wavepacket dynamics of diatomic molecules.* Phys. Rev. A, vol. **68**, page 023406, (2003).

[Dörner 00] R. Dörner, V. Mergel, O. Jagutzki, L. Spielberger, J. Ullrich, R. Moshammer, and H. Schmidt-Böcking. *Cold target recoil ion momentum spectroscopy: a 'momentum microscope' to view atomic collision dynamics.* Phys. Rep., vol. **330**, pages 95–192, (2000).

[Dudley 06] John M. Dudley, Goëry Genty, and Stéphane Coen. *Supercontinuum generation in photonic crystal fiber.* Rev. Mod. Phys., vol. **78**, pages 1135–1184, (2006).

[Ehlotzky 98] F. Ehlotzky, A. Jaroń, and J. Z. Kamiński. *Electron-atom collisions in a laser field.* Phys. Rep., vol. **297**, pages 63–153, (1998).

[Einstein 17] A. Einstein. *Zur Quantentheorie der Strahlung.* Physikalische Zeitschrift, vol. **18**, pages 121–128, (1917).

[Ergler 06a] Th. Ergler, B. Feuerstein, A. Rudenko, K. Zrost, C. D. Schröter, R. Moshammer, and J. Ullrich. *Quantum-Phase Resolved Mapping of Ground-State Vibrational D_2 Wave Packets via Selective Depletion in Intense Laser Pulses*. Phys. Rev. Lett., vol. **97**, page 103004, (2006).

[Ergler 06b] Th. Ergler, A. Rudenko, B. Feuerstein, K. Zrost, C. D. Schröter, R. Moshammer, and J. Ullrich. *Spatiotemporal Imaging of Ultrafast Molecular Motion: Collapse and Revival of the D_2^+ Nuclear Wave Packet*. Phys. Rev. Lett., vol. **97**, page 193001, (2006).

[Ergler 06c] Thorsten Ergler. *Zeitaufgelöste Untersuchungen zur Fragmentationsdynamik von H_2 (D_2) in ultra-kurzen Laserpulsen*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Germany, (2006).

[Ermolaev 99] A. M. Ermolaev, I. V. Puzynin, A. V. Selin, and S. I. Vinitsky. *Integral boundary conditions for the time-dependent Schrödinger equation: Atom in a laser field*. Phys. Rev. A, vol. **60**, pages 4831–4845, (1999).

[Ermoshin 96] V. A. Ermoshin, and A. K. Kazansky. *Wave packet study of H^- decay in front of a metal surface*. Phys. Lett. A, vol. **218**, pages 99–104, (1996).

[Esry 97] B. D. Esry, Chris H. Greene, James P. Burke Jr., and John L. Bohn. *Hartree-Fock Theory for Double Condensates*. Phys. Rev. Lett., vol. **78**, pages 3594–3597, (1997).

[Feit 82] M. D. Feit, J. A. Fleck Jr., and A. Steiger. *Solution of the Schrödinger equation by a spectral method*. J. Comp. Phys., vol. **47**, pages 412–433, (1982).

[Ferrante 81] G. Ferrante, L. Lo Cascio, and B. Spagnolo. *Laser-assisted symmetric charge transfer in atomic collisions.* J. Phys. B, vol. **14**, pages 3961–3976, (1981).

[Feuerstein 03a] B. Feuerstein, and U. Thumm. *Fragmentation of H_2^+ in strong 800-nm laser pulses: initial-vibrational-state dependence.* Phys. Rev. A, vol. **67**, page 043405, (2003).

[Feuerstein 03b] B. Feuerstein, and U. Thumm. *Mapping of coherent and decohering nuclear wave-packet dynamics in D_2^+ with ultrashort laser pulses.* Phys. Rev. A, vol. **67**, page 063408, (2003).

[Feuerstein 03c] Bernold Feuerstein, and Uwe Thumm. *On the computation of momentum distributions within wavepacket propagation calculations.* J. Phys. B, vol. **36**, pages 707–716, (2003).

[Feuerstein 06] B. Feuerstein. *private communication.* Kansas State University, Manhattan, KS, (2006).

[Feuerstein 07] B. Feuerstein, Th. Ergler, K. Zrost, C. D. Schröter, R. Moshammer, J. Ullrich, Thomas Niederhausen, and Uwe Thumm. *Towards a complete characterization of molecular dynamics in ultra-short laser fields.* to be published, (2007).

[For] *Intel Fortran Compiler, version 9.1.* `http://www.intel.com/cd/software/products/asmo-na/eng/279831.htm`.

[Forest 90] Etienne Forest, and Ronald D. Ruth. *Fourth-order symplectic integration.* Physica D, vol. **43**, pages 105–117, (1990).

[Frasinski 99] L. J. Frasinski, J. H. Posthumus, J. Plumridge, K. Codling, P. F. Taday, and A. J. Langley. *Manipulation of Bond Hardening in H_2^+ by Chirping of Intense Femtosecond Laser Pulses*. Phys. Rev. Lett., vol. **83**, pages 3625–3628, (1999).

[Gauyacq 96] J. P. Gauyacq, A. Borisov, and D. Teillet-Billy. *Destruction of Negative Ions in Heavy Particle-Surface Collisions*, in: Negative Ions, edited by V. Esaulov. Cambridge University Press, Cambridge, (1996).

[Ge 98] Jiu-Yuan Ge, and John Z. H. Zhang. *Use of negative complex potential as absorbing potential*. J. Chem. Phys., vol. **108**, pages 1429–1433, (1998).

[Gealy 87] M. W. Gealy, and B. Van Zyl. *Cross sections for electron capture and loss. I. H^+ and H^- impact on H and H_2*. Phys. Rev. A, vol. **36**, pages 3091–3099, (1987).

[Giusti-Suzor 90] A. Giusti-Suzor, X. He, O. Atabek, and F. H. Mies. *Above-threshold dissociation of H_2^+ in intense laser fields*. Phys. Rev. Lett., vol. **64**, pages 515–518, (1990).

[Goll 06] Erich Goll, Günter Wunner, and Alejandro Saenz. *Formation of Ground-State Vibrational Wave Packets in Intense Ultrashort Laser Pulses*. Phys. Rev. Lett., vol. **97**, page 103003, (2006).

[Grosser 96] J. Grosser, D. Hohmeier, and S. Klose. *Differential cross sections for radiation assisted collisions between atoms*. J. Phys. B, vol. **29**, pages 299–306, (1996).

[Guillemot 99] L. Guillemot, and V. A.Esaulov. *Interaction Time Dependence of Electron Tunneling Processes between an Atom and a Surface*. Phys. Rev. Lett., vol. **82**, pages 4552–4555, (1999).

[Harris 97] Steve Harris. *Electromagnetically induced transparency.* Physics Today, vol. **50**, pages 36–42, (1997).

[Hau 99] Lene Vestergaard Hau, S. E. Harris, Zachary Dutton, and Cyrus H. Behroozi. *Light speed reduction to 17 metres per second in an ultracold atomic gas.* Nature, vol. **397**, pages 594–598, (1999).

[Hecht 00] T. Hecht, H. Winter, A. G. Borisov, J. P. Gauyacq, and A. K. Kazansky. *Role of the 2D Surface State Continuum and Projected Band Gap in Charge Transfer in Front of a Cu(111) Surface.* Phys. Rev. Lett., vol. **84**, pages 2517–2520, (2000).

[Henderson 23] G. H. Henderson. *Changes in the charges of an α-particle passing through matter.* Proc. Roy. Soc. A, vol. **102**, page 496, (1923).

[Hertel 06] I. V. Hertel, and W. Radloff. *Ultrafast dynamics in isolated molecules and molecular clusters.* Rep. Prog. Phys., vol. **69**, pages 1897–2003, (2006).

[Ho 83] Y. K. Ho. *The method of complex coordinate rotation and its applications to atomic collision processes.* Phys. Rep., vol. **99**, pages 1–68, (1983).

[Höhr 05] C. Höhr, A. Dorn, B. Najjari, D. Fischer, C. D. Schröter, and J. Ullrich. *Electron Impact Ionization in the Presence of a Laser Field: A Kinematically Complete ($n\gamma e$,2e) Experiment.* Phys. Rev. Lett., vol. **94**, page 153201, (2005).

[Hussain 00] A. N. Hussain, and G. Roberts. *Procedure for absorbing time-dependent wave functions at low kinetic energies and large bandwidths.* Phys. Rev. A, vol. **63**, page 012703, (2000).

[Iaconis 98] C. Iaconis, and A. Walmsley. *Spectral phase interferometryfor direct electric-field reconstructionof ultrashort optical pulses.* Optics Lett., vol. **23**, pages 792–794, (1998).

[Jaroń 97] A. Jaroń, and J. Z. Kamiński. *Diffraction electron-atom scattering in an intense low-frequency laser field.* Phys. Rev. A, vol. **56**, pages R4393–4396, (1997).

[Joachain 00] C. J. Joachain, M. Dörr, and N. Kylstra. *High intensity laser-atom physics*, pages 225–286, in: Advances in Atomic, Molecular, and OpticalPhysics, edited by Benjamin Bederson, and Herbert Walther, volume 42. Elsevier - Academic Press, (2000).

[Jochim 03] S. Jochim, M. Bartenstein, A. Altmeyer, G. Hendl, S. Riedl, C. Chin, J. Hecker Denschlag, and R. Grimm. *Bose-Einstein Condensation of Molecules.* Science, vol. **302**, pages 2101–2103, (2003).

[Jones 00] David J. Jones, Scott A. Diddams, Jinendra K. Ranka, Andrew Stentz, Robert S. Windeler, John L. Hall, and Steven T. Cundiff. *Carrier-Envelope Phase Control of Femtosecond Mode-Locked Lasers and Direct Optical Frequency Synthesis.* Science, vol. **288**, pages 635–639, (2000).

[Katsuki 06] Hiroyuki Katsuki, Hisashi Chiba, Bertrand Girard, Christoph Meier, and Kenji Ohmori. *Visualizing Picometric Quantum Ripples of Ultrafast Wave-Packet Interference.* Science, vol. **311**, pages 1589–1592, (2006).

[Keller 95] A. Keller. *Asymptotic analysis in time-dependent calculations with divergent coupling.* Phys. Rev. A, vol. **52**, pages 1450–1457, (1995).

[Killian 04] B. J. Killian, R. Cabrera-Trujillo, E. Deumens, and Y. Öhrn. *Resonant charge transfer between H+ and H from 1 to 5000 eV.* J. Phys. B, vol. **37**, pages 4733–4747, (2004).

[Kirchner 99] T. Kirchner, H. J. Lüdde, O. J. Kroneisen, and R. M. Dreizler. *New trends in the description of ion-atom collisions by time-dependent quantum methods*. Nucl. Instr. Meth. B, vol. **154**, pages 46–53, (1999).

[Kirchner 02] Tom Kirchner. *Manipulating Ion-Atom Collisions with Coherent Electromagnetic Radiation*. Phys. Rev. Lett., vol. **89**, page 093203, (2002).

[Kirchner 04] T. Kirchner. *Laser-field-induced modifications of electron-transfer processes in ion-atom collisions*. Phys. Rev. A, vol. **69**, page 063412, (2004).

[Kirchner 07] T. Kirchner. *Laser-field enhanced electron transfer in p-Ne and p-Ar collisions*. Phys. Rev. A, vol. **75**, page 025401, (2007).

[Kołos 64] W. Kołos, and L. Wolniewicz. *Accurate Adiabatic Treatment of the Ground State of the Hydrogen Molecule*. J. Chem. Phys., vol. **41**, pages 3663–3673, (1964).

[Kołos 65] W. Kołos, and L. Wolniewicz. *Potential-Energy Curves for the* $X\ ^1\Sigma_g^+$, $b\ ^3\Sigma_u^+$, *and* $C\ ^1\Pi_u$ *States of the Hydrogen Molecule*. J. Chem. Phys., vol. **43**, pages 2429–2441, (1965).

[Kosloff 86] R. Kosloff, and D. Kosloff. *Absorbing boundaries for wave propagation problems*. J. Comp. Phys., vol. **63**, pages 363–376, (1986).

[Kosloff 96] Ronnie Kosloff. pages 185–230, in: Quantum Molecular Dynamics on Grids, edited by R. E. Wyatt, and J. Z. Zhang. Marcel Dekker, New York, (1996).

[Krause 92] Jeffrey L. Krause, Kenneth J. Schafer, and Kenneth C. Kulander. *Calculation of photoemission from atoms subject to intense laser fields*. Phys. Rev. A, vol. **45**, pages 4998–5010, (1992).

[Kroll 73] Norman N. Kroll, and Kenneth M. Watson. *Charged-particle scattering in the presence of a strong electromagnetic wave.* Phys. Rev. A, vol. **8**, page 804, (1973).

[Kroneisen 99] O. J. Kroneisen, H. J. Ldde, T. Kirchner, and R. M. Dreizler. *The basis generator method: optimized dynamical representation of the solution of time-dependent quantum problems.* J. Phys. A, vol. **32**, pages 2141–2156, (1999).

[Kulander 96] K. C. Kulander, F. H. Mies, and K. J. Schafer. *Model for studies of laser-induced nonlinear processes in molecules.* Phys. Rev. A, vol. **53**, pages 2562–2570, (1996).

[Kürpick 97] P. Kürpick, U. Thumm, and U. Wille. *Resonance formation of hydrogenic levels in front of metal surfaces.* Phys. Rev. A, vol. **56**, pages 543–554, (1997).

[Kürpick 98] P. Kürpick, and U. Thumm. *Hybridization of ionic levels at metal surfaces.* Phys. Rev. A, vol. **58**, pages 2174–2182, (1998).

[Lane 58] A. M. Lane, and R. G. Thomas. *R-Matrix Theory of Nuclear Reactions.* Rev. Mod. Phys., vol. **30**, pages 257–353, (1958).

[LAP] *LAPACK – Linear Algebra PACKage, collectrion of FORTRAN 77 routines for solving linear equation systems and eigenvalue problems.* `http://www.netlib.org/lapack/`.

[Leanhardt 03] A. E. Leanhardt, T. A. Pasquini, M. Saba, A. Schirotzek, Y. Shin, D. Kielpinski, D. E. Pritchard, and W. Ketterle. *Cooling Bose-Einstein Condensates Below 500 Picokelvin.* Science, vol. **301**, pages 1513–1515, (2003).

[Légaré 05] F. Légaré, Kevin F. Lee, I. V. Litvinyuk, P. W. Dooley, A. D. Bandrauk, D. M. Villeneuve, and P. B. Corkum. *Imaging the time-dependent structure of a molecule as it undergoes dynamics*. Phys. Rev. A, vol. **72**, page 052717, (2005).

[Lehoucq 01] R. B. Lehoucq. *Implicitly Restarted Arnoldi Methods and Subspace Iteration*. SIAM J. Matrix Anal. A, vol. **23**, pages 551–562, (2001).

[Lehtovaara 07] L. Lehtovaara, J. Toivanen, and J. Eloranta. *Solution of time-independent Schrödinger equation by the imaginary time propagation method*. J. Comp. Phys., vol. **221**, pages 148–158, (2007).

[Lein 03] Manfred Lein, and Jan M.Rost. *Ultrahigh Harmonics from Laser-Assisted Ion-Atom Collisions*. Phys. Rev. Lett., vol. **91**, page 243901, (2003).

[Li 97] S.-M. Li, Y.-G. Miao, Z.-F. Zhou, J. Chen, and Y.-Y. Kiu. *Laser-assisted electron capture by a fast proton from a hydrogen atom*. Zeitschrift für Physik D, vol. **39**, pages 29–34, (1997).

[Li 02] S.-M. Li, J. Chen, and Z.-F. Zhou. *Ionization of atomic hydrogen by protons in the presence of a laser field*. J. Phys. B, vol. **35**, pages 557–566, (2002).

[Lin 06] C. D. Lin, X. M. Tong, and Toru Morishita. *Direct experimental visualization of atomic and electron dynamics with attosecond pulses*. J. Phys. B, vol. **39**, pages S419–426, (2006).

[Litvinyuk 03] I. V. Litvinyuk, Kevin F. Lee, P. W. Dooley, D. M. Rayner, D. M. Villeneuve, and P. B. Corkum. *Alignment-Dependent Strong Field Ionization of Molecules*. Phys. Rev. Lett., vol. **90**, page 233003, (2003).

[Liu 01] Chien Liu, Zachary Dutton, Cyrus H. Behroozi, and Lene Vestergaard Hau. *Observation of coherent optical information storage in an atomic medium using halted light pulses.* Nature, vol. **409**, pages 490–493, (2001).

[Los 90] J. Los, and J. J. C. Geerlings. *Charge exchange in atom-surface collisions.* Phys. Rep., vol. **190**, pages 133–190, (1990).

[Maazouz 97] M. Maazouz, A. G. Borisov, V. A. Esaulov, J. P. Gauyacq, L. Guillemot, S. Lacombe, and D. Teillet-Billy. *Effect of metal band characteristics on resonant electron capture: H^- formation in the scattering of hydrogen ions on Mg, Al, and Ag surfaces.* Phys. Rev. B, vol. **55**, pages 13869–13877, (1997).

[Madsen 02] L. B. Madsen, J. P. Hansen, and L. Kocbach. *Excitation inn ion-atom collisions inside subfemtosecond laser pulses.* Phys. Rev. Lett., vol. **89**, page 093202, (2002).

[Maharjan 06] C. M. Maharjan, A. S. Alnaser, I. V. Litvinyuk, P. Ranitovic, and C. L. Cocke. *Wavelength dependence of momentum-space images of low-energy electrons generated by short intense laser pulses at high intensities.* J. Phys. B, vol. **39**, pages 1955–1964, (2006).

[Maiman 60] T. H. Maiman. *Stimulated Optical Radiation in Ruby.* Nature, vol. **187**, pages 493–494, (1960).

[Mason 93] H. J. Mason. *Laser-assisted electron-atom collisions.* Rep. Prog. Phys., vol. **56**, pages 1275–1346, (1993).

[McKenna 07] J. McKenna, C. R. Calvert, W. A. Bryan, E. M. L. English, J. Wood, D. S. Murphy, I. C. E. Turcu, J. M. Smith, K. G. Ertel, O. Chekhlov, E. J. Divall, J. F. McCann, W. R. Newell, and I. D. Williams. *Imaging quantum vibrations on an ultrashort timescale: the deuterium molecular ion.* J. Phys.: Conf. Ser., vol. **58**, pages 375–378, (2007).

[Minami 06] T. Minami, M. S. Pindzola, T.-G. Lee, and D. R. Schultz. *Lattice, time-dependent Schrödinger equation approach for charge transfer in collisions of Be^{4+} with atomic hydrogen.* J. Phys. B, vol. **39**, pages 2877–2891, (2006).

[Mitchell 80] Andrew Ronald Mitchell, and David F. Griffiths. *The finite difference method in partial differential equations.* John Wiley & Sons Ltd., New York, (1980).

[Moon 06] E. Moon, Chengquan Li, Zuoliang Duan, J. Tackett, K. L. Corwin, B. R. Washburn, and Zenghu Chang. *Reduction of fast carrier-envelope phase jitter in femtosecond laser amplifiers.* Optics Express, vol. **14**, pages 9758–9763, (2006).

[Moshammer 02] R. Moshammer, B. Feuerstein, J. Crespo Lopez-Urrutia, J. Deipenwisch, A. Dorn, D. Fischer, C. Höhr, P. Neumayer, C. D. Schröter, J. Ullrich, H. Rottke, C. Trump, M. Wittmann, G. Korn, and W. Sandner. *Correlated two-electron dynamics in strong-field double ionization.* Phys. Rev. A, vol. **65**, page 035401, (2002).

[Muga 04] J. G. Muga, J. P. Palao, B. Navarro, and I. L. Egusquiza. *Complex absorbing potentials.* Phys. Rep., vol. **395**, pages 357–426, (2004).

[Niederhausen 04a] Thomas Niederhausen. *Zirkularer Dichroismus in Laser-assistierten Proton – Wasserstoff Stoßprozessen.* Master's thesis, Universität Kassel, Germany, (2004).

[Niederhausen 04b] Thomas Niederhausen, Bernold Feuerstein, and Uwe Thumm. *Circular dichroism in laser-assisted proton – hydrogen collisions.* Phys. Rev. A, vol. **70**, page 023408, (2004).

[Niederhausen 06] Thomas Niederhausen, and Uwe Thumm. *Capture and ionization in laser-assisted proton – hydrogen collisions.* Phys. Rev. A, vol. **73**, page 041404, (2006).

[Niederhausen 07] Thomas Niederhausen, and Uwe Thumm. *Controlled vibrational quenching of nuclear wave packets in* D_2^+. submitted to Phys. Rev. A, (2007).

[Niikura 02] Hiromichi Niikura, F. Légaré, R. Hasbani, A. D. Bandrauk, Misha Yu Ivanov, D. M. Villeneuve, and P. B. Corkum. *Sub-laser-cycle electron pulses for probing molecular dynamics.* Nature, vol. **417**, pages 917–922, (2002).

[Niikura 03] Hiromichi Niikura, F. Légaré, R. Hasbani, Misha Yu Ivanov, D. M. Villeneuve, and P. B. Corkum. *Probing molecular dynamics with attosecond resolution using correlated wave packet pairs.* Nature, vol. **421**, pages 826–829, (2003).

[Niikura 04] Hiromichi Niikura, D. M. Villeneuve, and P. B. Corkum. *Stopping a Vibrational Wave Packet with Laser-Induced Dipole Forces.* Phys. Rev. Lett., vol. **92**, page 133002, (2004).

[Niikura 06] Hiromichi Niikura, D. M. Villeneuve, and P. B. Corkum. *Controlling vibrational wave packets with intense, few-cycle laser pulses*. Phys. Rev. A, vol. **73**, page 021402, (2006).

[Nolting 97] Wolfgang Nolting. *Quantenmechanik – Teil 1*, volume 5 of *Grundkurs Thoretische Physik*. Vieweg, 4th edition, (1997).

[Obreshkov 06] Boyan Obreshkov, and Uwe Thumm. *Neutralization of H^- near vicinal metal surfaces*. Phys. Rev. A, vol. **74**, page 012901, (2006).

[Obreshkov 07] Boyan Obreshkov, and Uwe Thumm. *"Step-up" versus "step-down" scattering asymmetry in the neutralization of H^- on free-electron vicinal metal surfaces*. Surf. Sci., vol. **601**, pages 622–631, (2007).

[Ope] *OpenMP – Open Multi-Processing application programming interface for shared-memory multiprocessing*.
`http://www.openmp.org/`.

[Palaniyappan 05] S. Palaniyappan, A. DiChiara, E. Chowdhury, A. Falkowski, G. Ongadi, E. L. Huskins, and B. C. Walker. *Ultrastrong Field Ionization of Ne^{n+} ($n \leq 8$): Rescattering and the Role of the Magnetic Field*. Phys. Rev. Lett., vol. **94**, page 243003, (2005).

[Park 86] Tae Jun Park, and J. C. Light. *Unitary quantum time evolution by iterative Lanczos reduction*. J. Chem. Phys., vol. **85**, pages 5870–5876, (1986).

[Paulus 03] G. G. Paulus, F. Lindner, H. Walther, A. Baltuška, E. Goulielmakis, M. Lezius, and F. Krausz. *Measurement of the Phase of Few-Cycle Laser Pulses*. Phys. Rev. Lett., vol. **91**, page 253004, (2003).

[PGP] *PGPlot – Graphics Subroutine Library for C and Fortran.* http://www.astro.caltech.edu/~tjp/pgplot/.

[Pindzola 03] M. S. Pindzola, T. Minami, and D. R. Schultz. *Laser-modified charge-transfer processes in proton collisions with lithium atoms.* Phys. Rev. A, vol. **68**, page 013404, (2003).

[Pindzola 07] M. S. Pindzola, F. Robicheaux, S. D. Loch, J. C. Berengut, T. Topcu, J. Colgan, M. Foster, D. C. Griffin, C. P. Ballance, D. R. Schultz, Minami T, N. R. Badnell, M. C. Witthoeft, D. R. Plante, D. M. Mitnik, J. A. Ludlow, and U. Kleiman. *The time-dependent close-coupling method for atomic and molecular collision processes.* J. Phys. B, vol. **40**, pages R39–60, (2007).

[PLP] *PLPlot – A Scientific Plotting Library.* http://plplot.sourceforge.net/.

[Poirier 03] B. Poirier, and T. Carrington. *Semiclassical optimized complex absorbing potentials of polynomial form: II complex case.* J. Chem. Phys., vol. **119**, pages 77–89, (2003).

[Posthumus 04] J. H. Posthumus. *The dynamics of small molecules in intense laser fields.* Rep. Prog. Phys., vol. **67**, pages 623–665, (2004).

[Press 92] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in FORTRAN 77.* Cambridge University Press, Cambridge, England, (1992).

[Protopapas 97] M. Protopapas, C. H. Keitel, and P. L. Knight. *Atomic physics with super-high intensity laser.* Rep. Prog. Phys., vol. **60**, pages 389–486, (1997).

[Reinsch 00] Matthias W. Reinsch. *A simple expression for the terms in the Baker-Campbell-Hausdorff series.* J. Math. Phys., vol. **41**, pages 2434–2442, (2000).

[Robinett 04] R. W. Robinett. *Quantum wave packet revivals.* Phys. Rep., vol. **392**, pages 1–119, (2004).

[Roudnev 04] Vladimir Roudnev, B. D. Esry, and I. Ben-Itzhak. *Controlling HD^+ and H_2^+ Dissociation with the Carrier-Envelope Phase Difference of an Intense Ultrashort Laser Pulse.* Phys. Rev. Lett., vol. **93**, page 163601, (2004).

[Rudenko 06] A. Rudenko, Th. Ergler, B. Feuerstein, K. Zrost, C. D. Schröter, R. Moshammer, and J. Ullrich. *Real-time observation of vibrational revival in the fastest molecular system.* Chem. Phys., vol. **329**, pages 193–202, (2006).

[Russell 03] Philip Russell. *Photonic Crystal Fibers.* Science, vol. **299**, pages 358–362, (2003).

[Saenz 00] Alejandro Saenz. *On the influence of vibrational motion on strong-field ionization rates in molecules.* J. Phys. B, vol. **33**, pages 4365–4372, (2000).

[Sanchez 99] E. Sanchez, L. Guillemot, and V. A. Esaulov. *Electron Transfer in the Interaction of Fluorine and Hydrogen with $Pd(100)$: The Case of a Transition Metal versus Jellium.* Phys. Rev. Lett., vol. **83**, pages 428–431, (1999).

[Sansone 06] G. Sansone, E. Benedetti, F. Calegari, C. Vozzi, L. Avaldi, R. Flammini, L. Poletto, P. Villoresi, C. Altucci, R. Velotta, S. Stagira, S. De Silvestri, and M. Nisoli. *Isolated Single-Cycle Attosecond Pulses*. Science, vol. **314**, pages 443–446, (2006).

[Schultz 03] D. R. Schultz, and P. S. Krstić. *Ionization of helium by antiprotons: Fully correlated, four-dimensional lattice approach*. Phys. Rev. A, vol. **67**, page 022712, (2003).

[Scrinzi 06] A. Scrinzi, M. Yu. Ivanov, R. Kienberger, and D. M. Villeneuve. *Attosecond physics*. J. Phys. B, vol. **39**, pages R1–37, (2006).

[Seideman 95] Tamar Seideman, M. Yu. Ivanov, and P. B. Corkum. *Role of Electron Localization in Intense-Field Molecular Ionization*. Phys. Rev. Lett., vol. **75**, pages 2819–2822, (1995).

[Seife 03] Charles Seife. *Next Linear Collider: Why Physicists Long for the Straight and Narrow*. Science, vol. **299**, pages 1171–1172, (2003).

[Shan 05] Bing Shan, Shambhu Ghimire, and Zenghu Chang. *Generation of the attosecond extreme ultraviolet supercontinuum by a polarization gating*. J. Mod. Opt., vol. **52**, pages 277–283, (2005).

[Shao 94] H. Shao, D. C. Langreth, and P. Nordlander. *Theoretical description of charge transfer in atom-surface scattering*, pages 117–186, in: Low Energy Ion-Surface Interactions, edited by J. Wayne Rabalais. Wiley, New York, (1994).

[Shverdin 05] M. Y. Shverdin, D. R. Walker, D. D. Yavuz, G. Y. Yin, and S. E. Harris. *Generation of a Single-Cycle Optical Pulse*. Phys. Rev. Lett., vol. **94**, page 033904, (2005).

[Sidky 00] E. Y. Sidky, and B. D. Esry. *Boundary-Free Propagation with the Time-Dependent Schrödinger Equation*. Phys. Rev. Lett., vol. **85**, pages 5086–5089, (2000).

[Stapelfeldt 98] Henrik Stapelfeldt, Eric Constant, Hirofumi Sakai, and Paul B. Corkum. *Time-resolved Coulomb explosion imaging: A method to measure structure and dynamics of molecular nuclear wave packets*. Phys. Rev. A, vol. **58**, pages 426–433, (1998).

[Staudte 07] A. Staudte, D. Paviči, S. Chelkowski, D. Zeidler, M. Meckel, H. Niikura, M. Schöffler, S. Schössler, B. Ulrich, P. P. Rajeev, Th. Weber, T. Jahnke, D. M. Villeneuve, A. D. Bandrauk, C. L. Cocke, P. B. Corkum, and R. Dörner. *Attosecond Strobing of Two-Surface Population Dynamics in Dissociating* H_2^+. Phys. Rev. Lett., vol. **98**, page 073003, (2007).

[Stolow 04] A. Stolow, A. E. Bragg, and D. M. Neumark. *Femtosecond Time-Resolved Photoelectron Spectroscopy*. Chem. Rev., vol. **104**, pages 1719–1758, (2004).

[Su 91] Q. Su, and J. H. Eberly. *Model atom for multiphoton physics*. Phys. Rev. A, vol. **44**, pages 5997–6008, (1991).

[Sudiarta 07] I. Wayan Sudiarta, and D. J. Wallace Geldart. *Solving the Schrödinger equation using the finite difference time domain method*. J. Phys. A, vol. **40**, pages 1885–1896, (2007).

[Suzuki 97] Masuo Suzuki. *Compact exponential product formulas and operator functional derivative*. J. Math. Phys., vol. **38**, pages 1183–1196, (1997).

[Tal-Ezer 84] H. Tal-Ezer, and R. Kosloff. *An accurate and efficient scheme for propagating the time dependent Schrödinger equation.* J. Chem. Phys., vol. **81**, pages 3967–3971, (1984).

[Thumm 91] U. Thumm, and D. W. Norcross. *Evidence for very narrow shape resonances in low-energy electron-Cs scattering.* Phys. Rev. Lett., vol. **67**, pages 3495–3498, (1991).

[Thumm 00] U. Thumm, P. Kürpick, and U. Wille. *Size quantization effects in atomic level broadening near thin metallic films.* Phys. Rev. B, vol. **61**, pages 3067–3077, (2000).

[Thumm 02] Uwe Thumm. *Ion-surface interactions*, pages 592–603, in: Book of Invited Papers, XXII. ICPEAC, Santa Fe, NM, edited by J. Burgdorfer, J. S. Cohen, S. Datz, and C. R. Vane. Rinton Press, Princeton, NJ, (2002).

[Tong 02] X. M. Tong, Z. X. Zhao, and C. D. Lin. *Theory of molecular tunneling ionization.* Phys. Rev. A, vol. **66**, page 033402, (2002).

[Tong 06] X. M. Tong, and C. D. Lin. *Attosecond XUV pulses for complete mapping of the time-dependent wave packets of D_2^+.* Phys. Rev. A, vol. **73**, page 042716, (2006).

[Tong 07] X. M. Tong, and C. D. Lin. *Carrier-envelope phase dependence of non-sequential double ionization of H_2 by few-cycle laser pulses.* J. Phys. B, vol. **40**, pages 641–649, (2007).

[Udem 02] Th. Udem, R. Holzwarth, and T. W. Hänsch. *Optical frequency metrology.* Nature, vol. **416**, pages 233–237, (2002).

[Ullrich 97] J. Ullrich, R. Moshammer, R. Dörner, O. Jagutzki, V. Mergel, H. Schmidt-Böcking, and L. Spielberger. *Recoil-ion momentum spectroscopy.* J. Phys. B, vol. **33**, pages 2917–2974, (1997).

[Ullrich 03] J. Ullrich, R. Moshammer, A. Dorn, R. Dörner, L. P. H. Schmidt, and H. Schmidt-Böcking. *Recoil-ion and electron momentum spectroscoy: reaction-microscopes.* Rep. Prog. Phys., vol. **66**, pages 1463–1545, (2003).

[Urbain 04] X. Urbain, B. Fabre, E. M. Staicu-Casagrande, N. de Ruette, V. M. Andrianarijaona, J. Jureta, J. H. Posthumus, A. Saenz, E. Baldit, and C. Cornaggia. *Intense-Laser-Field Ionization of Molecular Hydrogen in the Tunneling Regime and Its Effect on the Vibrational Excitation of* H_2^+. Phys. Rev. Lett., vol. **92**, page 163004, (2004).

[Voitkiv 01a] A. B. Voitkiv, and J. Ullrich. *Binary-encounter electron emission in fast atomic collisions in the presence of coherent electromagnetic radiation.* J. Phys. B, vol. **34**, pages 1673–1683, (2001).

[Voitkiv 01b] A. B. Voitkiv, and J. Ullrich. *Modification of energy and angular spectra of binary-encounter emission in fast ion-atom collisions assisted by a low-frequency electromagnetic field.* J. Phys. B, vol. **34**, pages 4383–4400, (2001).

[Voitkiv 06] A. B. Voitkiv, N. Toshima, and J. Ullrich. *Ionization of hydrogen by ion impact in the presence of a laser field resonant to boundbound atomic transitions.* J. Phys. B, vol. **39**, pages 3791–3803, (2006).

[Wallbank 94] B. Wallbank, and J. K. Holmes. *Differential cross sections for laser-assisted elastic electron scattering from argon.* J. Phys. B, vol. **27**, pages 5405–5418, (1994).

[Wang 03] J. G. Wang, P. C. Stancil, A. R. Turner, and D. L. Cooper. *Charge transfer of O^{3+} ions with atomic hydrogen*. Phys. Rev. A, vol. **67**, page 012710, (2003).

[Weber 05] *Dynamics of Atoms and Molecules*, pages 105–200, in: Progress Report 2003 – 2004, edited by Thomas Weber, Gernot Vogt, and Bernd Martin. Max-Planck-Institut für Kernphysik, Heidelberg, Germany, (2005).

[Wethekam 03] S. Wethekam, A. Mertens, and H. Winter. *Survival of He^+ Ions during Grazing Scattering from a Ag(111) Surface*. Phys. Rev. Lett., vol. **90**, page 037602, (2003).

[Wigner 46] Eugene P. Wigner. *Resonance Reactions*. Phys. Rev., vol. **70**, pages 606–618, (1946).

[Wolniewicz 93] L. Wolniewicz. *Relativistic energies of the ground state of the hydrogen molecule*. J. Chem. Phys., vol. **99**, pages 1851–1868, (1993).

[Yang 02] Y. Yang, and J. A. Yarmoff. *Charge Exchange in Li Scattering from Si Surfaces*. Phys. Rev. Lett., vol. **89**, page 196102, (2002).

[Yoshida 90] Haruo Yoshida. *Construction of higher order symplectic integrators*. Phys. Lett. A, vol. **150**, pages 262–268, (1990).

[Yoshida 99] S. Yoshida, S. Watanabe, C. O. Reinhold, and J. Burgdörfer. *Reflection-free propagation of wave packets*. Phys. Rev. A, vol. **60**, pages 1113–1123, (1999).

[Zewail 00] Ahmed H. Zewail. *Femtochemistry: Atomic-Scale Dynamics of the Chemical Bond*. J. Phys. Chem. A, vol. **104**, pages 5660–5694, (2000).

[Zhao 02] Z. X. Zhao, B. D. Esry, and C. D. Lin. *Boundary-free scaling calculation of the time-dependent Schrödinger equation for laser-atom interactions.* Phys. Rev. A, vol. **65**, page 023402, (2002).

[Zou 02] Shiyang Zou, Lukáš Pichl, Mineo Kimura, and Takako Kato. *Total and differential cross-section calculations for proton-impact ionization of hydrogen at low energies.* Phys. Rev. A, vol. **66**, page 042707, (2002).

[Zuo 95] T. Zuo, and A. D. Bandrauk. *Charge-Resonant-Enhanced Ionization of Diatomic Molecular Ions by Intense Lasers.* Phys. Rev. A, vol. **52**, pages R2511–R2514, (1995).