TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DIPLOMARBEIT

Improvements of the Numerical Implementation of Matrix Elements Used to Calculate Fusion Relevant Ion-Atom Collisions

Ausgeführt am Institut für
Angewandte Physik
der Technischen Universität Wien

unter der Anleitung von
Univ.-Prof. Dr. Friedrich Aumayr
Dr. Katharina Igenbergs

durch
Alexander Veiter
Borschkegasse 16/12
1090 Wien

10.09.2013 Alexander Veiter

Abstract

In this thesis, the computational implementation of the well-known Atomic-Orbital Close-Coupling method was improved so that the calculation of partial cross sections for charge exchange reactions involving heavy, highly charged ions with relevance to fusion plasma diagnostics is feasible.

Accurate partial cross sections for charge exchange reactions are needed for the analysis of data measured by Charge Exchange Spectroscopy, an important tool within the field of fusion plasma diagnostics. This method makes use of the visible light emitted during electronic transitions within plasma particles. These electronic transitions follow a charge exchange reaction between highly charged plasma ions and particles from a neutral beam. Ions in a fusion plasma may originate either from an intrinsic or an artificial source. The latter come in play when using the Radiative Plasma Edge Cooling technique with which the peak power loads on the plasma facing components can be mitigated through inelastic collisions of plasma particles with a selected impurity gas. Currently, mostly Nitrogen and Neon are used for this purpose while the next fusion experiment ITER is planned to use Argon.

Describing large ions such as Argon within the framework of the Atomic-Orbital Close-Coupling method requires the inclusion of a large number of atomic basis states (of order 10^3). Such large basis sets not only pose computational difficulties like numerical errors during the evaluation of matrix elements, they also put great demand on the used hardware. In this thesis, the current limits of the computational implementation were tested when using large collisional systems, e.g. $Ar^{18+} + H$, and suggestions to extend those limits were proposed.

In order to find potential sources of numerical errors, a detailed and systematic study of the matrix elements which describe the interaction between the two colliding particles was conducted. As a result, numerical instabilities and inaccuracies could be significantly reduced by implementing more accurate basic mathematical functions and using quadruple precision numbers for the complex coefficients involved in the calculation.

The code described in this thesis is a parallel program designated to be executed on cluster computers like the Vienna Scientific Cluster (VSC-2). Using a large basis set entails the production of enormous amounts of data which need to be available during the entire calculation. However, each computing node on a cluster has a certain fixed memory limit. In this thesis, guidelines were developed on how to efficiently distribute the data over all available computing nodes. Furthermore, suggestions on how to reduce idle times to ensure a more efficient use of the computational resources are presented.

Finally, a detailed quantitative and qualitative analysis of the collisional system $B^{5+} + H(1s)$ is given. This may serve as a benchmark system for further developments of the code.

Kurzfassung

Das Ziel dieser Arbeit war es, die numerische Implementierung der Atomic-Orbital Close-Coupling Theorie zur Berechnung partieller Wirkungsquerschnitte von Ladungsaustauschprozessen zu verbessern, damit auch Stoßsysteme mit schweren, hoch geladenen Ionen berechnet werden können.

Solche Wirkungsquerschnitte sind für die Auswertung von Messdaten der Ladungsaustauschspektroskopie ("charge exchange spectroscopy") von großer Bedeutung. Diese wichtige Methode der Plasmadiagnostik nützt jenes sichtbare Licht, welches von angeregten Ionen im Plasma nach einem Ladungsaustausch mit den Atomen eines neutralen Diagnostik- oder Heizstrahls emittiert wird. Ionen in einem Plasma können entweder eine intrinsische Quelle haben oder künstlich eingebracht werden. Die Methode der strahlenden Randschichtkühlung ("radiative plasma edge cooling") versucht die Plasmarandschicht durch inelastische Stöße der Plasmateilchen mit einem künstlich eingebrachten Verunreinigungsgas zu kühlen, bevor diese auf die Wand des Reaktors trifft. Für ITER wurde Argon als Kühlgas vorgeschlagen, daher sind Daten zu Stößen zwischen Argon-Ionen und Wasserstoffatomen von besonderem Interesse.

Will man Argon im Rahmen der Atomic-Orbital Close-Coupling Methode beschreiben, so muss man eine beachtliche Anzahl an Basiszuständen (in der Größenordnung von 10^3) berücksichtigen. Eine so große Basis birgt nicht nur eine größere Gefahr für numerische Fehler während der Auswertung von Matrixelementen, sondern erhöht auch den Aufwand der Berechnung und die Anforderungen an die Hardware enorm. In dieser Arbeit wurden daher aktuelle Grenzen der numerischen Implementierung bei der Berechnung größerer Stoßsysteme, wie z.B. $Ar^{18+} + H$, untersucht und es wurden Vorschläge erarbeitet, wie man diese Grenzen noch ausweiten kann.

Um potentielle Fehlerquellen in der numerischen Genauigkeit zu finden, wurde zunächst die Berechnung der Matrixelemente welche die Wechselwirkungen zwischen den Stoßpartnern beschreiben untersucht. Durch die Implementierung von genaueren mathematischen Funktionen und das Einführen von quad-precision Zahlen für die komplexen Koeffizienten in der Berechnung, ist es gelungen numerische Instabilitäten und Ungenauigkeiten stark zu reduzieren.

Der in dieser Arbeit besprochene Code wurde konzipiert um parallel auf großen Cluster-Rechnern, wie etwa dem Vienna Scientific Cluster (VSC-2), zu laufen. Hat man eine große Basis für die Berechnung, so generiert man riesige Mengen an Daten, die noch dazu während der gesamten Rechnung auf den Nodes verfügbar sein müssen. Nun hat aber der Arbeitsspeicher auf jedem Node nur eine begrenzte Größe. Es wurden daher Leitlinien entwickelt, wie man die verfügbare Größe des Arbeitsspeichers möglichst effizient ausnützt. Zeitgleich werden auch Vorschläge gemacht, wie man Leerläufe im Programm noch weiter reduzieren kann, um die vorhandenen Rechenressourcen möglichst optimal zu nutzen.

Am Ende der Arbeit wird noch eine ausführliche quantitative und qualitative Analyse des Stoßsystems $B^{5+} + H(1s)$ durchgeführt. Dieses Stoßsystem kann bei zukünftigen Weiterentwicklungen des Codes als Benchmark-System herangezogen werden.

Contents

Chapter 1

Introduction

One of the most challenging problems for current and future generations will be to secure a sufficient supply of energy to meet the demands of an ever growing world population. While several renewable energy sources such as wind or solar energy are constantly improved and have proven to be a valuable part of a future energy mix, they still seem insufficient to fully substitute fossil fuels on a larger scale. Nuclear fission could present a stable backbone of world energy production, but due to the accidents in Chernobyl and more recently Fukushima in 2011 it is becoming increasingly politically unwanted. In this context, nuclear fusion has reclaimed some attention as a possible future source of energy. However, it has proven to be very hard to keep the fusion plasma stable and to achieve the extreme temperature, pressure and charge density needed in order for a controlled nuclear fusion to take place (see [1]).

There are multiple fusion experiments currently in operation, most notably the experiments *Joint European Torus (JET)*, which is operational in the UK since 1983 and ASDEX-Upgrade running since 1991 near Munich in Germany. Both JET and ASDEX-Upgrade are so-called *tokamak* reactors. The *stellarator* experiment *Wendelstein 7-X* aims at further developing and testing an alternative concept to the more well-known tokamak reactors and will be finished in 2015.

Currently, most of the attention and financial effort is focussed on the large project ITER, a tokamak being built in Cadarache in southern France. ITER has a total budget of roughly 15 billion euros (see [2], retrieved in Aug. 2013) financed by the European Union (EU), India, Japan, China, Russia, South Korea and the United States and will be by far the largest fusion experiment in the world. The construction of ITER is scheduled to be finished by 2022.

A lot of scientific research has been devoted to the study of the formation of a fusion plasma, its confinement and also the behavior of impurities inside the plasma. Such impurities can either be intrinsic (see [3]) or artificially induced (see [4]).

Intrinsic impurities mainly originate from sputtering of the plasma facing components, which are basically the inner wall of the reactor and (only for a reactor using the tokamak principle) a divertor plate where most of the heat load from the plasma is deposited. The most common elements for these plasma facing components are Carbon, Beryllium and Tungsten (see [3]). The wall and especially the divertor have to withstand an enormous heat load and particle flux from the core plasma. If the peak power load deposited on the wall is too high, then erosion might become too significant and the intrinsic impurities could extinguish the burning plasma (see [3]).

In order to reduce the burden on the plasma facing components and also to improve confine-

ment of the plasma, a technique called *plasma edge radiative cooling* (see [4]) using artificially induced impurities was developed. In this method, an impurity gas (mainly Nitrogen, Neon or Argon) is puffed into the plasma edge. The impurity ions form a highly radiative layer around the plasma edge, absorb heat by inelastic collisions with plasma particles and then distribute the power load more uniformly over the wall (see [4]). While it is sufficient to use Nitrogen as an impurity gas in ASDEX-Upgrade, ITER will have to use Neon or even Argon for plasma edge radiative cooling to be able to sufficiently reduce peak power loads. Since Argon has more electrons which can be ionized, it forms a thicker layer around the plasma edge resulting in better cooling properties in comparison to Nitrogen or Neon.

The diagnostic method *charge exchange spectroscopy (CXS)* (see [5]) makes use of electronic transitions following a *charge exchange (CX)* reaction between impurity ions and particles from a neutral heating or diagnostic beam in order to diagnose a plasma. Diagnosing usually includes, among other things, determining the density of the induced impurity ions, estimating the velocity of plasma rotation and calculating the effective charge of the plasma as well as the ion temperatures (see [5]). To be able to do all these calculations for the artificially induced Argon gas at ITER, it is necessary to have accurate underlying atomic data for the partial cross sections of a CX reaction (sometimes also referred to as *CX excitation*) like

$$Ar^{18+} + H \rightarrow Ar^{17+}(n,l)^* + H^+ \ ,$$

with Ar^{17+} being in an excited state (denoted by *) with quantum numbers (n, l).

After the CX reaction, the excited hydrogen-like Ar^{17+} ion will radiate light while its electron de-excites into lower energy levels. Since spectroscopic measurements using CCD cameras are most sensitive in the visual range of the spectrum, it is advantageous to use transitions which radiate light within this energy range. In the case of Ar^{17+}, these are transitions between higher n-shells, around $n = 14$ to $n = 18$.

This thesis discusses how partial cross sections for CX reactions including such high quantum numbers n can be calculated. To this end, chapter 2 first briefly outlines the theoretical description of ion-atom collisions within the well established *atomic-orbital close-coupling (AOCC) theory* before explaining the computational implementation of the theory into the parallel FORTRAN code "sic3ma". Chapter 3 first lists some of the problematic issues when calculating larger collisional systems in section 3.1. The main aspects under consideration in this chapter are ways to improve the numerical accuracy of the calculations (sections 3.2 and 3.3) as well as the computational performance of the code (section 3.4). Chapter 4 finally gives some detailed results for the much smaller collisional system $B^{5+} + H(1s)$. They can serve as benchmark results for any future changes to the code sic3ma, when trying to achieve the calculation of larger collisional systems. The results and observations of this thesis lay the foundation and outline the path to be followed towards implementing the calculation of larger collisional systems such as $Ar^{18+} + H$.

Atomic units ($\hbar = 1$, $e = 1$, $m_e = 1$) are used throughout this entire thesis unless stated otherwise. The computational results presented have been achieved using the Vienna Scientific Cluster (VSC-2).

In the course of this work, the author has been involved in the following publications in refereed scientific journals and presentations at international conferences:

A. Veiter, K. Igenbergs, J. Schweinzer, M. Wallerberger, F. Aumayr:
"Cross Sections for Charge Exchange in Ar^{18+}- H(n = 1,2) collisions";
currently in preparation

K. Igenbergs, J. Schweinzer, **A. Veiter**, L. Perneczky, E. Frühwirth, M. Wallerberger, R.E. Olson, F. Aumayr:
"Charge exchange and ionisation in N^{7+}-, N^{6+}-, C^{6+}- H(n = 1,2) collisions studied systematically by theoretical approaches";
Journal of Physics B: Atomic, Molecular and Optical Physics, **45** (2012), 065203 (21 pages)

F. Aumayr, **A. Veiter**, K. Igenbergs, M. Wallerberger:
"Atomic-orbital close-coupling calculations for collisions of highly charged ions with atomic hydrogen";
Poster: 11th European Conference on Atoms, Molecules and Photons 2013 (ECAMP-11), Aarhus University/Denmark; 24.06.2013.

K. Igenbergs, R. Olsen, J. Schweinzer, **A. Veiter**, M. Wallerberger, L. Perneczky, F. Aumayr:
"Cross sections for neutral hydrogen beam diagnostics of hot fusion plasmas";
Poster: 10th European Conference on Atoms Molecules and Photons (ECAMP-10), Salamanca/Spain; 07.07.2010.

Chapter 2

Calculating Ion-Atom Collisions

In this chapter, we want to describe the calculation of cross sections for ion-atom collisions. Section 2.1 will deal with the AOCC theory, while section 2.2 will discuss the computational implementation of the theoretical method in the parallel FORTRAN-code "sic3ma".

2.1 Atomic-Orbital Close-Coupling Theory

We consider a collision of a neutral projectile atom P (composed of an ionic core P^+ and a single electron e^-) with a target ion T^{q+}. To analyze this collisional system the close-coupling description of atomic collisions (for more details refer to [6], [7]) is used.

At the beginning of our considerations, we can introduce a couple of approximations to simplify the given problem:

1. For projectile energies above a few eV/amu, the deBroglie wavelength of the core is considerably small in comparison to the region of interaction between target and projectile. The movement of the core can thus be assumed to go along a classical trajectory, while the electron will still be described quantum mechanically. This gives a semiclassical formulation of the problem. (compare [8], section 3.2)

2. In addition to that, the so-called *impact parameter approximation* will be used, where elastic scattering of the core is neglected. At sufficiently high energies, the classical trajectory of the core can be taken to be a straight line traversed at the constant speed v (see [9], p.132). Introducing the impact parameter $\vec{b}$ as the vector of the closest distance between the two cores along a given trajectory, the internuclear separation $\vec{R}$ at a given velocity $\vec{v}$ can be written as (see [9], (3.167) in section 3.7)

$$\vec{R}(\vec{v}, \vec{b}, t) = \vec{v}t + \vec{b} \; . \tag{2.1}$$

The movement of the electron in the potential $V_p(\vec{r_p})$ of the projectile core, respectively $V_t(\vec{r_t})$ of the target core, is described quantum mechanically by the Schrödinger equation

$$i\frac{\partial}{\partial t}\Psi(\vec{r}, t) = H_e(t)\Psi(\vec{r}, t) \; . \tag{2.2}$$

The independent variables in this equation are the time t and the coordinate $\vec{r}$ of the electron. The electronic Hamiltonian H_e in (2.2) is given by (see [10], (2.1) on p.7)

$$H_e(t) = -\frac{1}{2}\nabla_r^2 + V_p(\vec{r_p}(\vec{r}, t)) + V_t(\vec{r_t}(\vec{r}, t)) \; , \tag{2.3}$$

where the vectors $\vec{r_p}$ and $\vec{r_t}$ refer to the coordinate system of the projectile or respectively the target, whereas $\vec{r}$ refers to a chosen inertial system. Bransden and McDowell explain, that in the impact parameter approximation, the cores are unaccelerated, so that any point on the internuclear line, for example the mid-point, can be employed as origin, since all such points define inertial frames (see [9], p.132). We choose the origin to be situated on the target. The geometry of the collisional system can be seen in fig. 2.1.

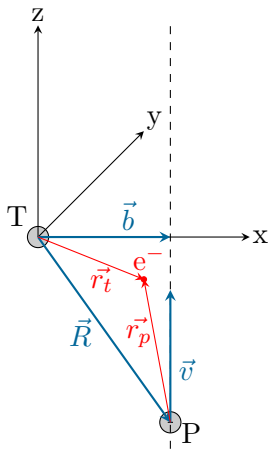


Figure 2.1: Geometry of the collisional system in the impact parameter approximation: the (x,z) collision plane is defined by the projectile velocity $\vec{v}$ and the target. $\vec{b}$ is the impact parameter, $\vec{R}$ the internuclear separation vector. $\vec{r_t}$ and $\vec{r_p}$ are the position vectors of the active electron in the coordinate system of the target center, respectively the projectile center. (graphic from [8])

In the described ion-atom collision, the following reactions are possible (compare [9], p.39):

1. final state of the electron is equal to initial state of the electron:
   $P + T^{q+} \rightarrow P + T^{q+}$

2. excitation:
   $P + T^{q+} \rightarrow P^*(n, \ell) + T^{q+}$

3. charge exchange:
   $P + T^{q+} \rightarrow P^+ + T^{(q-1)+}(n', \ell')$

4. ionization:
   $P + T^{q+} \rightarrow P^+ + T^{q+} + e^-$

In order to account for all these possible reactions, Fritsch and Lin explain that the motion of the electron is constrained to a configuration space which is given by a finite set of basis functions $\psi_k(\vec{r}, t)$ with $k = 1, ..., N$. The time-dependent electronic wavefunction is then approximated by the (truncated) expansion (see [10], p.7)

$$\Psi(\vec{r}, t) = \sum_{k=1}^{N} a_k(t)\psi_k(\vec{r}, t) \ , \tag{2.4}$$

where $a_k(t)$ are time-dependent complex probability amplitudes for the occupation of a certain state ψ_k. The value of the index k can be associated with a state on either the projectile P or target T, depending on how many states are defined on the respective centers. Fritsch and Lin continue to elaborate: Clearly, the decomposition (2.4) is introduced in order to reduce the problem of determining the full electron wavefunction (as function of a continuum of values for coordinate $\vec{r}$) to the problem of determining a finite set of amplitudes $a_k(t)$ (see [10], p.8). The states ψ_k have to sufficiently describe all possible reaction channels listed above. There are a number of different basis sets that can be used, depending on the system, that needs to be described. In the AOCC method, *atomic orbitals* are used to describe initial and final states on both centers. There are also other techniques available. The molecular-orbital close-coupling (MOCC) method, for example, uses *molecular orbitals* (for details refer to [10], section 3.2.1.), while for certain applications *Sturmian basis functions* ([10], section 3.2.3.) or *Gaussian orbitals* ([11], section 2.6) may be more suited. In general, all basis sets have to meet some basic requirements (see [10], p.7):

- The basis set should include the initial state of the electron and all significantly populated final states.
- Intermediate states, which might interact with the initial or final states during the collision, should also be included.
- The basis states should be easy to generate and allow for convenient evaluation of matrix elements, which might be needed later on in (2.18).

Additionally, the movement of projectile and target poses some difficulties that have to be considered: If we take a look at bound states on either target or projectile at infinite separations $R \to \infty$, we can write them as (see [10], (3.1) in section 3.1):

$$\begin{aligned} \hat{\phi}_k^T(\vec{r},t) &= \phi_k^T(\vec{r}_T)\exp\left(i\vec{v}_T\cdot\vec{r} - \frac{i}{2}\int^t dt'\vec{v}_T^2 - i\varepsilon_k^T t\right) \\ \hat{\phi}_k^P(\vec{r},t) &= \phi_k^P(\vec{r}_P)\exp\left(i\vec{v}_P\cdot\vec{r} - \frac{i}{2}\int^t dt'\vec{v}_P^2 - i\varepsilon_k^P t\right) \ , \end{aligned} \tag{2.5}$$

where $\vec{v}_{(T,P)}$ is the velocity of the moving target/projectile and $\varepsilon_k^{(T,P)}$ is the atomic energy of the state k of the respective atom/ion:

$$H_{(T,P)}\phi_k^{(T,P)} = \varepsilon_k^{(T,P)}\phi_k^{(T,P)} \ . \tag{2.6}$$

The states in (2.5) obey the time-dependent Schrödinger equation

$$\left(i\frac{d}{dt} - H\right)\hat{\phi}_k^{T,P}(\vec{r}) = 0 \ . \tag{2.7}$$

The factor $\exp(i\vec{v}_{T,P}\vec{r} - \frac{i}{2}\int dt'\vec{v}_{T,P}^2)$ in expression (2.5) accounts for the fact, that for $R \to \infty$, the electron is attached to either target or projectile and thus carries a linear momentum and a kinetic energy. As Fritsch and Lin point out: Neglect of this factor leads to non-travelling states that are not stationary at infinite separations (see [10], p.10), i.e.

$$\left(i\frac{d}{dt} - H\right)\phi_k^T(\vec{r})exp(-i\varepsilon_k^T t) = -i\vec{v}_T\cdot\nabla\phi_k^T(\vec{r})exp(-i\varepsilon_k^T t) \ . \tag{2.8}$$

We do of course usually assume the target to be at rest, yielding $\vec{v}_T = 0$, which fixes the problem at least for the states on the target but then $\vec{v}_P \neq 0$, since not both centers can rest in a common coordinate system. To account for this motion of the center, which is "dragging" the electron wavefunction with it, Bates and McCarroll (see [6]) already proposed the use of various types of so-called electron translational factors (ETF) $f(\vec{R}, \vec{r})$, yielding basis states consisting of product functions

$$\psi_k(\vec{r}, t) = \chi_k(\vec{r}, t) \cdot f(\vec{R}, \vec{r}) \ , \tag{2.9}$$

where $\chi_k(\vec{r}, t)$ denotes a "normal" non-traveling basis function for the state k. Since we want to be able to give a direct physical interpretation of the basis states in (2.4), the $\psi_k(\vec{r}, t)$ at infinite past or future have to represent bound states on either target or projectile, thus obeying the boundary conditions

$$\psi_k(\vec{r}, t) \xrightarrow{t \to \pm\infty} \hat{\phi}_k^{T,P}(\vec{r}) \ . \tag{2.10}$$

To make this possible, the various ETF have to approach the proper linear momentum at large seperations

$$f(\vec{R}, \vec{r}) \xrightarrow{R \to +\infty} exp(i\vec{v}_{(T,P)}\vec{r}) \ . \tag{2.11}$$

The initial amplitude A_i and the transition amplitude A_f for populating a given final state ϕ_f are then simply given by the coefficients in expansion (2.4),

$$A_i = \langle \hat{\phi}_i(-\infty) | \Psi(-\infty) \rangle = a_i(-\infty), \qquad A_f = \langle \hat{\phi}_f(+\infty) | \Psi(+\infty) \rangle = a_f(+\infty) \ . \tag{2.12}$$

The choice of a suitable basis set is not trivial. As Bransden and McDowell put it: In applications of the model, the key decisions are (a) the nature of the basis functions, (...) (b) the size of the basis, and (c) the type of translation factors to be employed (see [9], p.133). In our calculations we model the atomic orbital basis states $\chi_k(\vec{r}, t)$ as Hydrogen wavefunctions, when describing the hydrogen atom projectile and as so-called *Rydberg states* when describing the wavefunction of the electron at the ion target. ETF are only applied to the projectile, while the target is assumed to be at rest. A Rydberg state wavefunction for a given charge Z and quantum numbers (n, l, m) in spherical coordinates (r, θ, ϕ) is given by (compare [12], p.436, B.1.3, using the Bohr radius in atomic units $a_0 = 1$):

$$|Z, n, l, m\rangle = \sqrt{\left(\frac{Z}{n}\right)^3 \frac{4\ (n-l-1)!}{n \cdot (n+l)!}} \cdot \left(\frac{2Z}{n} r\right)^l \cdot e^{-\frac{Z}{n}r} \cdot L_{n-l-1}^{2l+1}\left(\frac{2Z}{n} r\right) \cdot Y_{lm}(\theta, \phi) \tag{2.13}$$

The definition (compare [12], p.435, B.1.2) of the associated Laguerre polynomials $L_p^k(x)$ in (2.13) is given by [1]:

$$L_p^k(x) = \sum_{s=0}^{p} (-1)^s \frac{(p+k)!}{(p-s)!(k+s)!\ s!}\ x^s \tag{2.14}$$

[1] Note, that our definitions (2.13) and (2.14) differ slightly from the one used by Messiah in [12]. Messiah defines the Laguerre polynomial $L_p^k(z)$ in (B.13) with an additional factor of $(p+k)!$ in the numerator. For the Laguerre polynomial $L_{n-l-1}^{2l+1}(x)$ used in (B17/2), we can see that this factor $(p+k)! = (n+l)!$ and thus will be compensated by the additional factor $(n+l)!$ in the denominator of (B.17/1), which we do not have in (2.13).

Having set up the suitable basis states to span the truncated Hilbert space in (2.4), we can insert the expansion of Ψ into the Schrödinger equation (2.2), which yields

$$\left(i\frac{\partial}{\partial t} - H_e(\vec{r},t)\right)\sum_{k=1}^{N} a_k(t)|\psi_k(\vec{r},t)\rangle = 0\ . \tag{2.15}$$

We can now fix the amplitudes $a_k(t)$ by requiring the approximate wavefunction (2.4) to obey the time-dependent Schrödinger equation within the space of basis functions (see [10], p.8), in other words: We project an arbitrary basis state ψ_j from the left to get

$$\langle\psi_j|\left(i\frac{\partial}{\partial t} - H_e(\vec{r},t)\right)\sum_{k=1}^{N} a_k(t)|\psi_k\rangle = 0\ , \tag{2.16}$$

which, after performing the time derivation, yields the equation

$$\sum_{k=1}^{N}\langle\psi_j|\psi_k\rangle\ i\frac{d\ a_k(t)}{dt} = \sum_{k=1}^{N}\langle\psi_j|\left(H_e(\vec{r},t) - i\frac{\partial}{\partial t}\right)|\psi_k\rangle\cdot a_k(t)\ . \tag{2.17}$$

In the above equation, one can identify the elements S_{jk} of the $N\times N$ *overlap matrix* and the elements M_{jk} of the $N\times N$ *coupling matrix* as

$$S_{jk} = \langle\psi_j|\psi_k\rangle \quad \text{and} \quad M_{jk} = \langle\psi_j|\left(H_e(\vec{r},t) - i\frac{\partial}{\partial t}\right)|\psi_k\rangle\ . \tag{2.18}$$

Both matrices consist of one-center and two-center parts. (compare [8], section 3.2) The term "one-center" implies that only states defined on the same center of impact are used in the calculation of the respective matrix element, whereas "two-center" indicates states on opposing centers. The indices j and k of the matrices can be associated with either the target or the projectile, depending on which basis states in (2.4) are involved. The basis sets are diagonalized on each center separately leading to two independent orthonormal systems. After the diagonalization the one-center overlap matrix elements form the unit matrix and the two-center overlap matrix elements fulfill

$$S_{PT} = S_{TP}^{\dagger}\ . \tag{2.19}$$

The indices $_{PT}$ indicate that the index j of the matrix corresponds to a state on the projectile, while k corresponds to a target state. Put in a more descriptive way, the overlap and coupling matrices look like

$$\hat{S} = \left(\begin{array}{c|c} \mathbb{1} & S_{TP} \\ \hline S_{TP}^{\dagger} & \mathbb{1} \end{array}\right) \qquad \hat{M} = \left(\begin{array}{c|c} M_{TT} & M_{TP} \\ \hline M_{PT} & M_{PP} \end{array}\right). \tag{2.20}$$

Using the fact that the states $|\psi\rangle$ in (2.18) are eigenstates of the total electronic Hamilton H_e (2.3), as well as eigenstates on the respective centers T and P, the two-center coupling (TWC) matrix elements can be further simplified

$$\begin{aligned} M_{TP} &= {}_T\langle\psi|\left(H_e(\vec{r},t) - i\frac{\partial}{\partial t}\right)|\psi\rangle_P = {}_T\langle\psi|\left(\underbrace{-\frac{1}{2}\nabla^2 + V_P}_{H_P} + V_T - i\frac{\partial}{\partial t}\right)|\psi\rangle_P \\ &= {}_T\langle\psi|\underbrace{H_P|\psi\rangle_P}_{\varepsilon_P|\psi\rangle_P} - {}_T\langle\psi|\underbrace{i\frac{\partial}{\partial t}|\psi\rangle_P}_{\varepsilon_P|\psi\rangle_P} + {}_T\langle\psi|V_T|\psi\rangle_P \\ &= {}_T\langle\psi|V_T|\psi\rangle_P \\ M_{PT} &= {}_P\langle\psi|V_P|\psi\rangle_T \end{aligned} \tag{2.21}$$

The one-center coupling (ONC) matrix elements can be simplified in the same way by following the steps in (2.21):

$$\begin{aligned} M_{TT} &= {}_T\langle\psi|V_P|\psi\rangle_T \\ M_{PP} &= {}_P\langle\psi|V_T|\psi\rangle_P \end{aligned} \tag{2.22}$$

This gives rise to three different types of matrix elements that occur during the calculation:

1. two-center overlap matrix elements S_{TP} and S_{PT}
   ${}_P\langle\psi|\psi\rangle_T = {}_T\langle\psi|\psi\rangle_P^\dagger$
2. TWC matrix elements M_{TP} and M_{PT}
   ${}_T\langle\psi|\ V_T\ |\psi\rangle_P$ and ${}_P\langle\psi|\ V_P\ |\psi\rangle_T$
3. ONC matrix elements M_{PP} and M_{TT}
   ${}_P\langle\psi|\ V_T\ |\psi\rangle_P$ and ${}_T\langle\psi|\ V_P\ |\psi\rangle_T$

The involved potentials are simply given by the Coulomb potential $V = Z/r$ of the respective core, i.e. (using the charge $Z_{P/T}$)

$$V_P = \frac{Z_P}{|\vec{r}_P|} \qquad \text{and} \qquad V_T = \frac{Z_T}{|\vec{r}_T|}\ , \tag{2.23}$$

with $|\vec{r}_P|$ and $|\vec{r}_T|$ giving the magnitude of the position vector of the active electron in the coordinate system of the projectile center, or target center respectively.

With the definitions (2.18) and writing all expansion coefficients (i.e. probability amplitudes) $a_k(t)$ into the vector $\vec{a}$

$$\vec{a}(t) = \begin{pmatrix} a_1(t) \\ \vdots \\ a_k(t) \\ \vdots \\ a_N(t) \end{pmatrix}\ , \tag{2.24}$$

we can rewrite (2.17) into a matrix equation

$$\begin{aligned} i\ \hat{S}\ \frac{d}{dt}\vec{a} &= \hat{M}\ \vec{a} \\ \frac{d}{dt}\vec{a} &= \underbrace{-i\hat{S}^{-1}\hat{M}}_{\hat{M}_{\text{eff}}}\vec{a}\ , \end{aligned} \tag{2.25}$$

where the *effective matrix* $\hat{M}_{\text{eff}}$ is defined. These equations are the so-called *coupled-channel equations*, which are a set of coupled ordinary differential equations of first order. To solve for the probability amplitudes a_k, (2.25) has to be integrated along the classical trajectory describing the motion of the projectile core. As we have stated above, the basis functions $\psi_i(\vec{r},t)$ and $\psi_f(\vec{r},t)$ represent the initial state i and a given final state f of the electron. The initial values of the respective probility amplitudes are thus given by

$$a_i(t=-\infty) = e^{i\alpha} \qquad \text{and} \qquad a_f(t=-\infty) = 0\ , \tag{2.26}$$

with α being an arbitrary real phase. After the collision the transition probability is given by

$$P_{i\to f}(\vec{v},\vec{b}) = |a_f(t=+\infty)|^2 \ , \tag{2.27}$$

yielding the integrated partial cross section as (see [10], (2.5) on p.8)

$$\sigma_{i\to f}(\vec{v}) = 2\pi \int_0^\infty db\ bP_{i\to f}(\vec{v},\vec{b}) \ . \tag{2.28}$$

This of course means that for a given value of $\vec{v}$, the amplitudes $a_k(t)$ have to be calculated for all (relevant) values of $\vec{b}$ in order to be able to calculate the integral (2.28).

2.2 Computational Implementation

The AOCC theory described in section 2.1 is implemented in the parallel FORTRAN code "sic3ma". The current version of the code described in this section was mainly written by Igenbergs during her PhD thesis [8] and complemented by Wallerberger in the scope of his Master thesis [11].

To obtain nl-resolved cross sections the probability amplitudes $a_{nl}(b)$ for the population of a given final state (at $t = +\infty$) with quantum numbers n and l at a given impact parameter b have to be calculated. Then the quadratic norm of these complex amplitudes $a_{nl}(b)$ has to be integrated over all impact parameters b, as described in (2.28). This can be done for various impact velocities (i.e. impact energies of the projectile). The needed probability amplitudes $a_{nl}(b)$ can be obtained by solving the coupled-channel equation (2.25) for every given impact parameter b. The calculation is started with an initial condition $a_{nl}(b, t = -\infty)$, but since we are interested in the final amplitude $a_{nl}(b, t = +\infty)$, we need to solve the coupled-channel equation step by step from $t = -\infty$ to $t = +\infty$. This is equivalent to calculating $a_{nl}(b, t)$ along the core's straight classical trajectory, which is parameterized by r_3. For each impact parameter b, there is a different trajectory as can be seen in fig. 2.2. The projectile moves along the trajectory with the constant given impact velocity v. In the computational implementation the trajectory obviously does not stretch from $-\infty$ to $+\infty$, but from a reasonably distant point at negative r_3 to a reasonably distant point at positive r_3. Additionally a finite mesh over all interesting values of b and v is chosen.

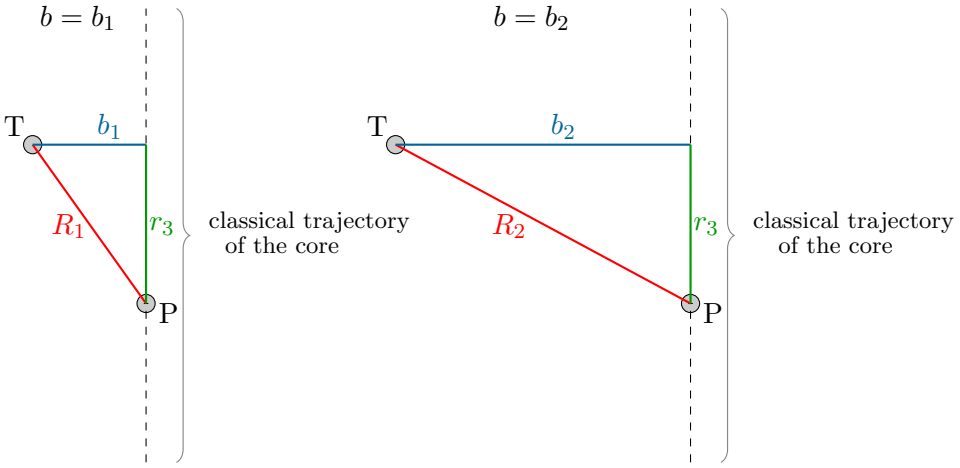


Figure 2.2: Geometry for the code "sic3ma" for two different values of b: The projectile core P moves along its classical trajectory (parameterized by r_3) with a constant given impact velocity. In theory, the trajectory would be infinite but this is approximated by choosing some finite maximum value r_{max} for the parameter r_3. This yields a range for r_3 of $-r_{max} \leq r_3 \leq +r_{max}$. Every value b_i in the b-mesh yields a different trajectory and thus a different R for a given point r_3 on the trajectory.

To calculate $a_{nl}(t)$ at every point along a given trajectory, the matrices $\hat{M}$ and $\hat{S}$ need to be known (in order to calculate the effective matrix $\hat{M}_{\text{eff}}$) at each of the trajectory's points r_3.

It is possible to use time symmetry and detailed balance [13] in order to get a relationship for both matrices $\hat{M}$ and $\hat{S}$ between points $+r_3$ and points $-r_3$ (see [8], p.28, section 4.2).

$$M_{ij}(-r_3) = (-1)^{l_i+m_i+l_j+m_j} M_{ij}(+r_3) \qquad S_{ij}(-r_3) = (-1)^{l_i+m_i+l_j+m_j} S_{ij}(+r_3) \tag{2.29}$$

This means, the matrices only have to be calculated for half the trajectory and (2.29) can be used to get the second half. Still, every impact parameter b gives a different trajectory which is travelled through at different speeds for every impact velocity v and thus the elements of $\hat{M}$ and $\hat{S}$ need to be calculated for all relevant values of r_3, b and v. This yields a huge amount of matrices, making parallel algorithms necessary. The code "sic3ma" implements the AOCC method described in section 2.1 in various separate steps (a very detailed description of this implementation can be found in [8], chapter 4 and [11], chapter 3):

1. Creation of symbolic structures for the matrix elements
2. Evaluation of the matrix elements of $\hat{M}$ and $\hat{S}$
3. Calculation of the effective matrix $\hat{M}_{\text{eff}}$
4. Calculation of the probability amplitudes $a_{nl}(b)$
5. Calculation of the nl-resolved cross sections

These five steps will now be explained in more detail:

2.2.1 Step 1: Creation of Symbolic Structures for the Matrix Elements

As explained above, the elements of $\hat{M}$ and $\hat{S}$ are calculated for all relevant values of r_3, b and v. To this end, they are stored in so-called *symbolic structures*, meaning that an equation rather than a number is stored. Storing an equation makes it easy and fast to subsequently evaluate the symbolic structures for all necessary points r_3 on every trajectory (one trajectory for every value of b) and for all needed values of v (this evaluation will be described in step 2). The symbolic structures are calculated using the Fourier transform method originally suggested by Shakeshaft [14], but in a more modern implementation [15]. With this method, the symbolic structures are calculated and stored in one file for each matrix element separately. The files for elements of the coupling matrix $\hat{M}$ carry the extension "-.rbM_ixj", with $1 \leq i \leq nstat$ and $1 \leq j \leq nstat$, while elements of the overlap matrix $\hat{S}$ carry the extension "-.rbS_ixj", with $1 \leq i \leq ntarg$ and $j > ntarg$. Only one quadrant of the overlap matrix $\hat{S}$ has to be calculated, since the other quadrants can then be obtained using (2.19) and (2.20). To have the elements written into files is beneficial, since the code "sic3ma" is a parallel code and the size of all these structures can become too large to efficiently broadcast them between parallel processors. With all structures being stored in a file, they can be accessed later by different parallel cores, whenever this is needed.

We now want to get a rough estimate of how much space these structures might take up for larger calculations. Symbolic structures for all coupling matrix elements and for the two-center overlap matrix elements (as seen in (2.20), the one-center overlap matrix elements simply give the unit matrix) need to be calculated and stored. This yields n_{stat}^2 symbolic structures for the

matrix elements of $\hat{M}$ and $n_{targ} \cdot n_{proj}$ for the matrix $\hat{S}$, with n_{targ} and n_{proj} being the number of target (respectively projectile) states and $n_{stat} = n_{targ} + n_{proj}$ being the total number of basis states. This yields $n_{stat}^2 + n_{targ}n_{proj} \approx n_{stat}^2$ symbolic structures in total. The average size of such a structure is roughly 500 KB. To describe a large system like e.g. Ar^{17+} sufficiently, one needs at least $n_{stat} = 10^3$ basis states, which leads to an average size of $10^3 \cdot 10^3 \cdot 5 \cdot 10^2$ KB $= 5 \cdot 10^2$ GB.

In order to make the calculation of such huge amounts of symbolic structures efficient, Igenbergs has implemented an elaborate algorithm featuring an event dispatch loop that reduces idle times to a minimum. The main processes (rank 0) tells the secondary processes (rank $\geq$ 1) to calculate the symbolic structure for a specified matrix element. The secondary processes perform the calculation, write a file with the symbolic structure, report back to the distributor main rank, and receive a new assignment. This goes on until the symbolic structures for all matrix elements have been calculated. (see [8], p.28, section 4.2)

The amount of parallel ranks that can be used in this step is basically unlimited, one must however consider the phenomenon of *parallel file I/O*, where performance is heavily impaired due to too many ranks simultaneously opening files, writing to files or reading from files. In oder to at least partially circumvent this problem for the following step 2 in the calculation, the files ".rbM_*i*x*j*" and ".rbS_*i*x*j*" are combined to one file per column of the matrices $\hat{M}$ and $\hat{S}$. This means that all rbM-files and rbS-files with a given j are written into one single file, which then carries the extension "-.rbc$j$" (c for column), where $1 \leq j \leq nstat$. The file ".rbc1" then holds the first column for the matrix $\hat{M}$, while the file "rbc.$ntarg + 1$" holds the first column of matrix $\hat{S}$ and column number $ntarg + 1$ of matrix $\hat{M}$. Combining the single rbM-files and rbS-files into rbc-files is beneficial for reading those files in before evaluation in step 2, since only $nstat$ rather than $nstat^2$ single files have to be read in.

2.2.2 Step 2: Evaluation of the Matrix Elements of $\hat{M}$ and $\hat{S}$

Once all the symbolic structures for the elements of $\hat{M}$ and $\hat{S}$ are obtained by step 1, they are then evaluated for all points on all given trajectories and for all impact velocities. Due to the limited memory available on one single parallel process, it is not possible to store all structures on a single rank as they can account for several GB (see estimation above). Hence, the symbolic structures are distributed over all ranks using a column-wise distribution, where each rank is only responsible for a subset of the rbc-files described above. It will be shown in section 3.4.1, that a simple column-wise distribution will not suffice for larger systems and a more optimized distribution of the symbolic structures over all available ranks has to be found.

After distributing all symbolic structures an elaborate mechanism for the evaluation is used, where each rank is responsible for only its subset of the symbolic structures. The rank evaluates its assigned matrix elements for all given values of b and v before continuing with step 3.

In the code step 2 and 3 are actually implemented together inside a loop over the parameter r_3, so that all necessary values are calculated. (see [8], section 4.2)

2.2.3 Step 3: Calculation of the Effective Matrix $\hat{M}_{\text{eff}}$

In step 2 each rank evaluated a certain amount of symbolic structures for all values of b and v. Once this is done, each rank is assigned a pair of (b,v) for which it will calculate the effective matrix $\hat{M}_{\text{eff}}$. In oder to do so, the ranks need to "collect" all evaluated matrix elements for their assigned pair (b,v). This is achieved by an MPI broadcast scheme in the following way (see [8], section 4.2):

1. Each rank has its subset of matrix elements evaluated for all possible pairs (b,v) and now broadcasts those evaluated elements to the ranks responsible for each given pair (b,v).

2. Each rank keeps its evaluated subset of matrix elements for its own pair (b,v).

3. Each rank gradually receives all other subsets of evaluated matrix elements for its pair (b,v), eventually having a complete matrix $\hat{M}$ and one quadrant of the matrix $\hat{S}$.

Then each rank first builds the complete matrix $\hat{S}$ using (2.19) together with (2.20) and then subsequently calculates the effective matrix $\hat{M}_{\text{eff}}$

$$\hat{M}_{\text{eff}} = -i\hat{S}^{-1}\hat{M} \tag{2.30}$$

by matrix inversion of $\hat{S}$ and matrix multiplication with $\hat{M}$. In the code this step is actually combined with step 2, as has already been mentioned above. These two steps are then again within a loop over every point r_3 on the trajectory, making it possible to split the calculation into several junks of values r_3 for scheduling purposes on cluster computers.

In a realistic calculation there are 400 trajectory points r_3 and on each of these 400 points one matrix $\hat{M}_{\text{eff}}$ is calculated for each pair (b,v). Roughly $nv = 12$ different impact velocities v and $nb = 24$ impact parameters b will be used, which yields 288 matrices for each trajectory point and $1,152 \cdot 10^5$ matrices in total. Each matrix is written into a file. These files of course take up a considerable amount of space on the hard disc. With an estimated 10^3 basis states, each matrix is of size 10^3 x 10^3 and each matrix element is a complex*16 number, requiring 16 bytes, yielding $16 \cdot 10^6$ bytes per matrix file. The total required hard disc capacity is then $1,152 \cdot 10^5 \cdot 16 \cdot 10^6 \approx 2 \cdot 10^{12}$ byte ≈ 2 TB.

The amount of parallel ranks that can be used in step 2 and 3 are basically again unlimited, however one must consider that during step 3 only those ranks assigned with a pair (b,v) will calculate $\hat{M}_{\text{eff}}$, while all others (those with rank $> nv \cdot nb$) will be idle. The optimal number of parallel ranks would thus be $nv \cdot nb$, provided that this number of ranks is sufficiently high to accommodate all symbolic structures for step 2. If this is not the case and step 2 needs a higher number of ranks to have enough memory for all structures (memory needs during step 2 will be discussed in more detail in section 3.4.1), then one simply has to accept some idle time for all those ranks not involved in step 3.

In the end the calculated matrices $\hat{M}_{\text{eff}}$ are written into files, so that they are available in step 4. The files carry the extensions "-l.k", where l counts the pair (b,v) and k gives the

number of the trajectory point r_3. The number l counts in the following way:

$$\begin{aligned} l = 0 &\rightarrow (b_1, v_1) \\ l = 1 &\rightarrow (b_1, v_2) \\ \vdots &\rightarrow \vdots \\ l = nv - 1 &\rightarrow (b_1, v_{nv}) \\ \vdots &\rightarrow \vdots \\ l = nv \cdot nb - 1 &\rightarrow (b_{nb}, v_{nv}) \end{aligned} \tag{2.31}$$

2.2.4 Step 4: Calculation of the Probability Amplitudes $a_{nl}(b, v)$

The probability amplitudes are computed for every given impact parameter and every given impact velocity by numerically integrating the coupled ordinary differential equations of first order (2.25) stepwise along the given trajectory (there is one trajectory for every impact parameter b, see fig. 2.2). To speed up the computation in this step, the code uses a simple parallel algorithm. Similar to step 3, each rank is assigned a pair (b,v) and integrates along the trajectory (defined by b) using its assigned impact velocity v. During the integration, all needed values for the matrix elements of $\hat{M}_{\text{eff}}$ are interpolated using the values calculated during step 3. It is therefore very important for the values of the matrix elements to be continuous along the trajectory (compare to section 4.3). The results of this step are probability amplitudes $a_{nl}(b, v)$ for every given b and v. (see [8], chapter 4)

The total number of parallel ranks therefore has to be $nv \cdot nb$.

2.2.5 Step 5: Calculation of the nl-resolved Cross Sections

Following (2.28) this calculation is done by integrating $|a_{nl}(b, v)|^2$ over all values of b, yielding the nl-resolved cross section $\sigma_{nl}(v)$ for a given impact velocity. Since this integration is significantly less complex than all previous steps of the computation, it can simply be done locally using cubic splines (see [8], chapter 4).

Chapter 3

Improvement of the Computation

This chapter will describe efforts to optimize the computational implementation with respect to numerical accuracy as well as computational performance. These optimizations become necessary when studying larger collisional systems. In this chapter, we focus on the system $Ar^{18+} + H$ as an example of such a large system.

Section 3.1 will first outline some of the limits of the current code "sic3ma" and suggest ways to push these limits further. Section 3.2 and section 3.3 are dedicated to the study and improvement of the numerical calculation of the matrix elements (2.18). To identify and reduce numerical instabilities in these calculations, a profound systematic study was conducted. As discussed in section 2.1, the matrix elements are classified, depending on the states involved in their computation, as *one-center* (will be discussed in section 3.2) and *two-center* elements (treated in section 3.3). Several improvements could be achieved which pushed the sic3ma code's previous limitations.

Finally, section 3.4 will discuss hardware requirements for the calculations, computational considerations and performance optimization.

3.1 Ways to Push the Limits of sic3ma

This section will give an overview over some of the limits and problems of the current implementation of sic3ma and suggest ways to overcome them. Section 3.1.1 starts by discussing the problem of cancellation errors, while section 3.1.2 talks about the effect of changing the precision of the involved numbers. Section 3.1.3 then outlines ways to get reference values for the matrix elements and section 3.1.4 shows the improvement of the implementation of the Gamma Function.

In order to sufficiently describe a collisional system like $Ar^{18+} + H$, an appropriate basis set has to be determined. While it is enough to have bound states up to $n = 3$ (to account for excitation) and some unbound pseudo states (modeling ionization into the continuum) on the Hydrogen center, it is necessary to include eigenstates for much higher n on the Argon ion. The visible spectral lines of Ar^{17+} most suitable for CXS are $n = 18 \to 17$, $17 \to 16$, $16 \to 15$ and $15 \to 14$. In order to realistically estimate the transition probabilities for de-excitation from an initial state (e.g. $n = 18$) to a final state (e.g. $n = 17$), one has to consider that the population of the initial state not only originates from a direct CX into this state, but also from electrons de-exciting by cascade effects from higher n-shells into the initial state. It is therefore advisable to include one or two higher n-shells than the initial state of interest. For example,

if one would like to estimate the transition from $n = 18 \rightarrow 17$, one should include states up until $n = 20$ to account for cascade effects. Unfortunately, such high quantum numbers n pose some computational difficulties. In her PhD thesis, Igenbergs therefore calculated the system $Ar^{18+} + H(1s)$ including only states up to $n = 15$ (see [8], section 5.6) on the Argon ion and the so-called "H19" basis expansion on the Hydrogen atom including Hydrogen states for $1 \leq n \leq 3$ and a few unbound pseudo states (see [8], section 5.5). For higher values of n, the cross sections did not compute to a reasonable value or the computation failed entirely, because the integration of (2.25) along the trajectory did not converge. The reason for these numerical stabilities was soon suspected to be the evaluation of the matrix elements. When the matrix elements do not evaluate to a reasonably correct value, neither can the cross sections or, even worse, the computation fails completely, since numerical boundaries are hit.

3.1.1 Cancellation Errors and Various Precisions

Once source for errors in numerical evaluations are so-called *cancellation errors* (see [16], page 41, Bsp. 2.38). These occur, when a small value is calculated as a sum of very large numbers with opposite signs. If these very big numbers are not accurate enough, then the result will be wrong. As an example, one can take a look at the following summation:

$$\begin{array}{r} 100000004{,}226 \\ -100000004{,}224 \\ \hline 0{,}002 \end{array} \tag{3.1}$$

Here, the two large numbers have to be accurate to the last digits in order for the result to be correct. If the accuracy is reduced by rounding the last digit, the result is already wrong by almost an order of magnitude:

$$\begin{array}{r} 100000004{,}23 \\ -100000004{,}22 \\ \hline 0{,}01 \end{array} \tag{3.2}$$

To circumvent such errors, one possibility is to increase the precision (i.e. size of the memory) for the involved numbers. A *single precision* number has a size of 4 bytes = 32 bits and can store ≈ 7 digits. If the result of a summation like (3.1) is of the order 10^{-3}, then terms with alternating signs in the sum must not be greater then 10^3, otherwise the digits do not fit into the 7 digits of a single precision number and the accuracy is lost as in the example (3.2). If one increases the precision to a 8 bytes = 64 bits *double precision* number, one can store ≈ 15 digits while a 16 bytes = 128 bits *quadruple precision* number can store ≈ 34 digits. In this case, the terms in the sum may be as large as 10^{30}. Of course not only the mere amount of digits (i.e. type of precision) is of importance but also the accuracy of the involved calculations, as will be seen in section 3.2.1 and section 3.2.5.

3.1.2 Changing the Precision of the Involved Numbers

Since a lot of sums of large numbers with alternating signs occur during the evaluation of the matrix elements in sic3ma, cancellation errors are significant and it can thus be expected that

changing the precision of the real numbers involved in the calculation might have a huge impact on the quality and stability of the computation. To get an idea of the involved matrix elements in the calculation of the system $Ar^{18+} + H(1s)$, we quickly want to repeat them:

1. two-center overlap matrix elements:
   $\langle Ar^{18+}|H\rangle = \langle H|Ar^{18+}\rangle^{\dagger}$

2. TWC matrix elements:
   $\langle Ar^{18+}|\ V_{Ar}\ |H\rangle$ and $\langle H|\ V_H\ |Ar^{18+}\rangle$

3. ONC matrix elements:
   $\langle Ar^{18+}|\ V_H\ |Ar^{18+}\rangle$ and $\langle H|\ V_{Ar}\ |H\rangle$

Using (2.23) the Coulomb potentials of the respective centers are given by

$$V_H = \frac{1}{|\vec{r}_H|} \quad \text{and} \quad V_{Ar} = \frac{18}{|\vec{r}_{Ar}|}\ . \tag{3.3}$$

It is obvious, that the calculation of $\langle H|\ V_{Ar}\ |H\rangle$ will never reach the critical quantum number around $n = 15$, since there are only states up to $n = 3$ included in the basis expansion on the hydrogen center. All other elements do however include Argon states and since Argon should be included up to $n = 20$, we can suspect errors to arise there. For a quick overview, where numerical problems might occur, all three kinds of matrix elements were evaluated for some selected states, namely Argon s-states $|Z, n, l, m\rangle = |18, n, 0, 0\rangle$ with $1 \leq n \leq 26$ and the Hydrogen $1s$ state. The impact parameter b and the point on the trajectory r_3 were arbitrarily chosen to be $b = 5$ and $r_3 \approx -5$, yielding an internuclear distance $|\vec{R}| = \sqrt{b^2 + r_3^2} \approx 7$ (compare with the geometry in fig. 2.2).

In the original implementation of sic3ma all real values involved in the evaluation of the matrix elements were defined as double precision numbers. Since new versions of compilers supporting quad (quadruple) precision numbers are available, implementing quad precision numbers in the evaluation seems like a reasonable path to go forward. To see if this is indeed promising, quick evaluations were done using first single precision numbers and then double precision numbers for the involved real numbers. If there is an improvement by changing from single to double precision, it is justified to extrapolate this and expect an improvement also for a change from double to quad precision.

In fig. 3.1, it can be seen how the change from single to double precision improves the calculations of ONC elements $\langle Ar(n, 0, 0)|\ V_H\ |Ar(n, 0, 0)\rangle$. When single precision is used, the values only remain stable up to the element for $n = 6$, while when using double precision the values still seem reasonable for the element with $n = 12$. In the same way, fig. 3.2 shows the effect of changing from single to double precision for the two-center overlap matrix elements in fig. 3.2(a) and for the TWC matrix elements in fig. 3.2(b). The limit for single precision is $n = 11$, while the limit when using double precision is pushed as high as $n = 20$. Thus, the calculation of two-center elements seems already good enough for double precision numbers. However, it needs to be considered here that the calculations yielding fig. 3.1 and fig. 3.2 only give a quick overview into where numerical problems might occur. They serve no purpose other then to show that it is indeed justified to expect an improvement in the calculations when changing the precision from double to quad precision.

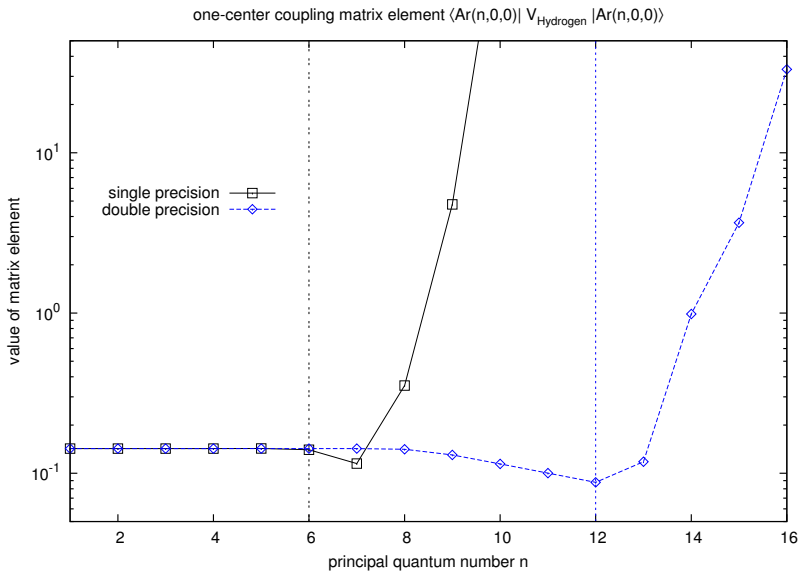
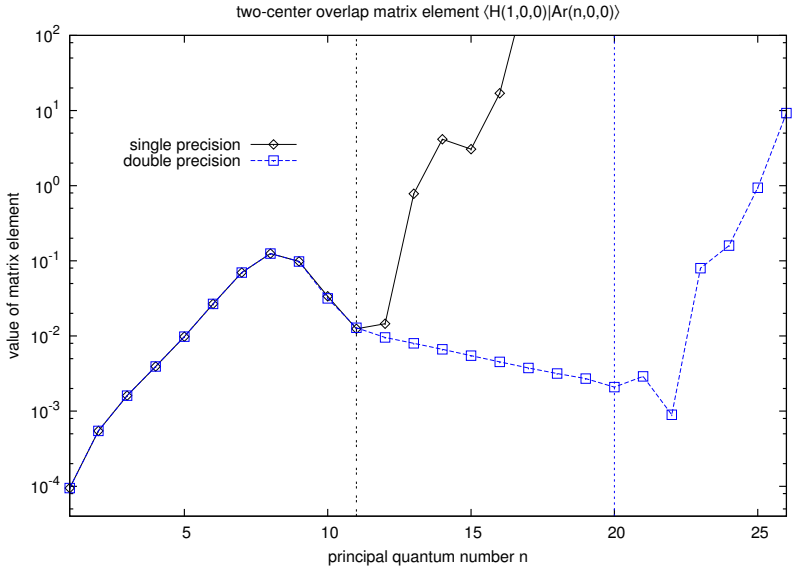
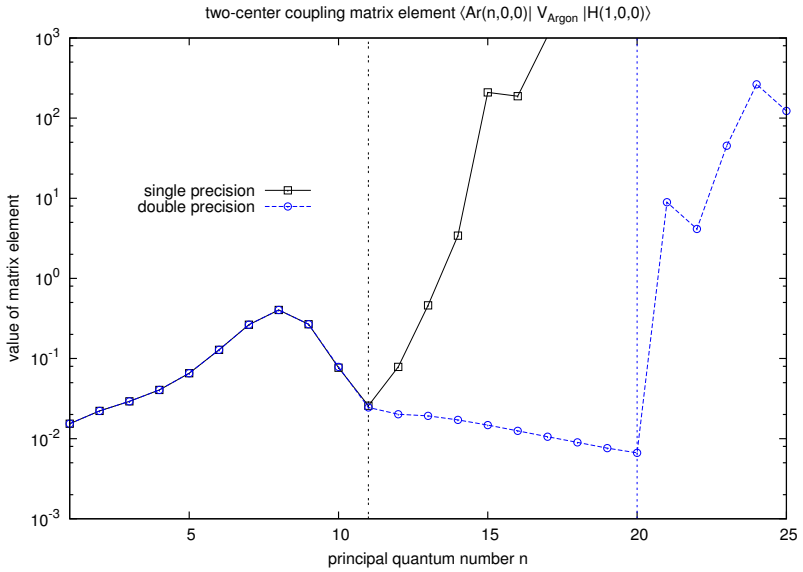


Figure 3.1: Calculation of ONC matrix elements $\langle Ar(n,0,0)|\ V_H\ |Ar(n,0,0)\rangle$ for $1 \leq n \leq 16$ and $b = 5$, $r_3 \approx -5$, $R \approx 7$ (compare with geometry in fig. 2.2) using single and double precision numbers. The dashed vertical lines show for which n the calculations start to become unstable and yield wrong results. This limit is at $n = 6$ for single precision and at $n = 12$ for double precision. When calculating the collisional system $Ar^{18+} + H$ states up to $n = 20$ should be included and thus this limit is not high enough. However, since it is pushed from $n = 6$ to $n = 12$ when changing from single to double precision, implementing quad precision numbers seems a reasonable path to further push the limit towards higher values of n.



(a)



(b)

Figure 3.2: Fig. 3.2(a) shows the calculation of the two-center overlap elements $\langle H(1,0,0)|Ar(n,0,0)\rangle$ with $n \leq 26$, while fig. 3.2(b) shows the TWC matrix elements $\langle Ar(n,0,0)|\ V_{Ar}\ |H(1,0,0)\rangle$ for $n \leq 25$. All results were calculated using single and double precision numbers and the values $b = 5$, $r_3 \approx -5$, $R \approx 7$ (compare with geometry in fig. 2.2). The dashed vertical lines show for which n the calculations start to become unstable and yield wrong results. In both figures, the limit is at $n = 11$ for single precision and at $n = 20$ for double precision.

Furthermore, it remains to be determined, which of the steps in calculating the matrix elements (as they are described in section 2.2) need to be conducted in which precision. Is it enough to calculate the symbolic structures (step 1) using quad precision or does the evaluation of the symbolic structures for given values of b, r_3 and v itself (step 2) also require a higher precision? Also, in which precision do the symbolic structures have to be stored, before they are used to evaluate the matrix elements? As discussed in section 2.2, the symbolic structures for a system like Ar^{17+} need $\approx 10^2$ GB and having to store them in quad precision instead of double precision would double the required amount of space. To answer these questions, a more detailed and systematic study on the effects of changing the precisions are given in section 3.2.4.

3.1.3 Compare Calculated Results to Reference Values

The first quick evaluations of matrix elements, like the ones shown in fig. 3.1 and fig. 3.2, simply served the purpose of assessing whether numerical instabilities occur and where they might be encountered. Even though the correct values for the matrix elements in fig. 3.1 and fig. 3.2 are unknown, it is safe to assume that all values larger then ≈ 10 are certainly wrong. In addition, it seems wrong that the values for higher n suddenly differ so significantly from those for lower n. Still, for a more detailed analysis and to actually check the calculated values, it is necessary to find other means of evaluating the matrix elements.

There were two ways used to calculate reference values: using the computational software program "Mathematica" and making an analytical calculation. The analytic calculation is done using a subroutine called "onc_coupling" (explained in more detail in section 3.2.1), which is however limited to ONC matrix elements involving only s-states. Mathematica on the other hand is in principal able to calculate all needed matrix elements to arbitrary accuracy, as long as one figures out the geometrical description of the two centers in a common coordinate system. Unfortunately, the calculation in mathematica is far too slow to actually use it in the computation. It takes mathematica in the order of 10^2 seconds to evaluate a typical matrix element, while the implementation in sic3ma needs in the order of 10^{-2} seconds.

3.1.4 New Implementation of the Gamma Function

In the existing code sic3ma, the *normalized lower incomplete gamma function* $P(s,x)$ and the *gamma function* $\Gamma(s)$ were implemented following the Numerical recipes (see [17], p. 160). The two functions $P(s,x)$ and $\Gamma(s)$ are defined as:

$$P(s,x) = \frac{1}{\Gamma(s)} \int_0^x dt\ t^{s-1}e^{-t} \qquad \text{with} \qquad \Gamma(s) = \int_0^\infty dt\ t^{s-1}e^{-t} \tag{3.4}$$

It turned out, that the precision of the implementation of the gamma function was not sufficient to avoid cancellation effects becoming significant and since $\Gamma(s)$ is used to normalize $P(s,x)$, the inaccuracies were passed on. Thus, a more precise implementation of $\Gamma(s)$ was needed. If the gamma function's argument is an integer value n, then $\Gamma(n)$ can easily be calculated as

$$\Gamma(n) = (n-1)! \quad , \ \forall\ n \in \mathbb{N} \tag{3.5}$$

The occurring values of s in the calculation of (3.4) are in fact always integer values. In addition to that, the implementation of the normalized lower incomplete gamma function actually

did not use $\Gamma(n)$ to normalize the value, but the natural logarithm $\ln[\Gamma(n)]$. Using the definition (3.5) and the well known relation $\ln(a \cdot b) = \ln a + \ln b$, the logarithm of the gamma function for integer arguments can easily be calculated:

$$\begin{aligned} \ln\Big[\Gamma(n)\Big] &= \ln\Big[(n-1)!\Big] = \ln\Big[(n-1)\cdot(n-2)\cdot\ \ldots\ \cdot 3\cdot 2\Big] = \\ &= \Big[\ln(n-1) + \ln(n-2) + \ \ldots\ + \ln 3 + \ln 2\Big] \\ &= \sum_{i=2}^{n-1} \ln i \end{aligned} \tag{3.6}$$

The above result was implemented to calculate the logarithm of the gamma function as a sum of logarithms of integer values and this enormously improved the accuracy of $\ln[\Gamma(n)]$ and subsequently the normalized incomplete gamma function as seen in table 3.1. For comparison the values were also calculated with Mathematica.

x=11.045361017187260774210913843344 3

s	*Mathematica*	new $P(s,x)$	old $P(s,x)$
14	0.22292622186022405	0.22292622186022405	0.222926221 **90685377**
13	0.31627727947036229	0.31627727947036229	0.316277279 **53626034**
12	0.42614817885005818	0.42614817885005818	0.426148178 **93822726**
11	0.54551510497449724	0.54551510497449724	0.54551510 **508613021**
10	0.66439181577875837	0.66439181577875837	0.66439181 **571117034**
9	0.77201773277620342	0.77201773277620342	0.7720177327 **3131158**
8	0.85971366697606819	0.85971366697606819	0.8597136669 **4929352**
7	0.92323060159751117	0.92323060159751117	0.9232306015 **8347947**
6	0.96348447303351467	0.96348447303351467	0.9634844730 **2723372**
5	0.98535095872286390	0.98535095872286390	0.98535095872 **055467**
4	0.99524945169893033	0.99524945169893033	0.995249451698 **27295**

Table 3.1: Improving the accuracy of the implementation $P(s,x)$ of the normalized lower incomplete gamma function. The new implementation agrees with the exact solution from mathematica up to the 33^{rd} digit (not shown here, as there is not enough space). The old implementation on the other hand only agrees up to the 10^{th} digit, as indicated by the **bold digits** in the last column. Such a low accuracy is of course quite problematic as far as cancellation errors are concerned.

The new $P(s,x)$ agrees with the exact solution from mathematica up to the 33^{rd} digit (not shown here for space reasons), which are all available digits for a quad precision number. Thus it is as precise as possible. However, the old implementation only agrees up to the 10^{th} digit (as indicated by the **bold digits** in the last column), which is by far not enough to avoid cancellation errors.

3.2 One-center Coupling Matrix elements

In this section we will focus on the improvement of calculating ONC matrix elements, where states on the center A couple to the potential of center B. The centers A and B can then be substituted with either the target or the projectile, depending on which matrix element is of interest. The general form of such elements, which are always given by real numbers, is the following:

$$ {}_A\langle\phi_\alpha|\ V_B\ |\phi_\beta\rangle_A = \ {}_A\langle Z_A, n_\alpha, l_\alpha, m_\alpha|\ \frac{Z_B}{|\vec{r}_B|}\ |Z_A, n_\beta, l_\beta, m_\beta\rangle_A \tag{3.7}$$

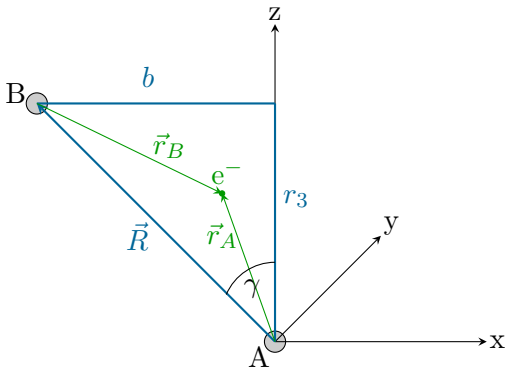


Figure 3.3: The geometry of ONC matrix elements. In the given considerations, the elements are calculated using states on center A and the potential from center B. Everything is expressed in terms of spherical coordinates (r, θ, ϕ) in coordinate system A. The element is thus given by ${}_A\langle Z_A, n_\alpha, l_\alpha, m_\alpha|\ \frac{Z_B}{|\vec{r}_A - \vec{R}|}\ |Z_A, n_\beta, l_\beta, m_\beta\rangle_A$.

From very simple geometric considerations in fig. 3.3, we can see that $\vec{r}_B = \vec{r}_A - \vec{R}$ and can thus express the matrix elements only in terms of coordinates on center A.

$$M_{\alpha\beta} = \ {}_A\langle Z_A, n_\alpha, l_\alpha, m_\alpha|\ \frac{Z_B}{|\vec{r}_A - \vec{R}|}\ |Z_A, n_\beta, l_\beta, m_\beta\rangle_A \tag{3.8}$$

There are various ways to calculate $M_{\alpha\beta}$. An analytical calculation for the special case when only s-states are involved is described in section 3.2.1, while section 3.2.2 elaborates how reference values can be obtained using mathematica. Section 3.2.3 then explains the implementation within the code "sic3ma". In section 3.2.4, it is determined which precision is required for the individual steps in the calculation and section 3.2.5 finally summarizes the achieved improvements and discusses the current limits of calculating ONC matrix elements using "sic3ma".

3.2.1 Analytical Calculation for s-states

In the special case that both states in (3.8) are s-states (i.e. $l_{\alpha/\beta} = m_{\alpha/\beta} = 0$), the matrix element can be calculated analytically. This calculation is implemented in the subroutine "onc_coupling" and will be used to obtain reference values and compare those values to results given by sic3ma as well as calculations done with mathematica. The calculation is only briefly

described here, for more detail refer to Appendix A.

Spherical coordinates on center A are chosen, i.e. $\vec{r}_A = \vec{r}_A(r,\theta,\phi)$ and $\vec{R} = \vec{R}(r,\theta,\phi)$. We then follow the steps taken in [18] (section 3.3) and first expand the s-states $|\phi_\alpha\rangle$ and $|\phi_\beta\rangle$ into so-called *Slater Orbitals* in the following way (see [18], eq.(3.3.4)):

$$\begin{aligned} |\phi_\alpha\rangle = |Z_A, n_\alpha, 0, 0\rangle &= \sum_{i=1}^{\infty} N_i^\alpha r^{a_i^\alpha} e^{-b_i^\alpha r} \cdot Y_{00} \\ |\phi_\beta\rangle = |Z_A, n_\beta, 0, 0\rangle &= \sum_{j=1}^{\infty} N_j^\beta r^{a_j^\beta} e^{-b_j^\beta r} \cdot Y_{00} \end{aligned} \tag{3.9}$$

Here r denotes the radial distance, $N_{i/j}^{\alpha/\beta}$ are normalization coefficients, $a_{i/j}^{\alpha/\beta}$ and $b_{i/j}^{\alpha/\beta}$ are coefficients determining the radial shape of the wavefunction and $Y_{00} = 1/\sqrt{4\pi}$ is the spherical harmonic for s-states. We also expand V_B using Legendre-Polynomials P_k and the angle θ between the two vectors $\vec{r}_A$ and $\vec{R}$ (see [18], eq.(3.3.12)):

$$V_B = \frac{Z_B}{|\vec{r}_A - \vec{R}|} = \begin{cases} \frac{Z_B}{r} \sum_{k=0}^{\infty} \left(\frac{R}{r}\right)^k P_k(\cos\theta), & r > R \\ \frac{Z_B}{R} \sum_{k=0}^{\infty} \left(\frac{r}{R}\right)^k P_k(\cos\theta), & r < R \end{cases} \tag{3.10}$$

The expansions (3.9) and (3.10) are then inserted into (3.8) and some orthogonality relations [1], as well as the *lower incomplete gamma function* $\gamma(s,x)$ and the *upper incomplete gamma function* $\Gamma(s,x)$ are used. The incomplete gamma functions are defined as:

$$\begin{aligned} \gamma(s,x) &= \int_0^x dt\ t^{s-1} e^{-t} \\ \Gamma(s,x) &= \int_x^\infty dt\ t^{s-1} e^{-t} \end{aligned} \tag{3.11}$$

This finally gives the expression

$$M_{\alpha\beta} = Z_B \cdot \sum_{ij} N_i^\alpha N_j^\beta \left(\frac{1}{R}\ \frac{1}{g^{h_1}}\ \gamma(h_1, R\cdot g) + \frac{1}{g^{h_2}}\ \Gamma(h_2, R\cdot g) \right), \tag{3.12}$$

where $h_1 = (a_i^\alpha + a_j^\beta + 3)$, $h_2 = (a_i^\alpha + a_j^\beta + 2)$ and $g = (b_i^\alpha + b_j^\beta)$. The coefficients $N_{i/j}^{\alpha/\beta}$, $a_{i/j}^{\alpha/\beta}$ and $b_{i/j}^{\alpha/\beta}$ need to be determined by comparing the expansions in (3.9) with the general expression of a hydrogen-like wave function in (2.13).

Equation (3.12) is quite easily evaluated, provided one has highly accurate incomplete gamma functions. The needed functions (3.11) are implemented in the subroutine as

$$\begin{aligned} \gamma(s,x) &= \Gamma(s) \cdot P(s,x) \\ \Gamma(s,x) &= \Gamma(s) \cdot \big[1 - P(s,x)\big]\ , \end{aligned} \tag{3.13}$$

[1] Using the polar angle θ of the coordinate system on center A in (3.10) is only possible, because we are dealing with s-states on center A. Only then are the states independent of the angles θ and ϕ and we can turn the z-axis of the coordinate system in fig. 3.3 in alignment with $\vec{r}_A$ (see fig. A.1). Then we can identify the angle between $\vec{r}_A$ and $\vec{R}$ with the polar angle θ of the spherical coordinates on center A. If the states were not s-states and we could not turn the z-axis, we would have to use a different angle θ' as the angle between the vectors $\vec{r}_A$ and $\vec{R}$ and the entire calculation would not be possible since the orthogonality relations would not be applicable. For further detail see APPENDIX A.

where the *normalized lower incomplete gamma function* $P(s,x)$ and the *gamma function* $\Gamma(s)$ for integer values of s were defined in (3.4) and (3.5) to be:

$$P(s,x) = \frac{1}{\Gamma(s)} \int_0^x dt\ t^{s-1}e^{-t}$$
$$\Gamma(s) = (s-1)! \quad , \quad s \in \mathbb{N} \tag{3.14}$$

This means that in fact only $\Gamma(s)$ and subsequently $P(s,x)$ are calculated and then used to determine $\gamma(s,x)$ and $\Gamma(s,x)$ according to (3.13). The implementations of both $P(s,x)$ and $\Gamma(s)$ were explained in section 3.1.4, where it was already mentioned that cancellation errors might occur if $P(s,x)$ is not accurate enough. As is shown in fig. 3.4, this is exactly what happens within the sum in (3.12).

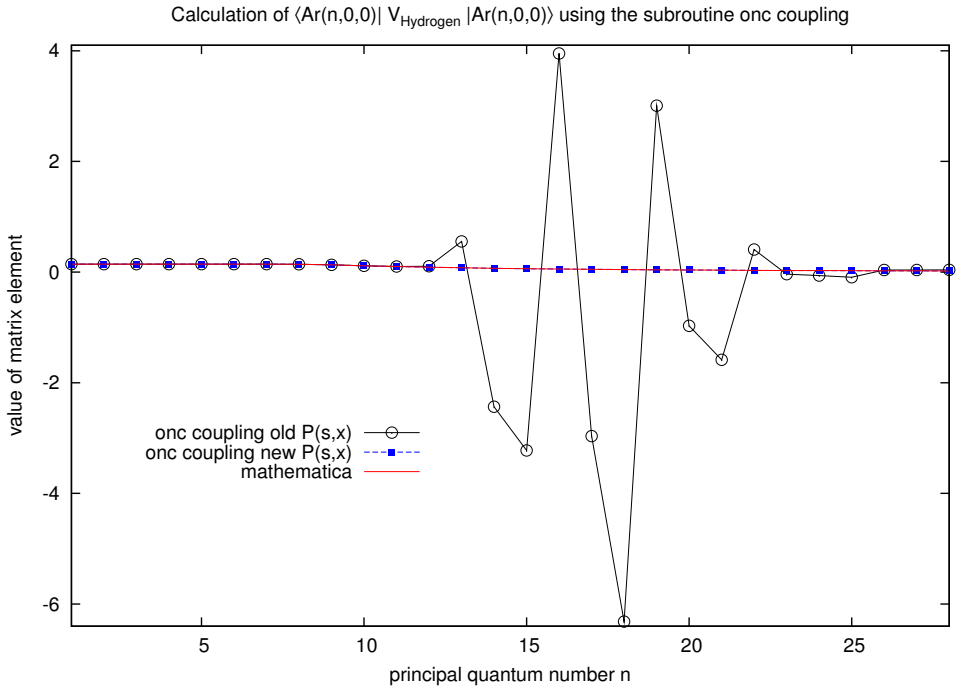


Figure 3.4: The effects of different implementations (see section 3.1.4) of the *normalized lower incomplete gamma function* $P(s,x)$ on the calculations in the subroutine onc_coupling. When a less accurate implementation is used, cancellation errors occur within the sum in (3.12) and lead to obvious numerical instabilities above $n = 12$. With the new implementation these instabilities vanish and the results calculated by the subroutine onc_coupling agree perfectly with reference values calculated by mathematica (mathematica implementation see section 3.2.2). The calculations were done for the arbitrary value $R \approx 7$.

3.2.2 Mathematica Implementation

In this section we briefly outline the mathematica calculation, for detailed mathematica codes refer to Appendix B. The values of the elements (3.8) are calculated by integrating in spherical coordinates (r,θ,ϕ) on center A. Thus we need to express the potential $V_B = \frac{Z_B}{|\vec{r}_B|}$ from center B in terms of (r,θ,ϕ). From very simple geometric considerations in fig. 3.3, we already learned

that $\vec{r}_B = \vec{R} - \vec{r}_A$. We can easily give $\vec{r}_A$ in terms of (r, θ, ϕ):

$$\vec{r}_A(r,\theta,\phi) = \begin{pmatrix} r\sin\theta\cos\phi \\ r\sin\theta\sin\phi \\ r\cos\theta \end{pmatrix} \tag{3.15}$$

We still need $\vec{R}$ as $\vec{R}(r, \theta, \phi)$. The angle γ between the z-axis on center A and the vector $\vec{R}$ is given by

$$\gamma = \arcsin\frac{|\vec{b}|}{|\vec{R}|} \,. \tag{3.16}$$

This angle γ is the polar angle θ in the coordinate system on center A. Since our collision only takes place in the x-z plain, we can set the azimuth angle $\phi = 0$ for the vectors $\vec{R}$ and $\vec{r}_A$.

$$\vec{R}(r,\theta,\phi) = \begin{pmatrix} R\sin\gamma \\ 0 \\ R\cos\gamma \end{pmatrix} \tag{3.17}$$

The potential V_B in terms of (r, θ, ϕ) is then given by

$$V_B(r,\theta,\phi) = \frac{Z_B}{|\vec{r}_B|} = \frac{Z_B}{|\vec{R} - \vec{r}_A|} = \frac{Z_B}{\left|\begin{pmatrix} R\sin\gamma \\ 0 \\ R\cos\gamma \end{pmatrix} - \begin{pmatrix} r\sin\theta \\ 0 \\ r\cos\theta \end{pmatrix}\right|} \tag{3.18}$$

We now express the states $|\psi\rangle_A$ in terms of (r, θ, ϕ), compare to (2.13)

$$\begin{aligned} |\psi\rangle_A &= |Z_A, n, l, m\rangle = R_{nl}(Z_A, r) \cdot Y_{lm}(\theta, \phi) \\ {}_A\langle\psi| &= \langle Z_A, n, l, m| = R_{nl}(Z_A, r) \cdot Y^*_{lm}(\theta, \phi) \end{aligned} \tag{3.19}$$

This finally yields the general expression of the integral $M_{\alpha\beta}$:

$$\begin{aligned} M_{\alpha\beta} = \int_0^\infty r^2 dr \int_0^\pi \sin\theta d\theta \int_0^{2\pi} d\phi \;\; & R_{n_\alpha l_\alpha}(Z_A, r) \cdot Y^*_{l_\alpha m_\alpha}(\theta, \phi) \;\times \\ & \times V_B(r, \theta, \phi) \cdot R_{n_\beta l_\beta}(Z_A, r) \cdot Y_{l_\beta m_\beta}(\theta, \phi) \end{aligned} \tag{3.20}$$

Equation (3.20) can then be solved by Mathematica. Check with Appendix B for Mathematica codes.

3.2.3 sic3ma Implementation

The code "sic3ma" implements all matrix elements using a method proposed by Shakeshaft [14] which was implemented in a more modern version by Wallerberger [15]. This is described in more detail in Wallerberger's Master Thesis (see [11], chapter 3).

The method uses the possibility to expand the involved state on center A/B in powers of $r_{A/B}$, in terms of the form $(\vec{r}_{A/B})^{\vec{l}}$ (this notation will be introduced later in (3.24)) and an exponential factor $\exp(\alpha_{A/B} \cdot r_{A/B})$. This expansion can be seen in (3.25). The potential $V(R, r_A, r_B)$ is also expanded in powers of $r_{A/B}$, terms of the form $(\vec{r}_{A/B})^{\vec{l}}$ and an exponential

phase $\exp[i\ (\vec{a}\vec{r}_A + \vec{b}\vec{r}_B)]$, which can be seen later on in (3.26). All occuring matrix elements can then be described by a general *Shakeshaft exchange integral* $I(n_1, \vec{l}_1, n_2, \vec{l}_2)$ (see [14]), which is of the form:

$$I(n_1, \vec{l}_1, n_2, \vec{l}_2) = \int d^3r_A\ r_A^{n_1-2}\ r_B^{n_2-2}\ (\vec{r}_A)^{\vec{l}_1}(\vec{r}_B)^{\vec{l}_2}\ \exp\left(i\vec{a}\vec{r}_A + i\vec{b}\vec{r}_B - cr_A - dr_B\right) \tag{3.21}$$

We now want to see, how (3.21) can be used to express a ONC matrix element. The basic form of a hydrogen-like wavefunction (2.13) is given by

$$|\phi\rangle = |Z, n, l, m\rangle = N_{nl}\ \rho^l\ e^{-\rho/2}\ L_{n-l-1}^{(2l+1)}\ Y_{lm}\ , \tag{3.22}$$

where N_{nl} is a normalization constant and $\rho = 2Zr/n$ is the reduced radius in atomic units. As was done in (3.9), the radial part of (3.22) can be expressed in terms of slater orbitals, while the spherical harmonics Y_{lm} may be written in Cartesian coordinates as

$$Y_{lm}(\vec{r}) = r^{-l} \sum_i C_i \cdot (\vec{r})^{\vec{l}_i} \quad \text{with} \quad l_{i,1} + l_{i,2} + l_{i,3} = l\ , \tag{3.23}$$

where we introduced *vector powers* defined as follows

$$(\vec{r})^{\vec{l}} = r_1^{l_1} r_2^{l_2} r_3^{l_3} = x^{l_1} y^{l_2} z^{l_3} \tag{3.24}$$

to shorten the notation. The wavefunction (3.22) then takes the following form (see [11], (A.1) on page 69):

$$\langle \vec{r}\ |\phi\rangle = |\alpha, n, \vec{l}\ \rangle = e^{-\alpha r}\ \sum_i \chi_i\ r^{n_i-2}(\vec{r})^{\vec{l}_i} \tag{3.25}$$

The potential is assumed to be radial-symmetric and is expanded as

$$V(r) = \sum_i a_i\ r^{\tilde{\rho}_i} \exp(-f_i r)\ . \tag{3.26}$$

Using (3.25) and (3.26), the one-center coupling matrix elements can be written as

$$\begin{aligned} {}_A\langle\phi_1|V_B|\phi_2\rangle_A = {}_A\langle\alpha, n, \vec{l}\ |V_B|\beta, m, \vec{k}\ \rangle_A &= \int d^3r_A \overbrace{e^{-\alpha r_A} \sum_p \chi_p^* r_A^{n_p-2}(\vec{r}_A)^{\vec{l}_p}}^{{}_A\langle\alpha,n,\vec{l}|} \times \\ \times \underbrace{\sum_i a_i\ r_B^{\tilde{\rho}_i} \exp(-f_i r_B)}_{V(r_B)} &\cdot \underbrace{e^{-\beta r_A} \sum_q \xi_q\ r_A^{m_q-2}(\vec{r}_A)^{\vec{k}_q}}_{|\beta,m,\vec{k}\rangle_A} \\ = \sum_{p,q,i} \chi_p^* \xi_q a_i \int d^3r_A\ r_A^{n_p+m_q-4}\ r_B^{\tilde{\rho}_i} &\cdot (\vec{r}_A)^{\vec{l}_p+\vec{k}_q}\ \exp\left(-(\alpha+\beta)r_A - f_i r_B\right)\ . \end{aligned} \tag{3.27}$$

By comparing the expression (3.27) for a one-center coupling matrix element to the form of a general exchange integral in (3.21), we can see that the one-center coupling matrix element can be written as

$${}_A\langle\phi_1|V_B|\phi_2\rangle_A = \sum_{p,q,i} \chi_p^* \xi_q a_i \cdot I(n_p + m_q - 2, \vec{l}_p + \vec{k}_q, \tilde{\rho}_i + 2, \vec{0})\ , \tag{3.28}$$

and also that $c = \alpha + \beta$, $d = f_i$ as well as $\vec{a} = \vec{b} = 0$. Shakeshaft has shown, that the integral I in (3.28) can be simplified to a one dimensional integral. Using this simplification in (3.28) yields the expression

$$ {}_A\langle\phi_1|V_B|\phi_2\rangle_A = \sum_i a_i R^{\rho_i} R_1^{\phi_i} R_3^{\psi_i} \int_0^1 dy \sqrt{y}^{\sigma_i} exp(-cR\sqrt{y}) \ , \tag{3.29} $$

where R denotes the internuclear distance, $R_1 = b$ gives the impact parameter and $R_3 = r_3$ gives the point on the trajectory. Making a variable substitution $u = cR\sqrt{y}$ finally gives the following sum for the ONC matrix element (for more detail refer to [11], section 3.1.3):

$$ {}_A\langle\phi_1|\ V_B\ |\phi_2\rangle_A = \sum_i \underbrace{\frac{2a_i\Gamma(\sigma_i+2)}{c^{\sigma_i+2}} R^{\rho_i-(\sigma_i+2)} R_1^{\phi_i} R_3^{\psi_i}}_{=\ \Omega_i} \cdot P(\sigma_i+2, cR) \tag{3.30} $$

The symbolic structures of ONC matrix elements are implemented in the code sic3ma as a Fortran type called "coulombint". This type includes the matrix elements (3.29) as a polynomial in 4 variables, namely $\sqrt{y}, R, R_1, R_3$. In step 1 of the implementation (see section 2.2), such a "coulombint" is produced and stored into files for every matrix element. In the subsequent evaluation of the matrix elements (step 2), the values of the variables are inserted and the coefficient Ω_i of the resulting polynomial (which has no more variables then) is multiplied with the normalized lower incomplete gamma function $P(\sigma_i+2, cR) = P(s, x)$ for every term i in the sum (3.30).

It is here, where cancellation errors may occur and numerical instabilities similar to the ones seen in fig. 3.4 may appear. Since the coefficients Ω_i include the gamma function $\Gamma(\sigma_i+2)$, they become very large numbers up to order 10^{12}, while the value of the normalized lower incomplete gamma function $P(s, x)$ always stays between 0 and 1, as the name *normalized* suggests. The final result ${}_A\langle\phi_1|\ V_B\ |\phi_2\rangle_A$ is of order 10^{-2}, which means that the accuracy of $P(s, x)$ has to be at least to the 14^{th} digit in order for the result to be calculated correctly (compare section 3.1.4). This was not the case with the old implementation of $P(s, x)$ (compare table 3.1) and numerical instabilities can in fact be observed in fig. 3.5.

3.2.4 Precision Scenarios

Besides the accuracy of the function $P(s, x)$, the precision of the involved coefficients is of major importance and has a huge impact on the numerical stability of the calculations. It is therefore necessary to determine, which precision is required for each step of the calculations. The critical steps are (see more detail in section 2.2) creating the symbolic structures (step A), writing them into files (step B) and then reading the symbolic structures and evaluating the matrix elements (step C). To determine which precision is required for each of these steps, four different scenarios where calculated:

- **scenario 8 - 8 - 8:** This is the old version of the code. Everything is calculated using only double precision numbers. This scenario is only used as a reference to the old results.
- **scenario 16 - 8 - 8:** The symbolic structures are created in quad precision (16 bytes number), then stored in files as double precision (8 bytes) numbers and also read in and used to evaluate the matrix elements in double precision.

- **scenario 16 - 8 - 16:** The creation of the symbolic structures is done in quad precision, but they are stored as double precision numbers. Then there is a cast from double to quad precision after reading the structures and the evaluation is again conducted in quad precision.

- **scenario 16 - 16 - 16:** Every single step A to C is done in quad precision.

An overview over the calculated scenarios is given in the following table:

Precision Scenarios

	8 - 8 - 8	**16 - 8 - 8**	**16 - 8 - 16**	**16 - 16 - 16**
step A (producing)	double precision (8 bytes)	quad precision (16 bytes)	quad precision (16 bytes)	quad precision (16 bytes)
step B (writing)	double precision (8 bytes)	double precision (8 bytes)	double precision (8 bytes)	quad precision (16 bytes)
step C (evaluating)	double precision (8 bytes)	double precision (8 bytes)	quad precision (16 bytes)	quad precision (16 bytes)

The results of the calculation for all four scenarios compared to reference values calculated by mathematica and onc_coupling can be seen in fig. 3.5. The ONC matrix elements $\langle Ar(n,0,0)|V_H|Ar(n,0,0)\rangle$ were calculated for the same internuclear distance R as in section 3.1.2, namely $R \approx 7$. Since scenario 8-8-8 is basically the old version of the code, where everything is calculated using double precision numbers, it is obvious that this scenario only gives reasonable results up to $n = 12$ (this limit is indicated by the solid vertical line in fig. 3.5). The same limit could already be observed in fig. 3.1.

However, changing the precision from double to quad precision does improve the calculations considerably. The calculated values are still not correct and there are obviously numerical instabilities above $n = 12$, as the values oscillate around the reference values. Still, in contrast to scenario 8-8-8 the results at least remain within a few orders of magnitude around the reference values even far beyond $n = 12$. Scenario 16-8-8 yields roughly correct values up until $n = 24$, while scenarios 16-8-16 and 16-16-16 reach $n = 25$, before the values grow up to order $\approx 10^6$ at $n = 30$ in all three scenarios. This approximate limit at $n = 25$ (after which the values grow enormously large) is indicated by the dotted vertical line in fig. 3.5.

Seeing as there are no considerable differences between all three scenarios involving quad precision numbers, one might assume that the calculation of ONC matrix elements only requires step A to be conducted in quad precision, while all other steps in the calculation may be done using double precision numbers. When taking a closer look though, one can see that scenarios 16-8-16 and 16-16-16 show a slightly better performance, since their limit is at $n = 25$, while scenario 16-8-8 only manages the limit $n = 24$. It must also be considered that the calculations shown in fig. 3.5 are conducted for $R \approx 7$ and there is a chance for problems to occur at higher values of R. In addition to that, we will see in section 3.3.3, that two-center matrix elements are only calculated correctly when using scenario 16-16-16. For these reasons and simply to be on the safe side, it was chosen to do all calculations using scenario 16-16-16.

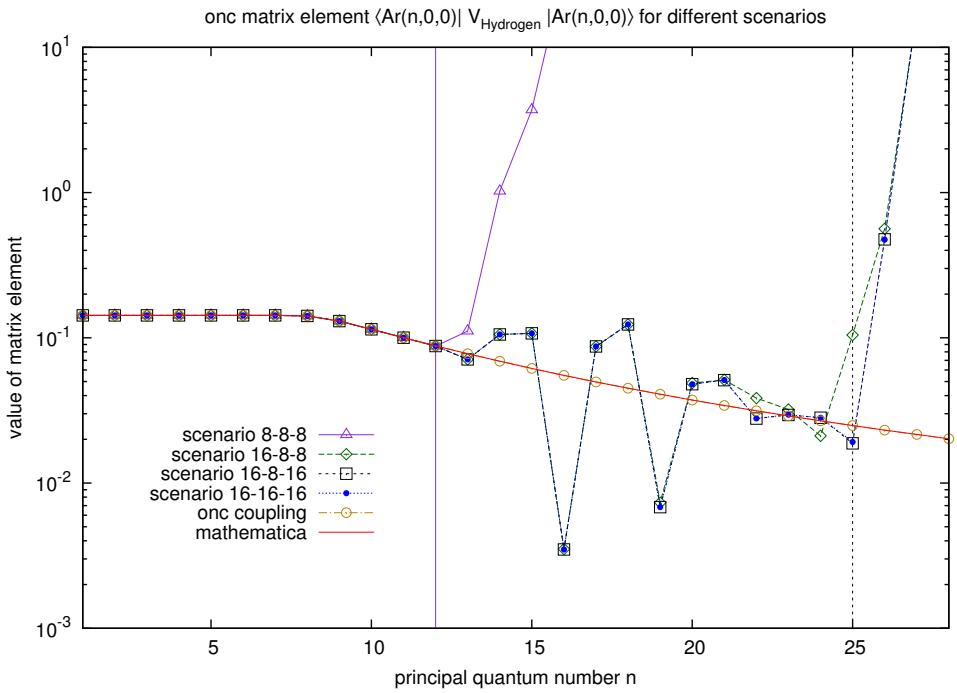
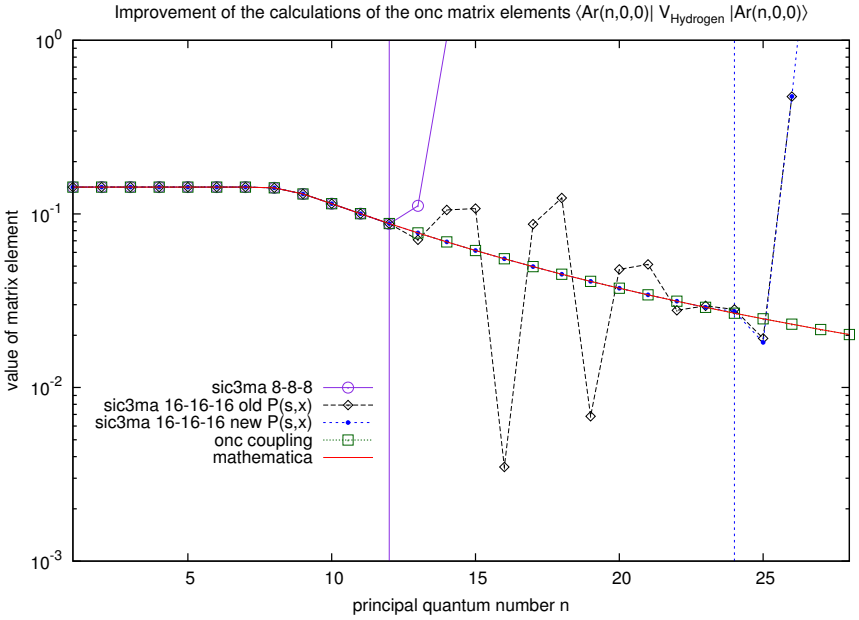
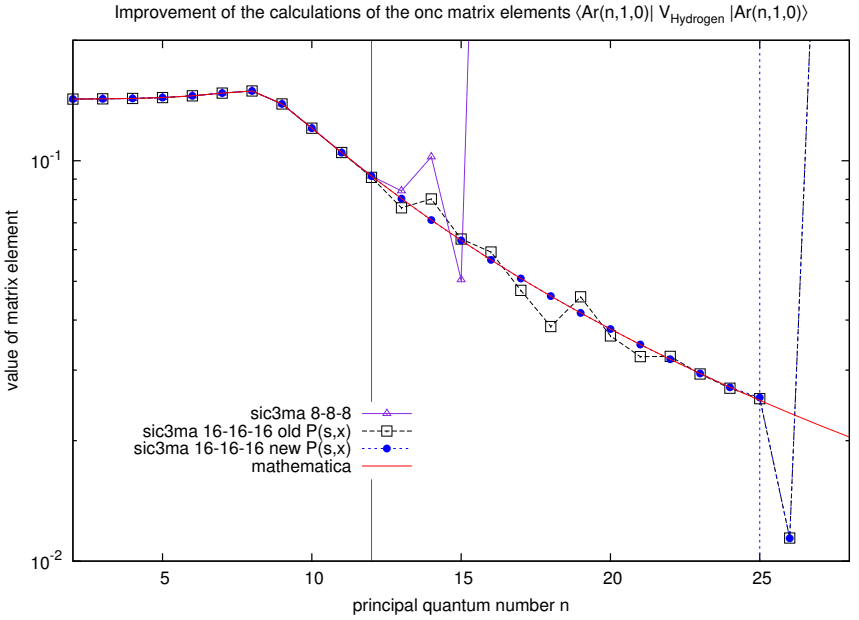


Figure 3.5: Fig. 3.5 shows the investigation of the four different scenarios compared to reference values (calculated by mathematica and independently also by onc_coupling). The ONC matrix elements $\langle Ar(n,0,0)|V_H|Ar(n,0,0)\rangle$ were calculated for the arbitrary value $R \approx 7$. Scenario 8-8-8 only gives reasonable results up to $n = 12$ (indicated by the solid vertical line). All other scenarios remain within a few orders of magnitude around the mathematica reference values up until $n = 24$ in case of scenario 16-8-8 and $n = 25$ for scenarios 16-8-16 and 16-16-16. This limit is indicated by the dotted vertical line at $n = 25$. However, there are still considerable numerical instabilities above $n = 12$ as the values oscillate around the reference values.



(a)



(b)

Figure 3.6: Fig. 3.6(a) shows the final improvements of the calculation of the ONC matrix elements $\langle Ar(n,0,0)|V_H|Ar(n,0,0)\rangle$. The internuclear distance was chosen to be $R \approx 7$. It can be seen that by changing to quad precision and using a new implementation of the normalized lower incomplete gamma function $P(s,x)$, the limit of sic3ma could be pushed from previously $n = 12$ to $n = 24$, making it possible to include a lot more states than before. As is shown in fig. 3.6(b), similar improvements could be achieved for the elements $\langle Ar(n,1,0)|V_H|Ar(n,1,0)\rangle$ involving p-states on the Argon center. Making the same changes in the code, the previous limit at $n = 12$ could even be pushed as far as $n = 25$. For elements involving p-states there are only mathematica reference values available, since onc_coupling only works for s-states.

3.2.5 Achieved Improvements

Now that it is determined to use scenario 16-16-16, the calculations can be further improved and refined. As seen in fig. 3.5, the values of the calculated ONC matrix elements show considerable instabilities above $n = 12$ even when using quad precision numbers during all involved steps (scenario 16-16-16). These instabilities are produced by cancellation errors in the sum (3.30) during the evaluation of the matrix elements due to an insufficiently accurate normalized lower incomplete gamma function $P(s, x)$ (compare to section 3.1.4 and section 3.2.3). A very similar effect was already observed for the calculations of the subroutine onc_coupling in fig. 3.4.

The improvements of the results when using the new implementation of $P(s, x)$ are shown in fig. 3.6. Using the same internuclear distance $R \approx 7$ as before, the ONC matrix elements involving s-states on the Argon center $\langle Ar(n, 0, 0)|V_H|Ar(n, 0, 0)\rangle$ were calculated in fig. 3.6(a), while fig. 3.6(b) shows the calculation of elements involving p-states on the Argon center $\langle Ar(n, 1, 0)|V_H|Ar(n, 1, 0)\rangle$. In both cases, numerical instabilities vanish when using the new implementation of $P(s, x)$ and the limits could thus be pushed from $n = 12$ (full vertical lines in the respective figures) to $n = 24$ for s-states and $n = 25$ for p-states. Fig. 3.6(b) also shows that the numerical instabilities when using the old $P(s, x)$ are in general smaller for matrix elements involving p-states.

3.3 Two-Center Matrix Elements

In this section, we will investigate the calculation of two-center overlap and TWC matrix elements. Both elements involve an integral over states defined on two different centers. Fig. 3.7 shows the geometry for a two-center system.

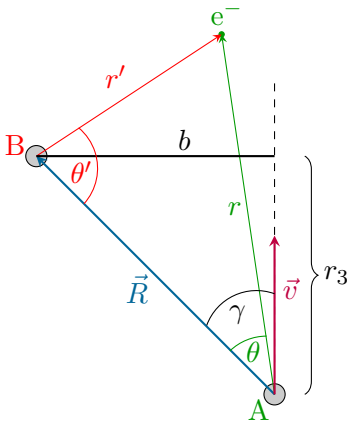


Figure 3.7: The geometry for TWC matrix elements. The z-axis of center A was turned to be in alignment with $\vec{R}$, while the z-axis of center B was turned to be antiparallel to $\vec{R}$. The coordinate system on center A is then determined by (r, θ), while the coordinate system on center B is determined by (r', θ'). It is then possible to use the shown general triangle A-B-e$^-$ to express the coordinates (r', θ') on center B via the coordinates (r, θ) from center A, i.e. to obtain the functions $r' = r'(r, \theta)$ and $\theta' = \theta'(r, \theta)$. The collisional geometry is also shown, where b is the impact parameter, $\vec{v}$ is the impact velocity and r_3 defines the point on the trajectory.

Obviously, one has to deal with coordinates (r, θ, ϕ) on center A and coordinates (r', θ', ϕ') on center B. The TWC matrix elements have the general form

$$\begin{aligned} M_{\alpha\beta} &= {}_A\langle \phi_\alpha(r,\theta,\phi)|\ V_A(r,\theta,\phi)\ |\phi_\beta(r',\theta',\phi')\rangle_B \\ &= {}_A\langle Z_A, n_\alpha, l_\alpha, m_\alpha|\ \frac{Z_A}{|\vec{r}_A|}\ |Z_B, n_\beta, l_\beta, m_\beta\rangle_B\ , \end{aligned} \tag{3.31}$$

while the two-center overlap matrix elements are given by

$$\begin{aligned} S_{\alpha\beta} &= {}_A\langle \phi_\alpha(r,\theta,\phi)|\phi_\beta(r',\theta',\phi')\rangle_B \\ &= {}_A\langle Z_A, n_\alpha, l_\alpha, m_\alpha|Z_B, n_\beta, l_\beta, m_\beta\rangle_B \end{aligned} \tag{3.32}$$

Centers A and B may be substituted with either the target or the projectile to obtain the wanted matrix element.

Section 3.3.1 will describe how reference values can be obtained from Mathematica, section 3.3.2 explains how the matrix elements are implemented in the code sic3ma, before section 3.3.3 finally explores which precision scenario should be used during the calculations and also shows the achieved improvements for the calculation of two-center matrix elements.

3.3.1 Mathematica Implementation

To use Mathematica for evaluating the two-center matrix elements, we want to solve the integral on center A in spherical coordinates (r, θ, ϕ). To this end, we need to find the relations between the coordinate system on center B and the coordinate system on center A. For reasons of simplicity, we restrict these considerations to matrix elements, where only s-states are involved. In this case, everything is independent of the angle ϕ (which we can set to zero) and we may turn the z-axis of center A in line with $\vec{R}$, while the z-axis of center B is turned to be antiparallel to $\vec{R}$. Taking a look at the triangle in fig. 3.7, we can get $\theta' = \theta'(r, \theta)$ and $r' = r'(r, \theta)$ by using the cosine formula

$$r'^2 = R^2 + r^2 - 2Rr\cos\theta \;\rightarrow\; r' = \sqrt{R^2 + r^2 - 2Rr\cos\theta} \tag{3.33}$$

and the sine law

$$\frac{r'}{\sin\theta} = \frac{r}{\sin\theta'} \;\rightarrow\; \sin\theta' = \frac{r\ \sin\theta}{r'} = \frac{r\ \sin\theta}{\sqrt{R^2 + r^2 - 2Rr\cos\theta}}\ . \tag{3.34}$$

This yields the relations between coordinates on center A and coordinates on center B:

$$\begin{aligned} r'(r,\theta) &= \sqrt{R^2 + r^2 - 2Rr\cos\theta} \\ \theta'(r,\theta) &= \arcsin\left[\frac{r\ \sin\theta}{\sqrt{R^2 + r^2 - 2Rr\cos\theta}}\right] \end{aligned} \tag{3.35}$$

Additionally, an expression for the ETF, which will be applied to projectile state A according to (2.9), needs to be implemented. It will take the form

$$ETF(\vec{r}, \vec{v}) = \exp(i\vec{v}\vec{r})\ . \tag{3.36}$$

The angle γ between $\vec{R}$ and $\vec{v}$ is given by

$$\gamma = \arcsin\frac{b}{|\vec{R}|} \tag{3.37}$$

and can be used to give $\vec{v}$ in terms of coordinates on center A (we can set $\phi = 0$):

$$\vec{v}(r, \theta, \phi) = \begin{pmatrix} v\sin\gamma \\ 0 \\ v\cos\gamma \end{pmatrix}\ . \tag{3.38}$$

The vector multiplication between $\vec{v}$ and $\vec{r}$ thus gives

$$\vec{v}\cdot\vec{r} = \begin{pmatrix} v\sin\gamma \\ 0 \\ v\cos\gamma \end{pmatrix} \cdot \begin{pmatrix} r\sin\theta \\ 0 \\ r\cos\theta \end{pmatrix} = rv(\sin\theta\sin\gamma + \cos\theta\cos\gamma) \tag{3.39}$$

From this follows the expression for the ETF

$$ETF(r, \theta) = \exp\left[irv(\sin\theta\sin\gamma + \cos\theta\cos\gamma)\right] \tag{3.40}$$

The matrix elements $M_{\alpha\beta}$ and $S_{\alpha\beta}$ are then obviously given in an analogous way to the calculation in section 3.2.2 as

$$\begin{aligned} M_{\alpha\beta} = \int_0^\infty r^2 dr \int_0^\pi \sin\theta d\theta \int_0^{2\pi} d\phi\ R_{n_\alpha l_\alpha}(Z_A, r)\cdot Y^*_{l_\alpha m_\alpha}(\theta, \phi = 0)\ \times \\ \times\ ETF(r,\theta)\cdot V_A(r,\theta,\phi=0)\cdot R_{n_\beta l_\beta}(Z_B, r')\cdot Y_{l_\beta m_\beta}(\theta', \phi' = 0) \end{aligned} \tag{3.41}$$

and

$$S_{\alpha\beta} = \int_0^\infty r^2 dr \int_0^\pi \sin\theta d\theta \int_0^{2\pi} d\phi \;\; R_{n_\alpha l_\alpha}(Z_A, r) \cdot Y^*_{l_\alpha m_\alpha}(\theta, \phi = 0) \times \\ \times \;\; ETF(r,\theta) \cdot R_{n_\beta l_\beta}(Z_B, r') \cdot Y_{l_\beta m_\beta}(\theta', \phi' = 0) \;, \tag{3.42}$$

where in both equations the primed coordinates have to be substituted by unprimed coordinates according to (3.35). Since our considerations are independent of the azimuthal angles, we can set $\phi = \phi' = 0$. Equations (3.41) and (3.42) can then again be solved by Mathematica. Check with Appendix B for Mathematica codes.

3.3.2 sic3ma Implementation

The two-center matrix elements are implemented using the general Shakeshaft exchange integral (3.21), which is of the form:

$$I(n_1, \vec{l}_1, n_2, \vec{l}_2) = \int d^3 r_A \; r_A^{n_1 - 2} \; r_B^{n_2 - 2} \; (\vec{r}_A)^{\vec{l}_1} (\vec{r}_B)^{\vec{l}_2} \;\; \exp\left(i\vec{a}\vec{r}_A + i\vec{b}\vec{r}_B - cr_A - dr_B\right) \tag{3.43}$$

According to Shakeshaft [14] the integral above can be rewritten into a one dimensional integral (see also [11], section 3.1.2)

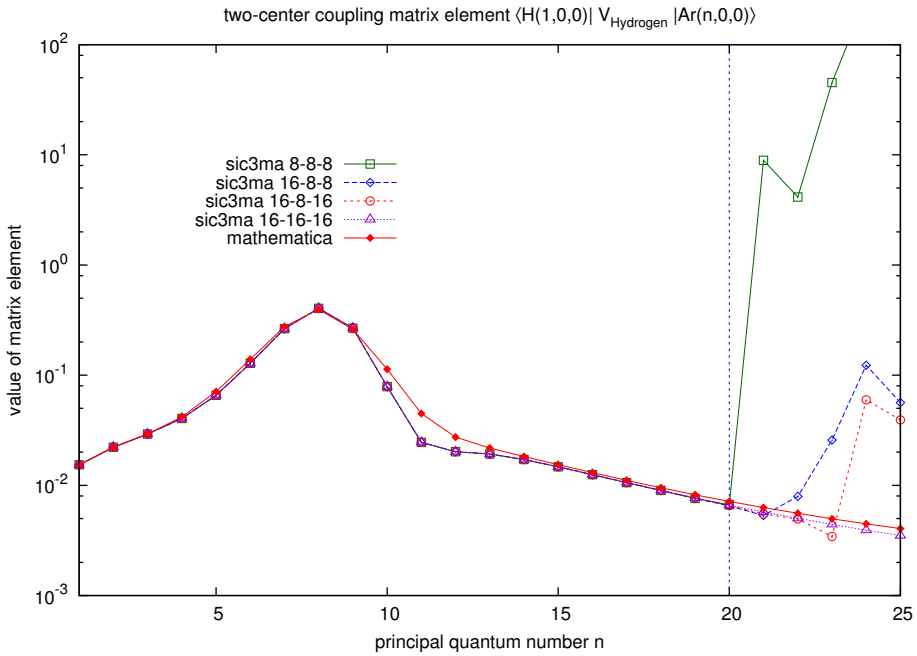
$$I = 2\pi(-i)^{l_1+l_2} (\nabla_{\vec{a}})^{\vec{l}_1} (\nabla_{\vec{b}})^{\vec{l}_2} \left(-\frac{\partial}{\partial c}\right)^{n_1 - 1} \left(-\frac{\partial}{\partial d}\right)^{n_2 - 1} \int_0^1 dy \frac{\exp(i\vec{B}\vec{R} - AR)}{A} \;, \tag{3.44}$$

where $\vec{R} = \vec{r}_A - \vec{r}_B$ is the internuclear distance. The newly introduced quantities A and $\vec{B}$ are given by $A^2 := y(1-y)|\vec{a} + \vec{b}|^2 + yc^2 + (1-y)d^2$ and $\vec{B} = y\vec{a} - (1-y)\vec{b}$. For our given collisional system we get $\vec{a} = v \cdot \vec{e}_z$ and $\vec{b} = 0$.

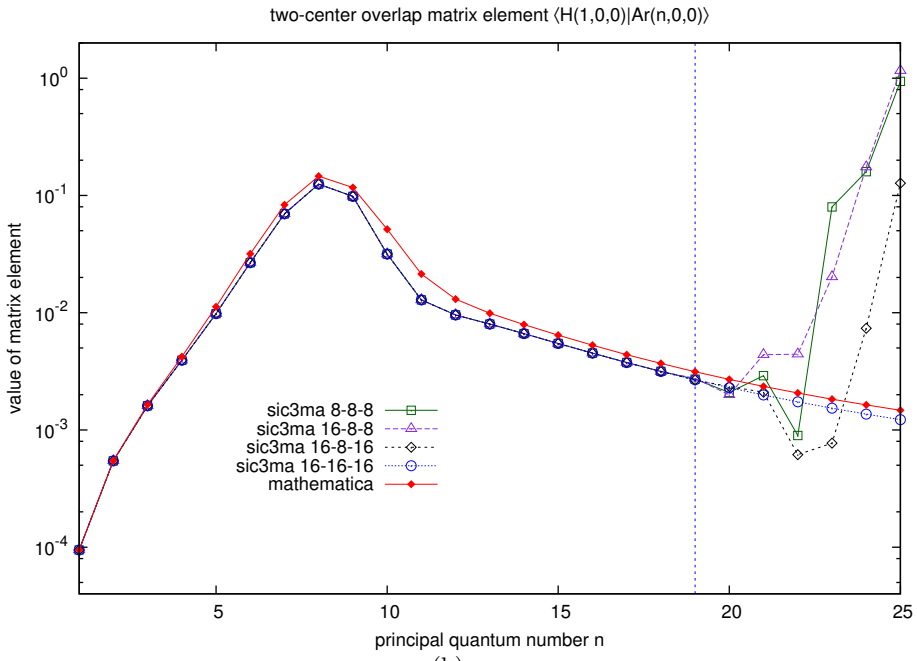
The two-center matrix elements (3.44) are stored as a symbolic structure in 7 variables during step 1 (see section 2.2.1) of the computation and are subsequently evaluated during step 2 (see section 2.2.2). For further details refer to [11], section 3.1.

3.3.3 Precision Scenarios

Using the precision scenarios described in 3.2.4, the TWC matrix elements $\langle H(1,0,0)|V_H|Ar(n,0,0)\rangle$ and the two-center overlap matrix elements $\langle H(1,0,0)|Ar(n,0,0)\rangle$ were calculated using states having main quantum number $1 \leq n \leq 25$. In both cases numerical instabilities leading to a rapid increase in the values around $n = 20$ could be overcome when using precision scenario 16-16-16. There are no more numerical inconsistencies as far as $n = 25$. There is some deviation from reference values calculated by Mathematica, but the differences are reasonably small. It is clear that the precision scenario 16-16-16 is needed to calculate two-center matrix elements. Even if the accuracy limit for scenario 8-8-8 is at or above $n = 20$ for the specific chosen values of R and v, it is not safe to assume, that this will be the case for all values of R and v. It should thus be calculated using scenario 16-16-16 to avoid any potential problems.



(a)



(b)

Figure 3.8: Fig. 3.8(a) shows the various precision scenarios in calculating the TWC matrix elements $\langle H(1,0,0)|V_H|Ar(n,0,0)\rangle$, showing that the scenario 16-16-16 clearly produces the best results. The internuclear distance was again chosen to be $R \approx 7$ and the velocity was $v = 0.5$. In fig. 3.8(b) similar improvements could be achieved for the two-center overlap matrix elements $\langle H(1,0,0)|Ar(n,0,0)\rangle$. Both matrix elements fit reasonably close to reference values calculated by mathematica.

3.4 Computational considerations

This section is dedicated to hardware and performance requirements which need to be considered when executing the various steps of sic3ma as outlined in 2.2. Hardware requirements (i.e. memory needs on the computing nodes) will de discussed in section 3.4.1, where we will show that a more elaborate distribution of all symbolic structures onto the available computing nodes is necessary to have all symbolic structures available in the memory. In addition to that, section 3.4.2 will discuss performance considerations and explain that, in order to avoid or at least reduce idle time, the distribution also has to be optimized to factor in evaluation time needed at each rank.

3.4.1 Memory Needs per Node

It has already been discussed at the beginning of section 3.1 that a lot of basis states need to be included to accurately describe a collisional system like $Ar^{18+}+H$. We now assume that we want to describe the Hydrogen projectile with bound states up to $n = 3$ (to account for excitation) and some unbound pseudo states (modeling ionization into the continuum), giving in total $nproj = 73$ states. The Argon ion target should be modeled with eigenstates for $7 \leq n \leq 20$, yielding $ntarg = 1484$ states. This gives a system size of $nstat = 1557$, which means that there will be $nstat^2 = 2\ 424\ 249$ single rbM-files and $nproj \cdot ntarg = 108\ 332$ single rbS-files. The total size of all rbc-files is roughly 850 GB, which means that enough computing nodes have to be used in order for all rbc-files to fit onto the memory at once during the calculations in step 2 (see 2.2.2) and step 3 (see 2.2.3). The Vienna Scientific Cluster-2 (VSC-2) has 16 processors per computing node with a total shared memory of 32 GB RAM per node. Besides the rbc-files there are several other things that need to fit onto one node during step 2 and 3 of the computation:

1. **Results of the Evaluation in step 2** ≈ 1.6 GB
   The results of the evaluation of the matrix elements have to be kept in memory until they are broadcasted during the MPI broadcast (see 2.2.3). Each rbc-file has either 1557 elements (if there is only a column of $\hat{M}$) or $1557 + 73 = 1630$ elements (if there is a column of $\hat{M}$ and a column of $\hat{S}$) stored in it. To be safe, we want to calculate with 1630. Each rank will evaluate all its elements for all given values of (b,v). In accordance with our estimations in 2.2.3, we choose $nb = 24$ impact parameters and $nv = 12$ impact velocities. This gives a factor of $nv \cdot nb = 12 \cdot 24 \approx 300$. The evaluated matrix elements are complex*16 numbers taking up 16 bytes. Finally, the amount of rbc files per column has to be estimated. This depends of course on the distribution, but a rough estimate would be a maximum of 200 rbc-files on one node. This yields a generous estimate of $1630 \cdot 300 \cdot 200 \cdot 16$ bytes ≈ 1.6 GB for the result arrays.

2. **Matrices** $\hat{M}$**,** $\hat{S}$ **and** $\hat{M}_{\text{eff}}$ **during step 3** ≈ 4 GB
   During step 3 each rank will first gradually fill one entire matrix $\hat{M}$ and receive one quadrant of the matrix $\hat{S}$ during the MPI broadcast. After the broadcast is completed, each rank will built one full matrix $\hat{S}$, invert it and then calculate one matrix $\hat{M}_{\text{eff}}$. All three matrices $\hat{M}$, $\hat{S}$ and $\hat{M}_{\text{eff}}$ are of size $nstat^2 = 1557^2$ and each matrix element is a complex*16 number. This gives ≈ 40 MB per matrix. Using the evaluated matrices $\hat{M}$ and $\hat{S}$ at a point $-r_3$ and the time symmetry relation (2.29), each rank can directly calculate $\hat{M}$ and $\hat{S}$ at the point $+r_3$. Subsequently $\hat{M}_{\text{eff}}$ at $+r_3$ can also be calculated and thus three

more matrices need to be stored. For the entire node with 16 ranks and 6.25 matrices per rank this gives a needed memory of $16 \cdot 6.25 \cdot 40$ MB ≈ 4 GB.

We have now estimated, that approximately 6 GB per node are needed elsewhere. This leaves 26 GB free for the rbc-files. To be on the safe side, the limit for the symbolic structures should be 25 GB, leaving some free memory available. With a total memory need of 850 GB, it can now be estimated that 34 nodes will be needed to accommodate all rbc-files during one calculation.

In the original version of the code the rbc-files were simply distributed evenly among all available ranks. However, this did not account for the fact that the sizes of the rbc-files vary considerably, from a few MB to as much as 8 GB for the largest files. If a simple even distribution were to be applied to the system $Ar^{18+} + H$, node number 29 would have to store almost 100 GB. This is of course not possible, since the maximum available memory per node is 32 GB.

Thus a new and optimized distribution had to be found, driven by memory needs per node. In this new distribution, a shell-script first checks the sizes of all rbc-files. It then starts assigning the rbc-files to the 16 ranks within one node, beginning with a few large files before filling up the node with smaller files. As soon as the limit of 25 GB is reached, the next node is gradually filled up. Fig. 3.9 clearly shows the difference in memory needs per node between the old and the new distribution. Both distributions accommodate the same total amount of data, but the optimized distribution equally fills all nodes with roughly 25 GB, making it possible to store all 850 GB.

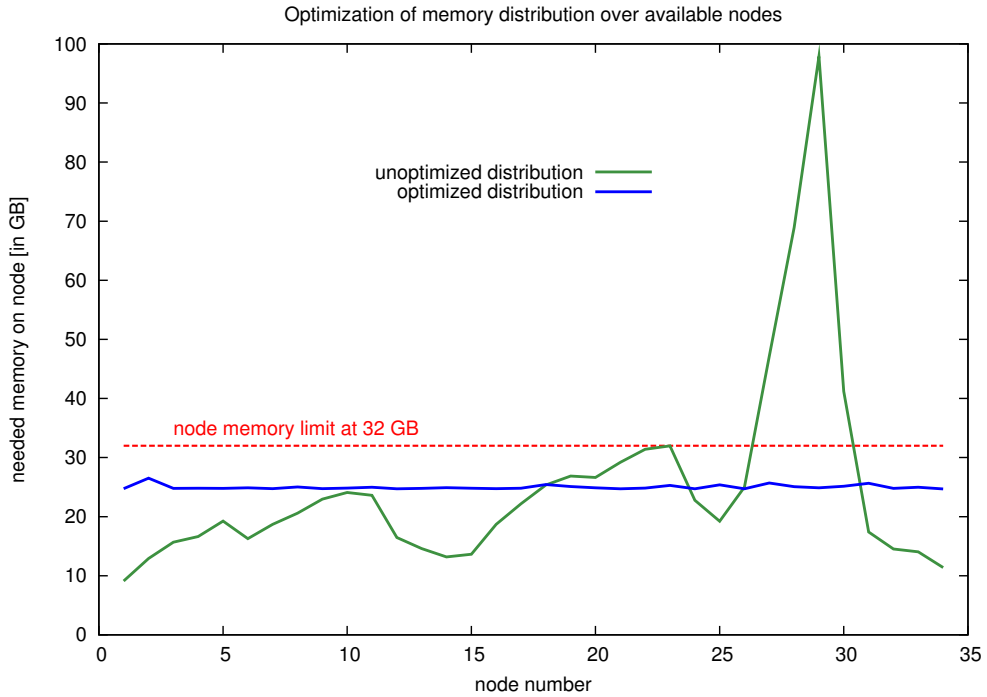


Figure 3.9: This figure shows the difference between an even distribution (green line) and an optimized distribution (blue line) of rbc-files. An even distribution over all available nodes results in a very unequal distribution of memory, making it impossible to accommodate the entire amount of data. The new and optimized distribution stores the same total amount of 850 GB, but spreads the data more evenly over all available nodes. This way, the maximum memory of 32 GB per node is not exceeded.

In order to achieve an equal distribution of memory, the number of rbc-files per node will most likely be different on each node, depending on whether a certain rbc-file is large in size or very small. This can be seen in fig. 3.10, where node 1 is assigned over 100 small rbc-files, while node 2 obviously only holds a few very large rbc-files.

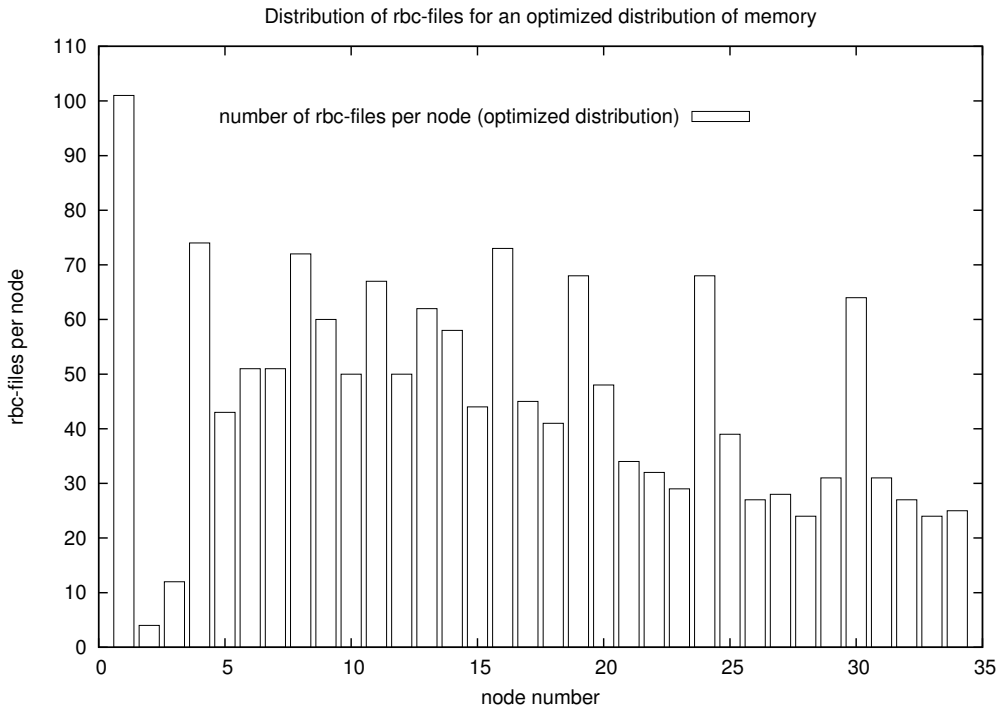


Figure 3.10: Given the fact, that the sizes of the rbc-files vary greatly, the number of rbc-files per node has to be very unevenly distributed in order to achieve an almost even distribution of memory. This is clearly shown in fig 3.10, where node number 1 holds over a hundred files, while node number 2 only accommodates a few large files. Nonetheless, both nodes roughly store 25 GB of data, as can be seen by looking at fig. 3.9.

3.4.2 Evaluation Time per Rank

In addition to hardware requirements which need to be met during the calculation, one should also consider performance aspects. One such aspect is how much time one rank needs for the evaluation of its assigned matrix elements in step 2 (see 2.2.2). Due to the conception of the code all ranks must be finished with the evaluation in step 2 before the MPI broadcast scheme can start in step 3 (compare 2.2.3). This can lead to considerable idle times when those ranks already finished with the evaluation have to wait for the others to finish.

To fit all rbc-files (as discussed before in section 3.4.1) and also to reduce evaluation time, it would be beneficial to use as many parallel cores as possible, since then every core has to evaluate less elements. However, during the building of the matrices $\hat{M}$ and $\hat{S}$ and the subsequent calculation of the effective matrix $\hat{M}_{\text{eff}}$ only $nv \cdot nb$ ranks will be used. The rest will be idle. It is obvious that there will always be a trade-off between these two aspects.

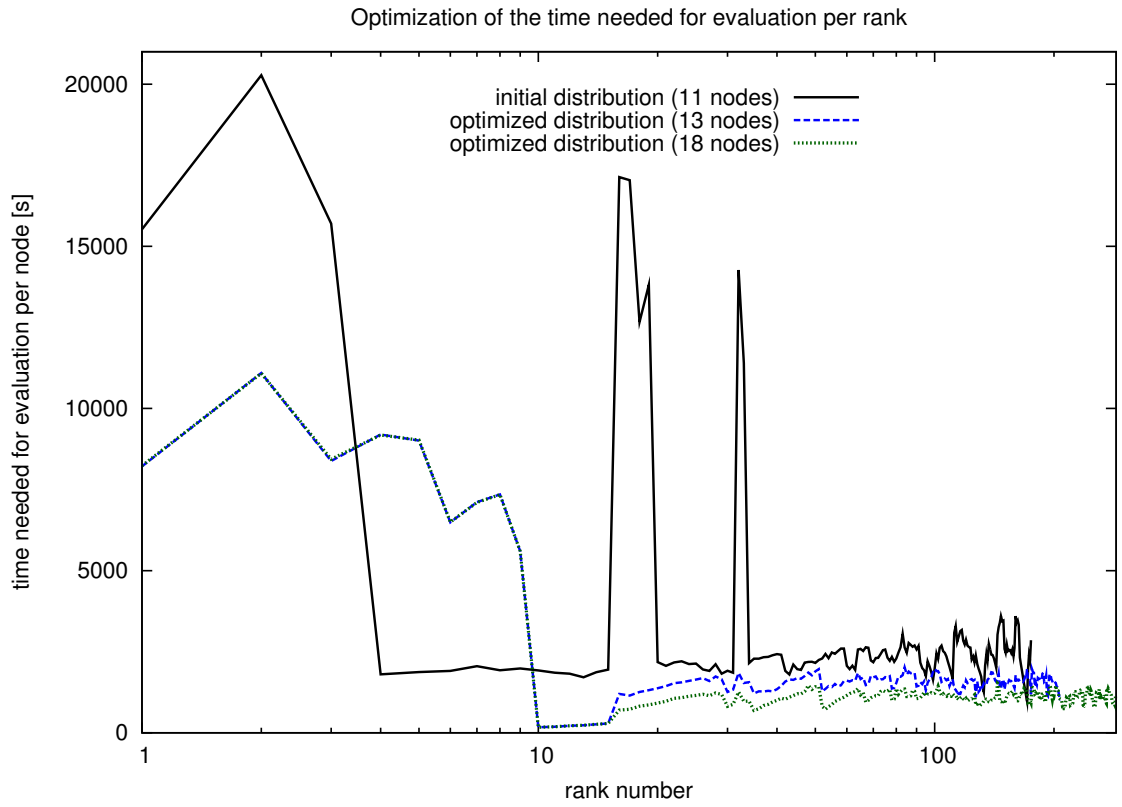


Figure 3.11: Using the initial distribution of rbc-files the evaluation time for rank 2 is as high as 20 000 seconds. This can be reduced to roughly 10 000 seconds when limiting the amount of files to one single file for the first node (rank 1 to 16). But even with this limitation and some reorganization of the rbc-files, the evaluation times are still much higher for the ranks on node 1 comparing to all other ranks.

When analyzing the evaluation time needed per rank it becomes clear that certain rbc-files need much, much longer than others and thus certain ranks take a very long time to finish with the evaluation while all other ranks are idle. This evaluation time correlates roughly with the size of the rbc-file, which seems obvious.

In fig. 3.11, the evaluation time for each rank is plotted. In the initial distribution of rbc-files on the ranks, it can be seen that rank 2 needs almost 20 000 seconds to evaluate, while those around rank 100 roughly need one order of magnitude less time. Once the ranks taking so much time to evaluate were identified, those ranks were only assigned one single rbc-file and the rest of the files were distributed over more parallel ranks. This way, the maximum evaluation time could be reduced, but even when evaluating only one single rbc-file, ranks 1 to 9 still need much more time than all the other ranks.

To really optimize this evaluation process, it will be necessary to reconsider the use of a column-wise distribution and maybe apply a distribution which is more suited to ease this problem concerning the evaluation time, while still considering memory needs (section 3.4.1).

Chapter 4

Benchmark-Results

In this chapter, we want to give benchmark results for the collisional system B^{5+} + H ($1s$). These results can be used to benchmark any further improvements and optimizations of the code sic3ma.

4.1 Cross Sections for B^{5+} + H ($n = 1$)

In this section we analyze the calculated CX and ionization (ION) cross sections for the collision of a fully stripped Boron ion B^{5+} with atomic Hydrogen. The calculations were conducted in the energy range of 1 to 300 keV using the following basis set:

- all complete capture channels from $n = 1$ to $n = 11$ on the Boron ion (= 286 states)
- the $n = 2$ and $n = 3$ excitation channels on the Hydrogen atom (= 10 states)
- 63 pseudostates on the Hydrogen atom modeling ionization channels

This gives a total of 359 states.

The initial condition H($n = 1$) is of course equivalent to the electron being in the $1s$ state on the Hydrogen atom when entering the collision. In fig. 4.1 the total cross section for a CX reaction into any state at the Boron ion is shown by the full line (●), while the ION cross section is indicated by the dotted line (⬟). Fig. 4.3 on the other hand displays the n-resolved cross section, enabling us to analyze the various contributions from different n-shells.

4.1.1 Total Charge Exchange Cross Section

We compare our results to various theoretical and some experimental data:

In fig. 4.1 we see a good agreement between our calculations and CTMC data by Illescas et al. (1999) [19](⊙). The same can be said for CDW-EFS data by H. Busnengo et al. (1997) [20](⊡), which are available in the high energy region of 100 to 300 keV.

AOCC data calculated by J. Hansen et al. in 1996 [21](▼) agree with our results below 6 keV, before dropping below our values for the rest of their energy range (up to 200 keV). This can most likely be attributed to the fact that J. Hanses et al. included as capture channels $2 \leq n \leq 4$ and the states $5s$ and $5p$ (see [21], Table III.). This basis set is significantly smaller than our

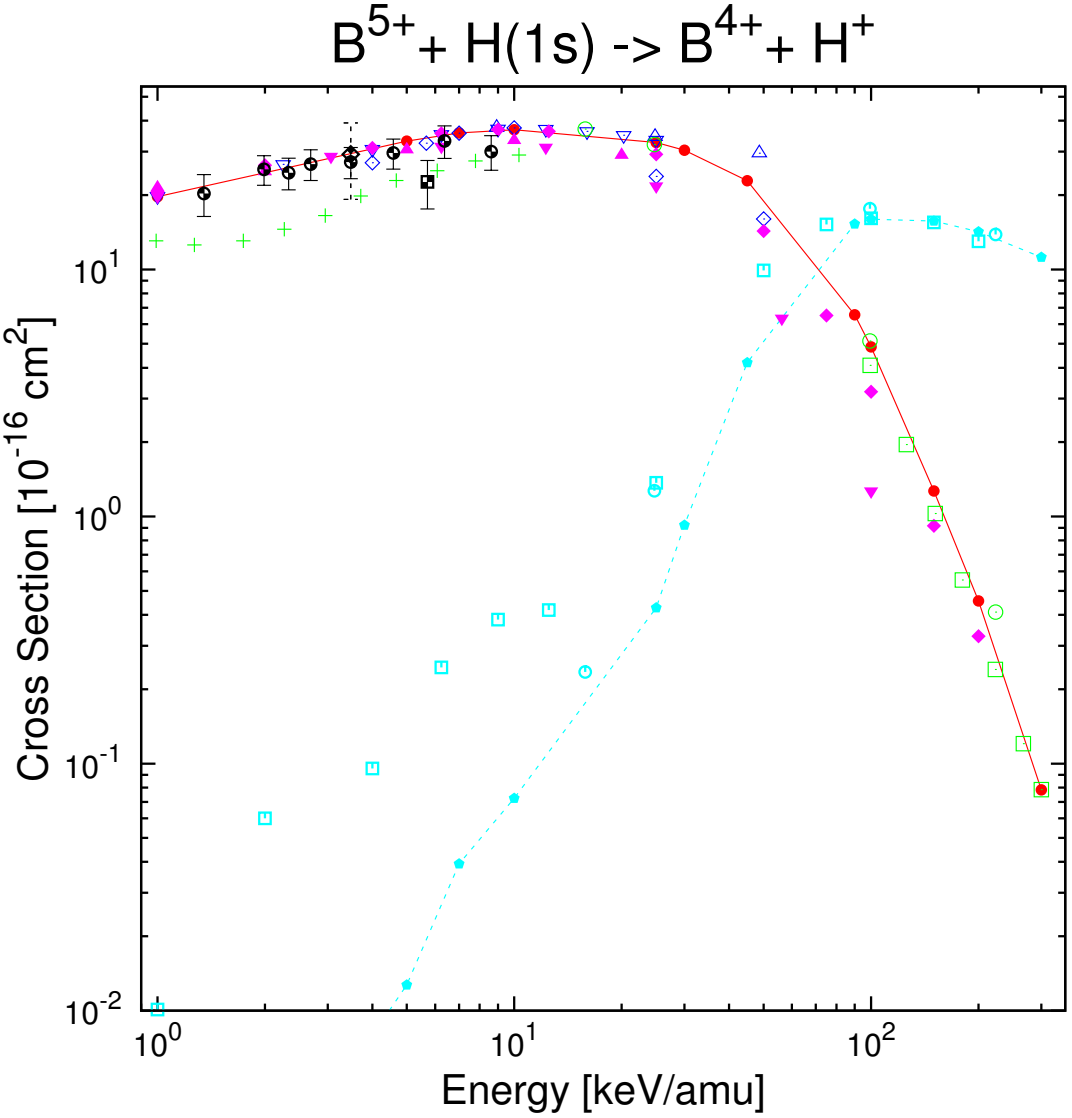


Figure 4.1: The full line (●) shows our AOCC calculations of the total cross section for electron capture from an H(1s) target to B^{5+}. Theoretical reference data for the total cross section: AOCC [21](▼), [22](▲) and [23](◆); MOCC [24](▽), [25](△) and [26](◇); CTMC [19](⊙), CDW-EFS [20](□) and EDWA [27](+). Experimental reference data for the total cross section [28] are for collisions with B^{5+}(◪), C^{5+}(❖) and N^{5+}(◓). A zoom into the low energy region $E \leq 30\ keV$ is given in fig. 4.2. The dotted line (⬟) shows the calculated ionization cross section. Reference data for ionization are AOCC [23](⊔) and CTMC [19](◷). Results are discussed in more detail in the text.

basis set and thus obviously did not contain all important capture channels for $E > 6keV$. Therefore the total cross section is too low in this energy region.

The exact same effect can be observed when comparing to results from the AOCC calculations by W. Fritsch and C.D. Lin [22](▲). They were limited to a basis of only 41 states back in 1984 and explain: "*In this study, electron transfer in B^{5+}+H collisions has been calculated mainly with a 41-AO+ basis set consisting, at the B center, of the n=3,4,5 B^{4+} orbitals and, at the H center, of the n=1 H and the 4f, 5g UA orbitals.*" (see [22], section III,D)

In 1994, Toshima included in his AOCC calculations [23](◆) bound states with $1 \leq n \leq 5$ and several unbound states on the Boron ion (see [23], TABLE V.). Toshima also explains his excitation channels on the H-atom: "*The basis set used for the atomic hydrogen is larger (...) in that the bound states with n = 3 and 4 are added.*" (see [23], section III.) These data still lie a little lower than our present results in the energy regions above 10 keV.

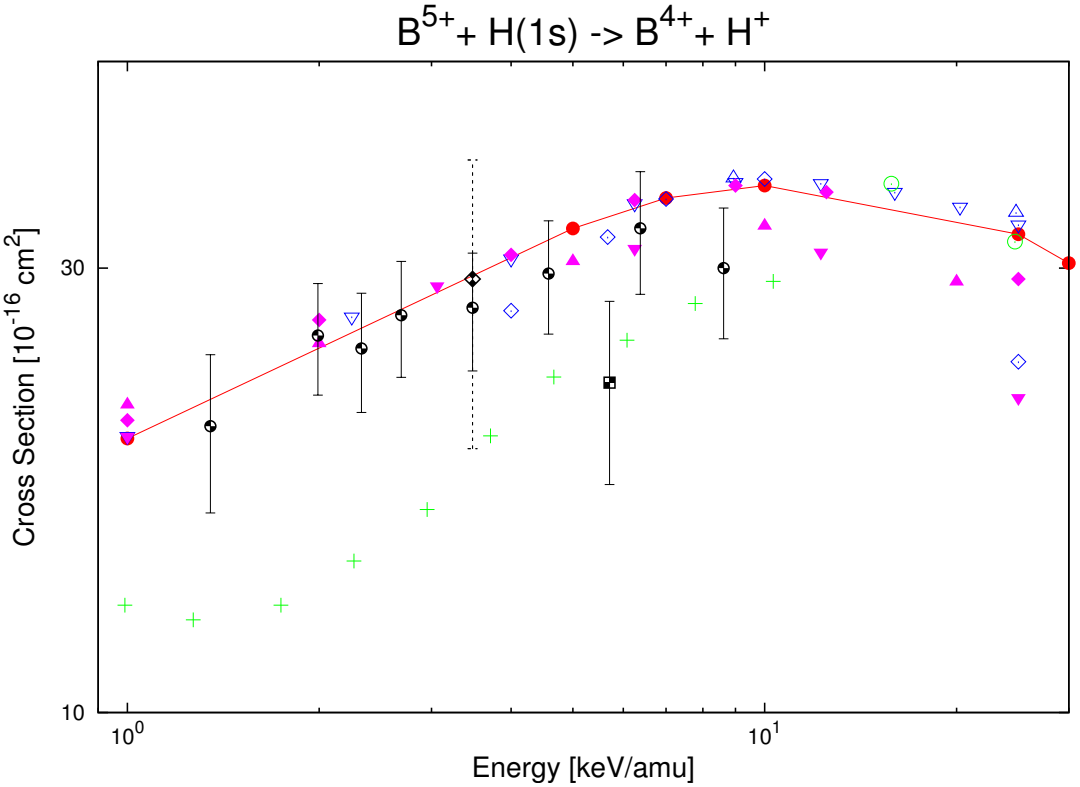


Figure 4.2: The low energy region $E \leq 30$ keV of the total cross section for electron capture from an H(1s) target to B^{5+}. Our AOCC calculations are given by the full line (●). Theoretical reference data for the total cross section: AOCC [21](▼), [22](▲) and [23](◆); MOCC [24](▽), [25](△) and [26](◇); CTMC [19](○) and EDWA [27](+). Experimental reference data for the total cross section [28] are for collisions with B^{5+}(■), C^{5+}(❖) and N^{5+}(◒). Results are discussed in more detail in the text.

Calculations using the MOCC method conducted by H. Lüdde et al. in 1982 [26](◇) deviate from our above 10 keV. The authors explain that the "*number of basis states involved is optimised during the collision process. In the energy range considered it proved to be sufficient to choose different basis sets within three ranges of the internuclear separation.*" (see [26], section 2). Their largest basis set arises from this selection for the ranges of the quantum numbers nlm:

$0 \leq m \leq 3$, $0 \leq n \leq 8$, $m \leq l \leq m + 3$ at the nucleaur separation $R \leq 1a_0$. This set is still significantly smaller than ours, since it only includes shells up to $n = 8$ and within those shells only certain l and m values.

To explain the argument given above more elaborate: for energies above $\approx$ 10 keV contributions from other channels (i.e. capture into higher n-shells, excitation to $n = 2$ or $n = 3$ and ionization) become significantly large to influence the total cross section. The different contributions at the given energy region can be seen when looking at the ION cross section in fig. 4.1 or at the contributions from higher n shells in fig. 4.3. It is obvious, that a sufficiently large basis sets needs to include all those channels.

Data from L. Errea et al. (1996) [25](△) match perfectly for their 9 keV and 25 keV value, but for 50 keV show higher values than our calculations. They used a basis set of 96 molecular orbits. They employed "*very large basis sets (96 states for both calculations), including capture channels up to n=9 as well as the n=2 excitation channel. All the sublevels up to the n=6 capture channel were included, whereas only 12 sublevels (4σ, 4π and 4δ substates) were included for the n=7, 8 and 9 capture channels;*" (see [25], section 3.1). Since MOCC calculations do not include so-called ETF, they are known to overestimate CX for higher energies.

MOCC data from C. Harel et al. (1998) [24](△) agree precisely over their entire energy range from 1 up to 25 keV, which is surprising, since their molecular basis was rather small and only "*includes all states correlating to excitation and capture channels up to n = 2 and 7, respectively, totaling 88 states.*" (see [24], page 283) Nevertheless, their value at 25 keV still fits to ours. One possible explanation could be that overestimating the cross section due to the lack of ETF was compensated by underestimating it by not including enough capture channels.

Calculations from 1984 using the EDWA approach [27](+) significantly underestimate the cross section over their entire low energy range.

As a conclusion we can say, that our calculations agree very well with various other calculations using different theoretical approaches and different computational methods in the region below 10 keV. Above this energy, older MOCC and AOCC calculations using smaller basis sets typically underestimate the CX cross section.

In 1979 D. Crandall et al. [28] measured CX for various types of ions. The available value for the collision of atomic Hydrogen with B^{5+}(■) is at $\approx$ 5.7 keV and coincides only with the significantly too low EDWA results [27], but not with any other AOCC or MOCC data.

The same paper by D. Crandall et al., however also includes measurements for collisions with C^{5+} and N^{5+}. The passive electron at the C^{5+} ion and the two passive electrons at the N^{5+} ion are very strongly bound to the nucleus and thus can be seen as a simple shielding of the core, having no significant influence on the active electron being captured into the outer shells. Igenbergs et al. [29] have conducted a numerical study on the effect of a strongly bound passive electron by comparing collisions with N^{6+} (one passive e^-) to those with C^{6+} (fully stripped). They found no significant difference in the CX cross sections for those two reactions and thus conclude, that "*it is, nevertheless, a reasonable assumption that the influence of closely bound core electrons on the active electron that captures into very high n-shells is negligible.*" (see [29], section 4.) In other words: the active electron undergoing a charge exchange reaction can hardly distinguish between a bare Boron nucleus having a charge of 5+ or another core (including one

or two strongly bound passive electrons) also having the charge 5+.

Therefore the CX process for ions having the same charge will be similar and we may also use the measurements of C^{5+} and N^{5+} for comparison. This becomes even more accurate with higher charge, since then the main capture channel is even higher and thus even further away from the passive electron(s). Following these considerations, it may be legitimate to doubt the accuracy of the B^{5+} measurement. However, all calculations lie within the error margin of the measurements for C^{5+}(❖) and N^{5+}(◒), except the 8.6 keV value for N^{5+} which is slightly lower than our calculations.

4.1.2 Ionization Cross Section

Our calculated data agree reasonably with both reference data above 100 keV. Below this energy, our calculations still roughly agree with CTMC [19](◯), but are an order of magnitude lower than Toshima [23](◫). This can be explained by the fact, that we included much more high n capture channels (up until n=11) on the Boron ion in our calculations than Toshima was able to include in 1994 (only up until n=5). These high n channels compete with ionization. This means that within Toshima's basis set the ION cross section is being overestimated at lower energy regions, since electrons that would actually be captured into higher n shells (see e.g. the n=6 or n=7 contributions at lower energies in fig. 4.3), can only choose the ionization channel.

4.1.3 n-resolved Charge Exchange Cross Section

In fig. 4.3 we see a n-resolved analysis of the CX reaction $B^{5+} + H\ (n = 1) \rightarrow B^{4+}(n) + H^{+}$. The *classical over-barrier model* (see [30], [11], section 4.1 and [31]) gives an equation to estimate which n-shell will be predominantly populated by the charge transfer reaction. It will be the n-shell with the largest integer $n_p \leq n$

$$n = q\sqrt{\frac{2\sqrt{q}+1}{2|I_t|(q+2\sqrt{q})}} = q\left(2|I_t|\left[\frac{q-1}{2\sqrt{q}}+1\right]\right)^{-1/2}, \tag{4.1}$$

q is the charge of the ion and $|I_t|$ is the binding energy of the electron at the Hydrogen atom. Plugging in the relevant values for the given system into (4.1) yields $n \approx 3.8$, making the n=4 shell the main capture channels closely followed by the n=3 shell. This can be seen in fig. 4.3. Reference data is available for n=3,4,5 and 6.

Data from C. Harel et al. (1998) [24] (available between 1 and 25 keV) fit perfectly to our results for n=4(▲) and n=5(▽), while they slightly deviate for n=3(△) and n=6(▼).

In their 1996 publication L. Errea et al. [25] calculated n-resolved cross sections up until 50 keV using the MOCC method. Their results are identical to those of C. Harel et al. and above 30 keV also fit to ours for n=4(◆) and n=3(◇). In this intermediate energy region, they are however a bit higher than our data for n=5(⬠).

AOCC calculations by J. Hansen et al. (1996) [21] overestimate the n=3(⊡) shell for energies below 10 keV, but give a good description for n=4(■). Above 10 keV, n=4 is slightly too low, while n=3 fits accurately.

Finally, we can compare to W. Fritsch and C.D. Lin [22] whose 1984 calculations agree very well with ours for n=4 (╳), while results for both n=3 (＋) and n=5 (✳) do not fully coincide with our data.

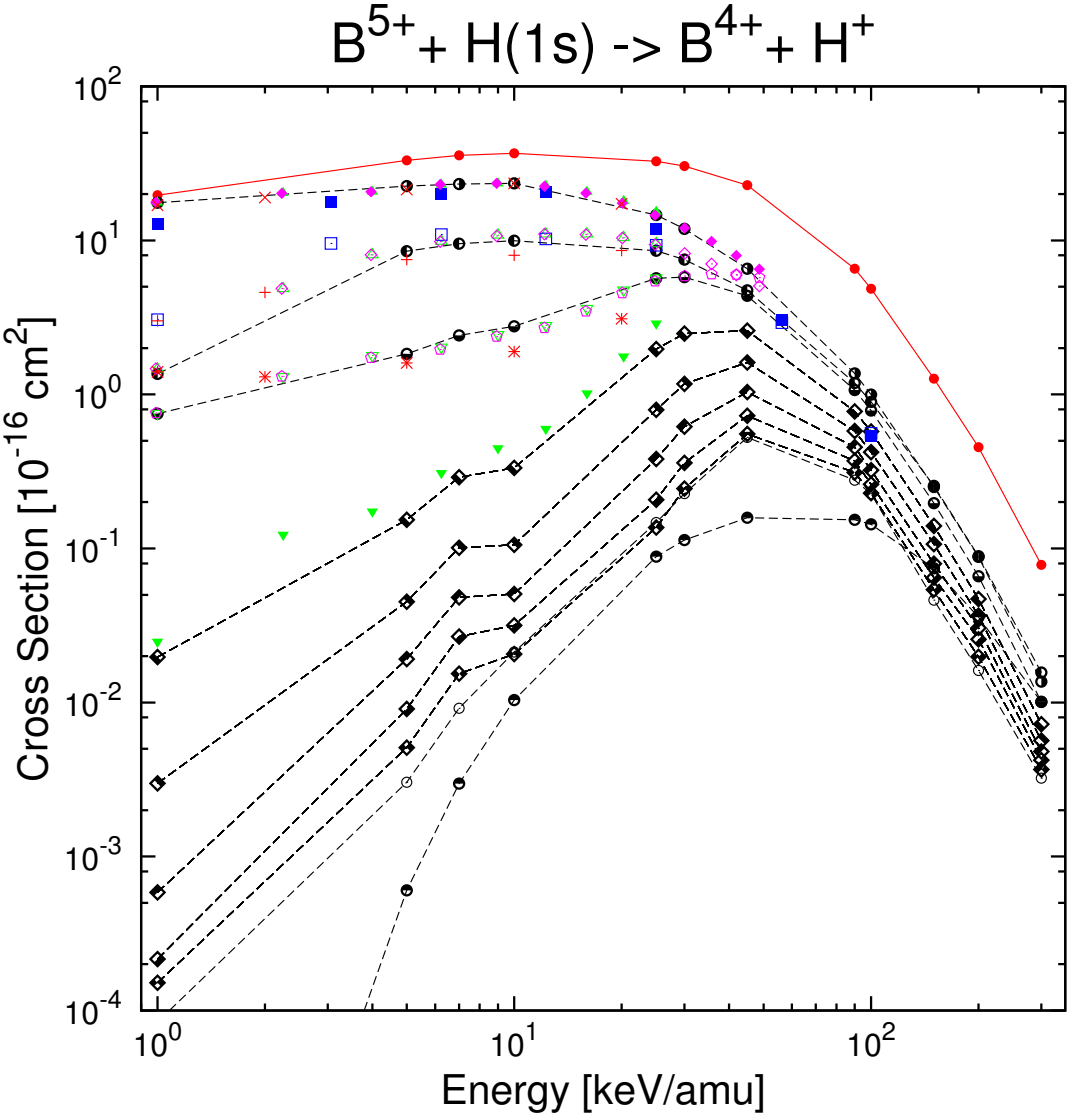


Figure 4.3: n-resolved cross section for electron capture from an H(1s) target to B^{5+}. Our calculations: full line with (●) gives the total cross section, dotted lines give the n resolved values for n=2 (◓), n=3 (◐), n=4 (◑), n=5 (◒), n=6 (◆), n=7 (◆), n=8 (◆), n=9 (◆), n=10 (❖) and n=11 (⊙). Reference data: MOCC [24] n=3(△), n=4(▲), n=5(▽), n=6(▼); MOCC [25] n=3(◇), n=4(◆), n=5(⬠); AOCC [21] n=3(□), n=4(■); AOCC [22] n=3 (┼), n=4 (╳), n=5 (✳) Results are discussed in more detail in the text.

4.2 Symbolic Structures

In this section, we want to give some benchmark data for selected symbolic structures. The polynomial of the symbolic structure stored in step 1 (compare section 2.2.1) will be given in a table to allow for comparison to newly calculated data. For each type of matrix element one selected symbolic structure is presented. Note that all involved coefficients in these structures are defined as double precision numbers.

4.2.1 One-Center Coupling Matrix Elements

The polynomial describing ONC matrix elements has four variables and section 3.2.3 explains in more detail how it can be obtained. One example of a symbolic structure (i.e. polynomial) describing such an element involving only states on the Boron ion is given in the following table. This is the polynomial describing the element $\langle B(n=4,l=0,m=0)|\ V_H\ |B(9,0,0)\rangle$:

Target-target coupling matrix element (c=1.806, d=0.000):
Linear-layout polynomial in 4 variables (13 terms):

Qy	R	R1	R3	coefficient
11	−1	0	0	1.180883109567105E+00, 0.000000000000000E+00
10	−1	0	0	−1.118558723230468E+01, 0.000000000000000E+00
9	−1	0	0	4.567820873374881E+01, 0.000000000000000E+00
8	−1	0	0	−1.059155725780084E+02, 0.000000000000000E+00
7	−1	0	0	1.551770487223557E+02, 0.000000000000000E+00
6	−1	0	0	−1.518309083220407E+02, 0.000000000000000E+00
5	−1	0	0	1.022480658540163E+02, 0.000000000000000E+00
4	−1	0	0	−4.779049136675845E+01, 0.000000000000000E+00
3	−1	0	0	1.529921944473309E+01, 0.000000000000000E+00
2	−1	0	0	−3.195802004697277E+00, 0.000000000000000E+00
1	−1	0	0	3.738209072306331E−01, 0.000000000000000E+00
0	−1	0	0	−1.944263392625372E−02, 0.000000000000000E+00
−1	−1	0	0	−1.944263391722017E−02, 0.000000000000000E+00

As an example for an element involving only states on the Hydrogen atom, the symbolic structure describing the element $\langle H(2,0,0)|\ V_B\ |H(2,0,0)\rangle$ is given in the following table:

Projectile-projectile coupling matrix element (c=1.000, d=0.000):
Linear-layout polynomial in 4 variables (4 terms):

Qy	R	R1	R3	coefficient
2	−1	0	0	−3.749999999903124E+00, 0.000000000000000E+00
1	−1	0	0	1.250000000000000E+00, 0.000000000000000E+00
0	−1	0	0	−1.250000000042938E+00, 0.000000000000000E+00
−1	−1	0	0	−1.250000000098003E+00, 0.000000000000000E+00

Note that all coefficients for ONC matrix elements are real numbers (i.e. the imaginary part of the coefficient is zero).

4.2.2 Two-Center Coupling Matrix Elements

The form of the symbolic structures describing two-center coupling matrix elements is explained in more detail in section 3.3.2. For reference we give the element $\langle H(1,0,0)|\ V_H\ |B(3,1,0)\rangle$ in the following table:

Projectile-target coupling matrix element (c=1.667, d=1.000):
Linear-layout polynomial in 7 variables (28 terms):

my	y	A	R	R1	R3	V3	coefficient
2	3	−5	4	0	0	1	0.000000000000000E+00, 3.765476942846007E+01
2	3	−6	3	0	0	1	0.000000000000000E+00, 3.765476942846007E+02
2	3	−7	2	0	0	1	0.000000000000000E+00, 1.694464624280703E+03
2	3	−8	1	0	0	1	0.000000000000000E+00, 3.953750789988308E+03
2	3	−9	0	0	0	1	0.000000000000000E+00, 3.953750789988308E+03
2	2	−4	3	0	0	1	0.000000000000000E+00, −4.066715098273687E+01
2	2	−5	2	0	0	1	0.000000000000000E+00, −2.440029058964212E+02
2	2	−6	1	0	0	1	0.000000000000000E+00, −6.100072647410531E+02
2	2	−7	0	0	0	1	0.000000000000000E+00, −6.100072647410531E+02
1	3	−4	3	0	1	0	3.765476942846007E+01, 0.000000000000000E+00
1	3	−4	3	0	0	1	0.000000000000000E+00, 2.711143398849125E+01
1	3	−5	2	0	1	0	2.259286165707604E+02, 0.000000000000000E+00
1	3	−5	2	0	0	1	0.000000000000000E+00, 1.626686039309475E+02
1	3	−6	1	0	1	0	5.648215414269011E+02, 0.000000000000000E+00
1	3	−6	1	0	0	1	0.000000000000000E+00, 4.066715098273688E+02
1	3	−7	0	0	1	0	5.648215414269011E+02, 0.000000000000000E+00
1	3	−7	0	0	0	1	0.000000000000000E+00, 4.066715098273688E+02
1	2	−3	2	0	1	0	−4.066715098273687E+01, 0.000000000000000E+00
1	2	−3	2	0	0	1	0.000000000000000E+00, −2.928034870757055E+01
1	2	−4	1	0	1	0	−1.220014529482106E+02, 0.000000000000000E+00
1	2	−4	1	0	0	1	0.000000000000000E+00, −8.784104612271166E+01
1	2	−5	0	0	1	0	−1.220014529482106E+02, 0.000000000000000E+00
1	2	−5	0	0	0	1	0.000000000000000E+00, −8.784104612271166E+01
0	3	−3	2	0	1	0	2.711143398849125E+01, 0.000000000000000E+00
0	3	−4	1	0	1	0	8.133430196547376E+01, 0.000000000000000E+00
0	3	−5	0	0	1	0	8.133430196547376E+01, 0.000000000000000E+00
0	2	−2	1	0	1	0	−2.928034870757055E+01, 0.000000000000000E+00
0	2	−3	0	0	1	0	−2.928034870757055E+01, 0.000000000000000E+00

The symbolic structure describing the two-center coupling matrix element $\langle B(3,1,0)|\, V_B\, |H(1,0,0)\rangle$ is given in the following table:

Target−projectile coupling matrix element (c=1.667, d=1.000):
Linear−layout polynomial in 7 variables (21 terms):

my	y	A	R	R1	R3	V3	coefficient	
2	3	−5	4	0	0	1	0.000000000000000E+00,	1.355571699424562E+01
2	3	−6	3	0	0	1	0.000000000000000E+00,	1.355571699424562E+02
2	3	−7	2	0	0	1	0.000000000000000E+00,	6.100072647410531E+02
2	3	−8	1	0	0	1	0.000000000000000E+00,	1.423350284395790E+03
2	3	−9	0	0	0	1	0.000000000000000E+00,	1.423350284395790E+03
2	2	−4	3	0	0	1	0.000000000000000E+00,	6.669412761168847E+01
2	2	−5	2	0	0	1	0.000000000000000E+00,	4.001647656701308E+02
2	2	−6	1	0	0	1	0.000000000000000E+00,	1.000411914175327E+03
2	2	−7	0	0	0	1	0.000000000000000E+00,	1.000411914175327E+03
2	1	−3	2	0	0	1	0.000000000000000E+00,	−5.856069741514110E+01
2	1	−4	1	0	0	1	0.000000000000000E+00,	−1.756820922454233E+02
2	1	−5	0	0	0	1	0.000000000000000E+00,	−1.756820922454233E+02
1	3	−4	3	0	1	0	1.355571699424562E+01,	0.000000000000000E+00
1	3	−5	2	0	1	0	8.133430196547375E+01,	0.000000000000000E+00
1	3	−6	1	0	1	0	2.033357549136844E+02,	0.000000000000000E+00
1	3	−7	0	0	1	0	2.033357549136844E+02,	0.000000000000000E+00
1	2	−3	2	0	1	0	6.669412761168847E+01,	0.000000000000000E+00
1	2	−4	1	0	1	0	2.000823828350654E+02,	0.000000000000000E+00
1	2	−5	0	0	1	0	2.000823828350654E+02,	0.000000000000000E+00
1	1	−2	1	0	1	0	−5.856069741514110E+01,	0.000000000000000E+00
1	1	−3	0	0	1	0	−5.856069741514110E+01,	0.000000000000000E+00

4.2.3 Two-Center Overlap Matrix Elements

The symbolic structures describing two-center overlap matrix elements are stored in the same form as for two-center coupling matrix elements (see section 3.3.2). Here, we give the element $\langle B(3,1,0)|H(1,0,0)\rangle$:

```
Target-projectile overlap matrix element (c=1.667, d=1.000):
Linear-layout polynomial in  7 variables (16 terms):
my  y  A  R R1 R3 V3        coefficient
2   3  -5 4  0  0  1    0.000000000000000E+00,-2.711143398849125E+01
2   3  -6 3  0  0  1    0.000000000000000E+00,-2.711143398849125E+02
2   3  -7 2  0  0  1    0.000000000000000E+00,-1.220014529482106E+03
2   3  -8 1  0  0  1    0.000000000000000E+00,-2.846700568791582E+03
2   3  -9 0  0  0  1    0.000000000000000E+00,-2.846700568791582E+03
2   2  -4 3  0  0  1    0.000000000000000E+00, 2.928034870757055E+01
2   2  -5 2  0  0  1    0.000000000000000E+00, 1.756820922454233E+02
2   2  -6 1  0  0  1    0.000000000000000E+00, 4.392052306135582E+02
2   2  -7 0  0  0  1    0.000000000000000E+00, 4.392052306135582E+02
1   3  -4 3  0  1  0   -2.711143398849125E+01, 0.000000000000000E+00
1   3  -5 2  0  1  0   -1.626686039309475E+02, 0.000000000000000E+00
1   3  -6 1  0  1  0   -4.066715098273688E+02, 0.000000000000000E+00
1   3  -7 0  0  1  0   -4.066715098273688E+02, 0.000000000000000E+00
1   2  -3 2  0  1  0    2.928034870757055E+01, 0.000000000000000E+00
1   2  -4 1  0  1  0    8.784104612271166E+01, 0.000000000000000E+00
1   2  -5 0  0  1  0    8.784104612271166E+01, 0.000000000000000E+00
```

4.3 Continuity of the Elements of $\hat{M}_{\text{eff}}$

After the effective matrices $\hat{M}_{\text{eff}}$ have been calculated in the computational step 3 (compare 2.2.3), it is advisable to check the calculated results. However, one is usually limited to making a check for plausibility, since it is computationally not feasible to calculate an entire matrix for a larger system by hand or even using Mathematica. Mathematica already needs a lot of time for evaluating a single matrix element (as briefly described in section 3.1.3), let alone an entire 1500×1500 matrix.

A good way to do such a plausibility check, is to check the change of any given matrix element of the effective matrices $\hat{M}_{\text{eff}}$ along the trajectory. Both real and imaginary part of the matrix elements have to be continuous along r_3. This continuity is important, because during step 4 (see section 2.2.4), matrix elements have to be interpolated for all the values of r_3 needed during the integration along the trajectory. An interpolation is only possible if the elements are continuous.

In fig. 4.4 we have plotted the first element $(1,1)$ of the effective matrix for two different energies 1 keV and 25 keV. The impact parameter determining the trajectory is given by $b = 1$. Fig. 4.5 shows the continuity of the imaginary part of the element $(1,1)$, using the same values for b and impact energy.

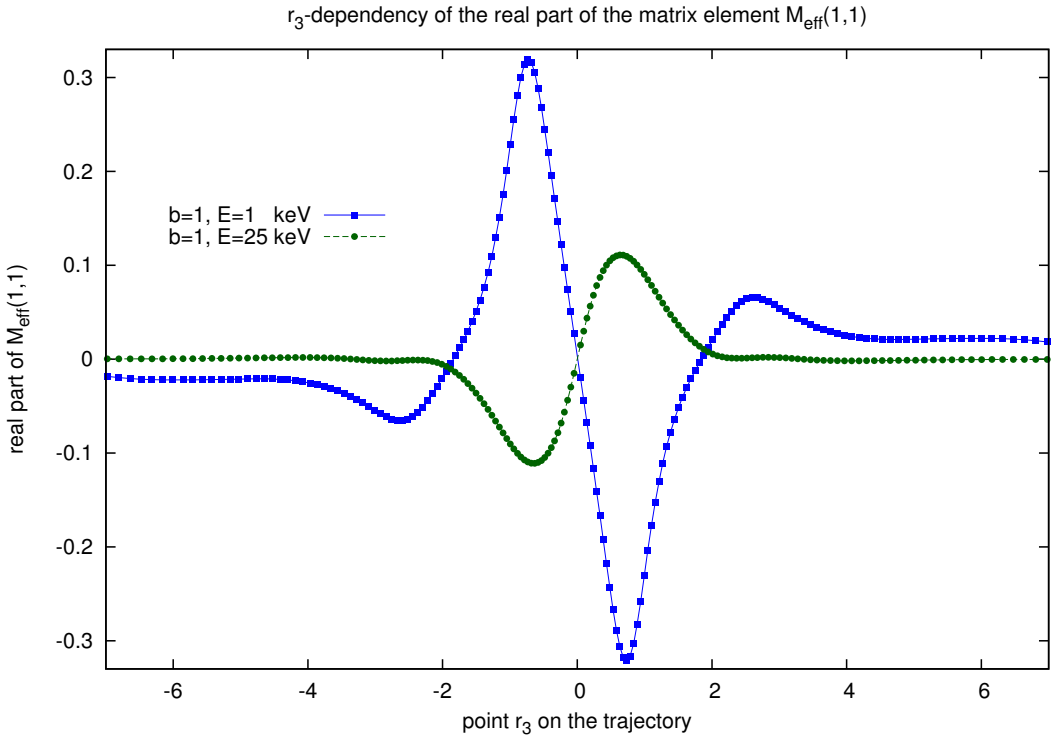


Figure 4.4: The dependency of the real part of $\hat{M}_{\text{eff}}(1,1)$ along the trajectory points r_3. The values are given for one certain impact parameter b and two different impact energies E.

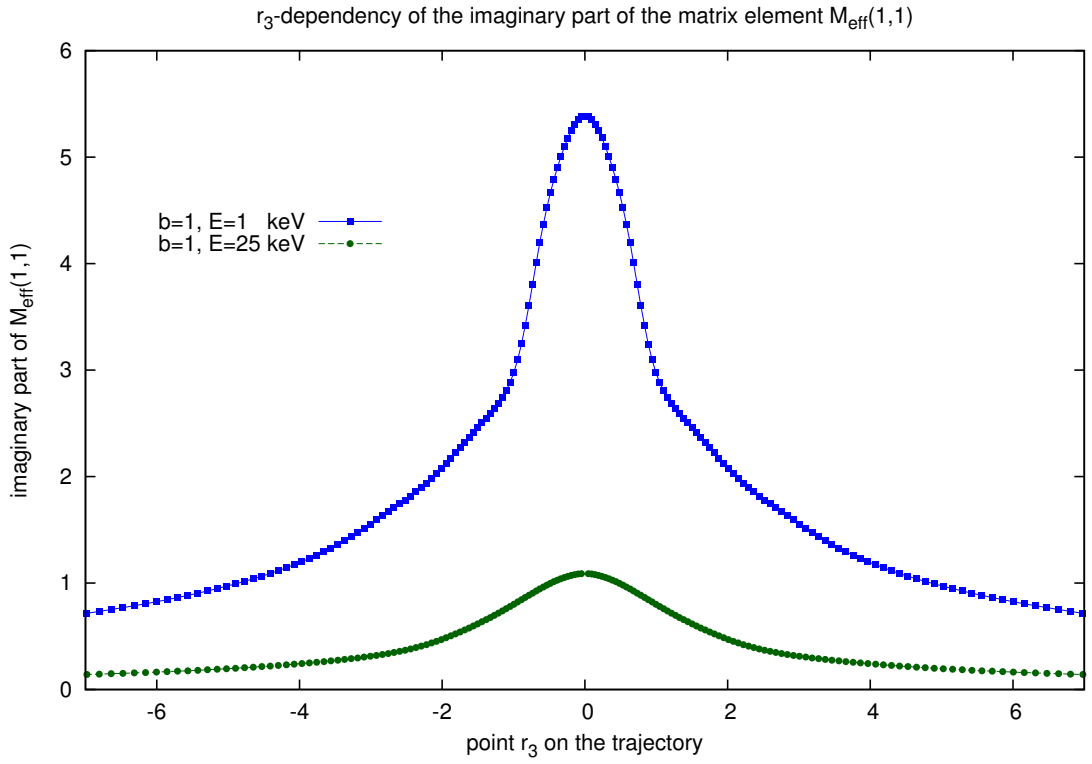


Figure 4.5: The imaginary part of $\hat{M}_{\text{eff}}(1,1)$ as it changes along the trajectory. The impact parameter is given by $b = 1$ and defines the given trajectory according to fig. 2.2 . The plot was done for two different impact energies E. The values of $\hat{M}_{\text{eff}}(1,1)$ are clearly continuous, making it possible to interpolate intermediate values in the subsequent integration during step 4 (see section 2.2.4).

Chapter 5

Conclusion and Outlook

In the first chapters of this thesis we outlined the theoretical foundation for the calculations of cross sections for ion-atom collisions, namely the well established semi-classical AOCC theory. We continued to explain in detail how the underlying theory is implemented into the parallel FORTRAN code "sic3ma" and how each computational step is performed.

In the following chapter, we gave an overview on current limits of the code "sic3ma" as far as numerical accuracy in the calculation of matrix elements is concerned and then suggested ways to push these numerical limits further. We have shown that numerical instabilities could be reduced by implementing a new and more accurate gamma function and by using quadruple precision instead of double precision numbers for the complex coefficients involved in the evaluation. Furthermore, we were able to correctly calculate matrix elements involving states with quantum numbers as high as $n = 25$, while the previous limit for these calculations was at $n = 12$. Along with this detailed numerical study of the evaluation of matrix elements in the context of the AOCC theory, we also presented codes for the computational software program "Mathematica" and the small subroutine "onc_coupling" which can both be used to obtain reference values for most of the considered matrix elements. To show the achieved improvements, these reference values were compared to the code's calculated results.

Besides improvements of the numerical accuracy of the calculations, we also discussed hardware requirements which have to be met in order for the calculation to be possible. The memory which is available per computing node on a cluster computer becomes a very problematic limiting factor for computations producing and calculating with huge amounts of data. We have shown that this memory limit needs to be considered and an appropriate distribution of all data needed during the calculation must be found. In addition to that, we also suggested enhancements to the computational performance with a special emphasis on the reduction of idle time during the calculation.

Finally, a detailed analysis of the results calculated for the collisional system $B^{5+} + H(1s)$ was given. This analysis not only includes the final results (i.e. partial cross sections for the various channels of the reaction) but also gives intermediate results, both quantitatively and qualitatively. These results can be used in the future as a benchmark for further developments of the code "sic3ma".

Using the work done in this thesis as a guideline for overcoming numerical and computational difficulties arising during the calculation of large collisional systems, one can now analyze a system like $Ar^{18+} + H$ including Argon states as far as $n = 20$. With the current version of

the code, the calculation of partial cross sections for this system at 12 different impact energies, including 73 basis states on the Hydrogen atom and 24 impact parameters (this represents a very detailed study of the collisional system) will approximately need the following computational resources[1]:

- approx. 3 TB of hard disc for the symbolic structures and the matrix files
- approx. 950 000 CPU-h

Given the fact that both computational power and hard disc memory on cluster computers keep getting cheaper, even a rather expensive calculation like the one above becomes more and more feasible. Most of the numerical difficulties could be solved within the scope of this work, but there are still some performance issues (e.g. reducing idle time during the evaluation of the matrix elements) which need to be addressed in order to use the available computational resources more efficiently.

This thesis shows that the AOCC method and its implementation into the code "sic3ma" is suitable to describe collisional systems with ions such as Ar^{18+}, although the computational efforts are quite considerable. Using a basis set including roughly 1500 basis states (to describe the Argon ion using $n = 1$ to 20) implies that the matrices involved in the calculations are of size 1500×1500. From a numerical point of view, the matrix elements could currently be calculated using states with quantum numbers as high as $n = 25$. However, the rest of the computational steps become increasingly unfeasible with a growing basis set. It seems very difficult to push the limit much further than $n = 25$, since the description of larger, heavier ions will require huge basis sets. For example, it takes roughly 5000 basis states to include all states up to $n = 30$ in the description of the ion. Such a huge basis set will make computation times far too long and it will basically become impossible to meet hardware requirements such as memory limits. In addition to that, it also seems difficult to push the numerical limit further than $n = 25$, since that would require higher precision numbers for the involved coefficients than the current highest possible quadruple precision.

Tungsten will probably be the main component of ITER's wall and the collisional system $W^{q+} + H$ (with a multitude of charge states $q+$) would thus be of great interest. Tungsten has an atomic number of $Z = 74$, which means that one would have to include states up until $n = 76$ to sufficiently describe its fully stripped ion. This yields a basis set of roughly 76 000 states. Seeing as this is simply impossible to calculate within the framework of the AOCC method, it seems as though different theoretical approaches (e.g. CTMC methods or using different types of basis states, such as Gaussian orbitals, within a close-coupling theory) will have to be used to calculate a collisional system involving Tungsten.

[1] We estimated 1 TB for the symbolic structures and 2 TB for the matrix files to get the memory needs on the hard disc. The computation time is obtained by the following estimate: It takes roughly 100 000 CPU-h to calculate all symbolic structures and roughly 70 000 CPU-h for each impact energy to evaluate the matrix files along a 400-point trajectory. Assuming 12 impact energies, we get $70\,000 \times 12 = 840\,000$. Evaluation time can definitely be reduced considerably by implementing the suggestions made at the end of section 3.4.2.

Appendix A

The subroutine "onc_coupling"

We shall give the general analytic calculation of ONC matrix elements, which have the form

$$M_{\alpha\beta} =_A \langle\phi_\alpha|\ V_B\ |\phi_\beta\rangle_A = \langle Z_A, n_\alpha, l_\alpha, m_\alpha|\ \frac{Z_B}{|\vec{r}-\vec{R}|}\ |Z_A, n_\beta, l_\beta, m_\beta\rangle\ , \tag{A.1}$$

for arbitrary s-states $|Z, n, 0, 0\rangle$ in section A.1. We will ignore Z_B during this calculation since it can simply be pulled in front of the entire expression and thus can be multiplied with the result at the very end. We then explicitly evaluate the matrix elements $\langle 1,1,0,0|V_B|1,1,0,0\rangle$ and $\langle 1,2,0,0|V_B|1,2,0,0\rangle$ for $Z_B = 1$ in A.2.

A.1 Analytical Calculation

The calculations are done in the coordinate system A, which is turned in alignment with the vector $\vec{r}$. The geometry is shown in fig. A.1. Turning the z-axis is possible, because the states on center A are s-states and thus have have spherical symmetry allowing an arbitrary choice of the z-axis.

A.1.1 Expansion of $|\phi_\alpha\rangle$, $|\phi_\beta\rangle$ and $|\vec{r}-\vec{R}|^{-1}$

To calculate the expression (A.1) analytically, we follow the steps taken in [18], section 3.3. The first step is to expand the s-states $|\phi_\alpha\rangle$ and $|\phi_\beta\rangle$ into so-called *Slater Orbitals* in the following way (see [18], eq.(3.3.4)):

$$|\phi_\alpha\rangle = \sum_{i=1}^{\infty} N_i^\alpha r^{a_i^\alpha} e^{-b_i^\alpha r} \cdot Y_{00} \tag{A.2}$$

$$|\phi_\beta\rangle = \sum_{j=1}^{\infty} N_j^\beta r^{a_j^\beta} e^{-b_j^\beta r} \cdot Y_{00} \tag{A.3}$$

Here Y_{00} is the spherical harmonic determining the angular shape of an s-state wavefunction, r denotes the radial distance, $N_{i/j}^{\alpha/\beta}$ are normalization coefficients, $a_{i/j}^{\alpha/\beta}$ and $b_{i/j}^{\alpha/\beta}$ are coefficients determining the radial shape of the wavefunction. In subsection A.1.3 we will see how they can be calculated. When looking at the general form of a Hydrogen-like wavefunction in (2.13), it can be seen, that these coefficients will always be real, since the only imaginary part would come from the spherical harmonic Y_{00} (which is incidentally also a real). This means, that the coefficients do not change when we built the bra-vector $\langle\phi_\alpha|$.

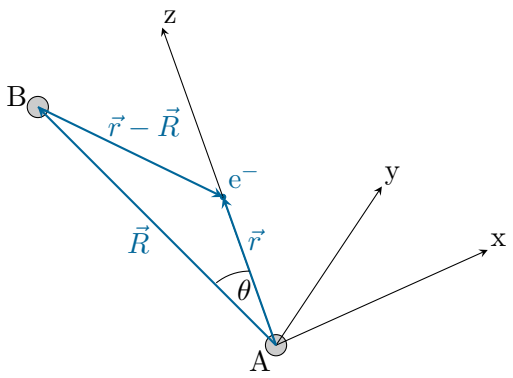


Figure A.1: The geometry used in "onc_coupling". The elements are calculated using states on center A and the potential from center B. Everything is expressed in terms of spherical coordinates (r, θ, ϕ) in coordinate system A, but this system is tilted to align the z-axis with the vector $\vec{r}$ in order to be able to use the polar angle θ as the angle between $\vec{r}$ and $\vec{R}$. This is needed to expand the expression $V = |\vec{r}-\vec{R}|^{-1}$ in terms of Legendre-Polynomials using θ.

Knowing this, we can rewrite (A.1) in the following way:

$$\begin{aligned} M_{\alpha\beta} &= \int dr\ r^2 \int d\Omega \sum_i N_i^\alpha r^{a_i^\alpha} e^{-b_i^\alpha r}\ Y_{00}^* \frac{1}{|\vec{r}-\vec{R}|} \sum_j N_j^\beta r^{a_j^\beta} e^{-b_j^\beta r}\ Y_{00} \\ &= \int dr\ r^2 \int d\Omega \sum_{ij} N_i^\alpha N_j^\beta\ r^{a_i^\alpha + a_j^\beta}\ e^{-r(b_i^\alpha + b_j^\beta)}\ Y_{00}^* \frac{1}{|\vec{r}-\vec{R}|} Y_{00} \end{aligned} \tag{A.4}$$

Now we expand $V = |\vec{r}-\vec{R}|^{-1}$ using Legendre-Polynomials and the angle θ between the vectors $\vec{r}$ and $\vec{R}$ [1] (see [18], eq.(3.3.12)):

$$V = \frac{1}{|\vec{r}-\vec{R}|} = \begin{cases} \frac{1}{r}\sum_{k=0}^{\infty} \left(\frac{R}{r}\right)^k P_k(\cos\theta), & r > R \\ \frac{1}{R}\sum_{k=0}^{\infty} \left(\frac{r}{R}\right)^k P_k(\cos\theta), & r < R \end{cases} \tag{A.5}$$

By inserting this expansion into (A.4) we can separate the integral into a radial part and an angular part.

$$\begin{aligned} M_{\alpha\beta} = \sum_{k=0}^{\infty} \Bigg[\int d\Omega\ Y_{00}^*\ P_k\ Y_{00}\ \cdot \sum_{ij} N_i^\alpha N_j^\beta \Bigg(\frac{1}{R^{k+1}} \int_0^R r^{(a_i^\alpha + a_j^\beta + k + 2)} e^{-r(b_i^\alpha + b_j^\beta)}\ dr \\ + R^k \int_R^\infty r^{(a_i^\alpha + a_j^\beta - k + 1)} e^{-r(b_i^\alpha + b_j^\beta)}\ dr \Bigg) \Bigg] \end{aligned} \tag{A.6}$$

We now define the quantities

$$A_{\alpha\beta}^k = \int d\Omega\ Y_{00}^*(\Omega)\ P_k(\cos\theta)\ Y_{00}(\Omega) \tag{A.7}$$

[1] It is of importance to mention here, that using the polar angle θ of the coordinate system A as the angle between $\vec{r}$ and $\vec{R}$ in (A.5) is only possible, because we are dealing with s-states on center A. Only then, are the states independent of the angles θ and ϕ and we can turn the z-axis in alignment with $\vec{r}$ allowing us to use the angle θ in the expansion (see fig. A.1). If the states were not s-states and we could not turn the z-axis, we would have to use a different angle θ' in (A.5) and then the integrant in (A.11) would read $Y_{00}^*(\Omega')\ Y_{k0}(\Omega)$ and the orthonormality would be not applicable, since Y_{00}^* and Y_{k0} would have different arguments Ω' and Ω.

$$R^k_{\alpha\beta} = \sum_{ij} N_i^\alpha N_j^\beta \left(\frac{1}{R^{k+1}} \int_0^R r^{(a_i^\alpha + a_j^\beta + k + 2)} e^{-r(b_i^\alpha + b_j^\beta)}\, dr + R^k \int_R^\infty r^{(a_i^\alpha + a_j^\beta - k + 1)} e^{-r(b_i^\alpha + b_j^\beta)}\, dr \right) \tag{A.8}$$

and can use them to write (A.6) in a very compact way:

$$M_{\alpha\beta} = \sum_{k=0}^{\infty} \left[A^k_{\alpha\beta} \cdot R^k_{\alpha\beta} \right] \tag{A.9}$$

A.1.2 Angular Integration

The integral (A.7) can be be solved easily by rewriting $P_k(\cos\theta)$ in terms of a spherical harmonic, using the relation $P_k = \sqrt{4\pi/(2k+1)} \cdot Y_{k0}$ (see [18], p.43). This yields:

$$A^k_{\alpha\beta} = \sqrt{\frac{4\pi}{2k+1}} \int d\Omega\; Y^*_{00}(\Omega)\; Y_{k0}(\Omega)\; Y_{00}(\Omega) \tag{A.10}$$

By evaluating $Y_{00} = 1/\sqrt{4\pi}$ (see [32], p.600) and using the orthonormality of spherical harmonics, we get

$$A^k_{\alpha\beta} = \sqrt{\frac{4\pi}{2k+1}} \underbrace{\int d\Omega\; Y^*_{00}(\Omega)\; Y_{k0}(\Omega)}_{=\delta_{0k}} \frac{1}{\sqrt{4\pi}} = \sqrt{\frac{1}{2k+1}}\, \delta_{0k}\ , \tag{A.11}$$

from where we can see that

$$A^k_{\alpha\beta} = \begin{cases} 1 & , \quad k = 0 \\ 0 & , \quad \text{all other k} \end{cases} \tag{A.12}$$

This implies two important things:

1. The sum over k in (A.9) is not infinite, but actually only consists of one term, namely the term with $k = 0$.
2. For s-states, the angular integral is always 1 and independent from the value of n.

A.1.3 Radial Integration

The radial integration can be solved by making a variable transformation $t = r(b_i^\alpha + b_j^\beta)$, which allows us to rewrite both integrals in (A.8) to the form (we already use $k = 0$)

$$R^0_{\alpha\beta} = \sum_{ij} N_i^\alpha N_j^\beta \left(\frac{1}{R} \frac{1}{(b_i^\alpha + b_j^\beta)^{(a_i^\alpha + a_j^\beta + 3)}} \underbrace{\int_0^{R(b_i^\alpha + b_j^\beta)} t^{(a_i^\alpha + a_j^\beta + 2)} e^{-t}\, dt}_{\gamma\{(a_i^\alpha + a_j^\beta + 3),\ R(b_i^\alpha + b_j^\beta)\}} + \frac{1}{(b_i^\alpha + b_j^\beta)^{(a_i^\alpha + a_j^\beta + 2)}} \underbrace{\int_{R(b_i^\alpha + b_j^\beta)}^\infty t^{(a_i^\alpha + a_j^\beta + 1)} e^{-t}\, dt}_{\Gamma\{(a_i^\alpha + a_j^\beta + 2),\ R(b_i^\alpha + b_j^\beta)\}} \right),$$

where we can identify the first integral with the *lower incomplete gamma function* $\gamma(s, x)$ and the second integral with the *upper incomplete gamma function* $\Gamma(s, x)$ which are defined as

$$\gamma(s, x) = \int_0^x dt\; t^{s-1} e^{-t} \tag{A.13}$$

$$\Gamma(s,x) = \int_x^\infty dt\ t^{s-1}e^{-t} \tag{A.14}$$

This yields the pretty manageable expression

$$R^0_{\alpha\beta} = \sum_{ij} N_i^\alpha N_j^\beta \left(\frac{1}{R}\ \frac{1}{g^{h_1}}\ \gamma(h_1, R\cdot g) + \frac{1}{g^{h_2}}\ \Gamma(h_2, R\cdot g) \right) , \tag{A.15}$$

where $h_1 = (a_i^\alpha + a_j^\beta + 3)$, $h_2 = (a_i^\alpha + a_j^\beta + 2)$ and $g = (b_i^\alpha + b_j^\beta)$. To solve this, we need to know the coefficients $N_{i/j}^{\alpha/\beta}$, $a_{i/j}^{\alpha/\beta}$ and $b_{i/j}^{\alpha/\beta}$. They can be determined by comparing the expansions (A.2) and (A.3) respectively, which are of the form

$$|Z,n,0,0\rangle = \sum_{i/j} N_{i/j}^{\alpha/\beta} r^{a_{i/j}^{\alpha/\beta}} e^{-b_{i/j}^{\alpha/\beta} r} \cdot Y_{00} , \tag{A.16}$$

to the general expression of a s-state hydrogen-like wavefunction. Such a wavefunction is given by (2.13). With the definition (2.14) of the Legendre Polynomials and if we set $l = m = 0$ in (2.13), we can always identify the coefficient $b_{i/j}^{\alpha/\beta}$ from (A.16) with $Z_{\alpha/\beta}/n_{\alpha/\beta}$, since Z/n is the exponent of the exponential in (2.13) and the expression (2.14) will not add any terms with an exponential in it. The coefficients $a_{i/j}^{\alpha/\beta}$ and $N_{i/j}^{\alpha/\beta}$ must be found by additionally inserting the given values Z and n into (2.13) and then making a direct comparison of coefficients to (A.16).

A.2 Evaluation of Some Matrix Elements

We now want to explicitly calculate the values of the expressions $\langle H(1s)|V_{Hydrogen}|H(1s)\rangle$ and $\langle H(2s)|V_{Hydrogen}|H(2s)\rangle$.

A.2.1 $\langle H(1s)|V_{Hydrogen}|H(1s)\rangle$

For the expression $\langle 1,1,0,0|V|1,1,0,0\rangle$, we get $|\phi_\alpha\rangle = |\phi_\beta\rangle$ and $Z_A = Z_B = n_\alpha = n_\beta = 1$. From (A.12), we know that the angular part of the integral is 1 and $k = 0$, so we only need to calculate the radial part $R^0_{\alpha\alpha}$.

To evaluate expression (A.15), we need to determine the values of $N_{i/j}$, $a_{i/j}$ and $b_{i/j}$ by comparing (A.16) to the wavefunction of the $H(1s)$ state, which (in atomic units) is given by (compare [33], chapter 5.1.3)

$$|1,1,0,0\rangle = \frac{e^{-r}}{\sqrt{\pi}} = e^{-r}\cdot 2\cdot \underbrace{\frac{1}{\sqrt{4\pi}}}_{Y_{00}} = 2\cdot r^0\cdot e^{-r}\cdot Y_{00} .$$

It is obvious that i and j only take the value 1, with $N_1^{\alpha/\beta} = 2$, $a_1^{\alpha/\beta} = 0$ and $b_1^{\alpha/\beta} = 1$. Entering all this into (A.15) yields

$$R^0_{\alpha\alpha} = 4\cdot \left(\frac{1}{R}\frac{1}{2^3}\ \gamma(3,2R)\ + \frac{1}{2^2}\ \Gamma(2,2R) \right) .$$

For our chosen value of $R \approx 7$ and by using *WolframAlpha* to compute

$$\gamma(3,14) \approx 1.9998 \quad , \quad \Gamma(2,14) \approx 0.00001$$

we get

$$R^0_{\alpha\alpha} \approx 0.1429 \ .$$

With $A^0_{\alpha\alpha} = 1$, we finally get:

$$\boxed{\langle 1,1,0,0|V_{Hydrogen}|1,1,0,0\rangle \approx 0.1429} \tag{A.17}$$

A.2.2 $\langle H(2s)|V_{Hydrogen}|H(2s)\rangle$

For the expression $\langle 1,2,0,0|V|1,2,0,0\rangle$ we have again $|\phi_\alpha\rangle = \phi_\beta\rangle$, $Z_A = Z_B = 1$, while $n_\alpha = n_\beta = 2$ in the case of the 2s state. The angular part is of course again $A^0_{\alpha\alpha} = 1$.

For the radial integral, we again compare (A.16) to the $2s$ wavefunction (compare [33], chapter 5.1.3)

$$|1,2,0,0\rangle = \frac{1}{4\sqrt{2\pi}}\Big(2-r\Big)\ e^{-\frac{1}{2}r} = \Big(\frac{1}{\sqrt{2}}\ e^{-\frac{1}{2}r}\ -\ \frac{1}{2\sqrt{2}}\ r\ e^{-\frac{1}{2}r}\Big)\cdot Y_{00}\ .$$

This gives us $N_1^{\alpha/\beta} = 1/\sqrt{2}$, $a_1^{\alpha/\beta} = 0$, $b_1^{\alpha/\beta} = b_2^{\alpha/\beta} = 1/2$, $N_2^{\alpha/\beta} = -1/(2\sqrt{2})$ and $a_2^{\alpha/\beta} = 1$. The indices i and j now run from 1 to 2 and we thus get 4 terms in the sum of (A.15). However, the expression for $i = 1,\ j = 2$ is the same as for $i = 2,\ j = 1$, thus we only need to compute 3 terms. We again take $R \approx 7$ and get

$$\begin{aligned} R^0_{\alpha\alpha} &= \sum_{i=1}^{2}\sum_{j=1}^{2} N_i N_j \left(\frac{1}{7}\ \gamma(a_i+a_j+3,7) + \ \Gamma(a_i+a_j+2,7)\right) \\ &= \frac{1}{2}\left(\frac{1}{7}\ \gamma(3,7) + \ \Gamma(2,7)\right) + \frac{1}{8}\left(\frac{1}{7}\ \gamma(5,7) + \ \Gamma(4,7)\right) \\ &\quad + 2\cdot(-\frac{1}{4})\left(\frac{1}{7}\ \gamma(4,7) + \ \Gamma(3,7)\right) \ \approx \ 0.135\ . \end{aligned}$$

This eventually yields:

$$\boxed{\langle 1,2,0,0|V_{Hydrogen}|1,2,0,0\rangle \approx 0.135} \tag{A.18}$$

Appendix B

Mathematica codes

B.1 One-Center Coupling Matrix Elements

In this section we give mathematica codes for ONC matrix elements. Refer to section 3.2.2 for documentation.

```
Radial[Z_, n_, l_, r_] :=
  Sqrt[(Z/n)^(3) (4 (n - l - 1)!)/((n (n + l)!))] E^(-((Z r)/
       n)) ((2 Z r)/n)^l LaguerreL[n - l - 1, 2 l + 1, (2 Z r)/n] ;
state[Z_, n_, l_, m_, r_, theta_, phi_] =
  Radial[Z, n, l, r] SphericalHarmonicY[l, m, theta, phi];
d = SetPrecision[7.011890604642960989022286130409 14, 32];
b = SetPrecision[5.0, 32];
alpha = ArcSin[b/d];
a[r_, theta_, phi_] := {r Sin[theta] Cos[phi], r Sin[theta] Sin[phi],
   r Cos[theta]};
V[r_, theta_, phi_] := 1/(Norm[a[r, theta, phi] - a[d, alpha, 0]]);
Do[Print[n] Print[
   NIntegrate[(r^2 Sin[theta] state[18, n, 1, 0, r, theta, phi] V[r,
       theta, phi] state[18, n, 1, 0, r, theta, phi]), {r, 0,
     Infinity}, {theta, 0, Pi}, {phi, 0, 2 Pi},
    Method -> {"GlobalAdaptive",
      "SingularityHandler" -> "DuffyCoordinates",
      Method -> "MultiDimensionalRule", MaxErrorIncreases -> 10000},
    PrecisionGoal -> 10, WorkingPrecision -> 30]], {n, 23, 23, 1}]
```

B.2 Two-Center Coupling Matrix Elements

In this section we give mathematica codes for TWC matrix elements. Refer to section 3.3.1 for documentation. This code only works for matrix elements involving s-states.

```
Radial[Z_, n_, l_, r_] :=
  Sqrt[(Z/n)^(3) (4 (n - l - 1)!)/((n (n + l)!))] E^(-((Z r)/
       n)) ((2 Z r)/n)^l LaguerreL[n - l - 1, 2 l + 1, (2 Z r)/n] ;
state[Z_, n_, l_, m_, r_, theta_] :=
  Radial[Z, n, l, r] SphericalHarmonicY[l, m, theta, 0];
d = SetPrecision[7.011890604642960989022286130409 14, 32];
b = SetPrecision[5.0, 32];
vv = SetPrecision[0.531293320774880406070933573639680, 32];
gamma = ArcSin[b/d];
r2[r_, theta_] := Sqrt[d^2 + r^2 - 2 r*d*Cos[theta]];
theta2[r_, theta_] :=
  ArcSin[(r*Sin[theta])/(Sqrt[d^2 + r^2 - 2 r*d*Cos[theta]])];
ETF[r_, theta_] :=
  E^(-I*vv*r*(Cos[theta]*Cos[gamma] + Sin[theta]*Sin[gamma]));
a[r_, theta_] := {r Sin[theta], 0, r Cos[theta]};
V[r_, theta_] := 18/(Norm[a[r, theta]]);
Do[Print[n] Print[
   Abs[NIntegrate[(r^2 Sin[theta] Conjugate[
        state[1, 1, 0, 0, r2[r, theta], theta2[r, theta]]] ETF[r,
        theta] V[r, theta] state[18, n, 0, 0, r, theta]), {r, 0,
      Infinity}, {theta, 0, Pi}, {phi, 0, 2 Pi},
     Method -> {"GlobalAdaptive",
       "SingularityHandler" -> "DuffyCoordinates",
       Method -> "MultiDimensionalRule", MaxErrorIncreases -> 10000},
     PrecisionGoal -> 10, WorkingPrecision -> 30]]], {n, 10, 14, 1}]
```

B.3 Two-Center Overlap Matrix Elements

In this section we give mathematica codes for two-center overlap matrix elements. Refer to section 3.3.1 for documentation. This code only works for matrix elements involving s-states.

```
Radial[Z_, n_, l_, r_] :=
  Sqrt[(Z/n)^(3) (4 (n - l - 1)!)/((n (n + l)!))] E^(-((Z r)/
       n)) ((2 Z r)/n)^l LaguerreL[n - l - 1, 2 l + 1, (2 Z r)/n] ;
state[Z_, n_, l_, m_, r_, theta_] :=
  Radial[Z, n, l, r] SphericalHarmonicY[l, m, theta, 0];
d = SetPrecision[7.011890604642960989022286130409 14, 32];
b = SetPrecision[5.0, 32];
v = SetPrecision[0.531293320774880406070933573639680, 32];
gamma = ArcSin[b/d];
r2[r_, theta_] := Sqrt[d^2 + r^2 - 2 r*d*Cos[theta]];
ETF[r_, theta_] :=
  E^(-I*v*r*(Cos[theta]*Cos[gamma] + Sin[theta]*Sin[gamma]));
theta2[r_, theta_] :=
  ArcSin[(r*Sin[theta])/(Sqrt[d^2 + r^2 - 2 r*d*Cos[theta]])];
Do[Print[n] Print[
   Abs[NIntegrate[(r^2 Sin[theta] Conjugate[
        state[1, 1, 0, 0, r2[r, theta], theta2[r, theta]]] ETF[r,
        theta] state[18, n, 0, 0, r, theta]), {r, 0,
      Infinity}, {theta, 0, Pi}, {phi, 0, 2 Pi},
     Method -> {"GlobalAdaptive",
       "SingularityHandler" -> "DuffyCoordinates",
       Method -> "MultiDimensionalRule", MaxErrorIncreases -> 10000},
     PrecisionGoal -> 10, WorkingPrecision -> 30]]], {n, 1, 26, 1}]
```

Appendix C

Data Tables for B^{5+} + H ($n = 1$)

Table C.1: Data for B^{5+} + H(1s)

		impact energies [kev/amu]											
		1.0	5.0	7.0	10.0	25.0	30.0	45.0	90.0	100.0	150.0	200.0	300.0
		total cross sections [10^{-16} cm^2]											
CX		1.97E+01	3.31E+01	3.57E+01	3.68E+01	3.26E+01	3.04E+01	2.29E+01	6.55E+00	4.86E+00	1.27E+00	4.55E-01	7.83E-02
ION		4.34E-04	1.27E-02	3.93E-02	7.24E-02	4.27E-01	9.24E-01	4.20E+00	1.53E+01	1.60E+01	1.57E+01	1.42E+01	1.12E+01
		state resolved CX cross sections [10^{-16} cm^2]											
n	ℓ												
1		9.19E-08	6.42E-08	3.96E-08	2.52E-08	1.89E-08	5.73E-08	2.53E-08	4.37E-07	5.89E-07	2.55E-06	6.37E-06	1.45E-05
1	0	9.19E-08	6.42E-08	3.96E-08	2.52E-08	1.89E-08	5.73E-08	2.53E-08	4.37E-07	5.89E-07	2.55E-06	6.37E-06	1.45E-05
2		1.61E-08	6.04E-04	2.98E-03	1.04E-02	8.83E-02	1.13E-01	1.58E-01	1.54E-01	1.43E-01	7.43E-02	3.57E-02	1.01E-02
2	0	3.55E-09	1.52E-04	7.35E-04	3.87E-03	3.56E-02	4.49E-02	5.95E-02	4.85E-02	4.20E-02	1.33E-02	3.96E-03	5.20E-04
2	1	1.26E-08	4.51E-04	2.24E-03	6.51E-03	5.28E-02	6.85E-02	9.87E-02	1.05E-01	1.01E-01	6.09E-02	3.18E-02	9.58E-03
3		1.36E+00	8.53E+00	9.53E+00	9.96E+00	8.58E+00	7.50E+00	4.74E+00	1.19E+00	8.94E-01	2.55E-01	8.91E-02	1.57E-02
3	0	3.63E-01	1.63E+00	1.37E+00	1.02E+00	4.68E-01	3.55E-01	1.98E-01	5.74E-02	4.52E-02	1.01E-02	2.81E-03	2.99E-04
3	1	7.56E-01	4.04E+00	4.41E+00	4.03E+00	2.33E+00	1.87E+00	9.11E-01	1.13E-01	8.94E-02	4.08E-02	2.10E-02	5.89E-03
3	2	2.40E-01	2.86E+00	3.76E+00	4.91E+00	5.78E+00	5.28E+00	3.63E+00	1.02E+00	7.60E-01	2.04E-01	6.53E-02	9.48E-03
4		1.76E+01	2.25E+01	2.32E+01	2.35E+01	1.46E+01	1.19E+01	6.54E+00	1.37E+00	9.97E-01	2.51E-01	8.82E-02	1.36E-02
4	0	9.15E-01	9.52E-01	7.48E-01	5.17E-01	2.32E-01	1.91E-01	1.36E-01	4.36E-02	3.36E-02	6.35E-03	2.38E-03	2.78E-04
4	1	2.98E+00	3.30E+00	2.97E+00	1.98E+00	1.03E+00	8.83E-01	4.85E-01	9.41E-02	7.75E-02	3.18E-02	2.18E-02	4.69E-03
4	2	5.14E+00	7.40E+00	7.18E+00	6.32E+00	3.09E+00	2.58E+00	1.66E+00	5.69E-01	4.38E-01	1.25E-01	4.30E-02	6.52E-03
4	3	8.52E+00	1.08E+01	1.23E+01	1.47E+01	1.03E+01	8.23E+00	4.26E+00	6.61E-01	4.48E-01	8.79E-02	2.10E-02	2.15E-03
5		7.49E-01	1.83E+00	2.42E+00	2.77E+00	5.68E+00	5.78E+00	4.37E+00	1.07E+00	7.85E-01	1.97E-01	6.61E-02	1.01E-02
5	0	1.13E-01	5.39E-02	5.57E-02	7.13E-02	9.82E-02	9.73E-02	9.30E-02	2.83E-02	2.22E-02	5.07E-03	1.26E-03	2.13E-04
5	1	1.72E-01	1.30E-01	1.68E-01	1.84E-01	3.63E-01	3.71E-01	2.83E-01	7.81E-02	6.39E-02	2.72E-02	1.13E-02	3.07E-03
5	2	2.00E-01	3.43E-01	3.05E-01	2.89E-01	1.07E+00	1.06E+00	8.99E-01	3.48E-01	2.79E-01	8.44E-02	2.85E-02	4.62E-03
5	3	1.82E-01	7.36E-01	9.47E-01	8.70E-01	2.04E+00	2.11E+00	1.61E+00	4.21E-01	2.98E-01	6.12E-02	1.99E-02	1.84E-03
5	4	8.23E-02	5.73E-01	9.41E-01	1.36E+00	2.11E+00	2.14E+00	1.48E+00	1.91E-01	1.21E-01	1.91E-02	5.06E-03	3.43E-04
6		1.97E-02	1.54E-01	2.90E-01	3.34E-01	1.97E+00	2.48E+00	2.60E+00	7.77E-01	5.76E-01	1.40E-01	4.70E-02	7.23E-03
6	0	1.47E-03	1.33E-02	2.03E-02	1.39E-02	3.96E-02	5.03E-02	5.63E-02	2.13E-02	1.53E-02	3.05E-03	8.24E-04	1.98E-04
6	1	2.58E-03	2.22E-02	3.77E-02	3.48E-02	1.59E-01	1.75E-01	1.78E-01	6.62E-02	5.30E-02	2.00E-02	8.52E-03	2.12E-03
6	2	3.36E-03	2.37E-02	2.94E-02	5.15E-02	3.99E-01	4.90E-01	4.98E-01	2.37E-01	1.84E-01	5.10E-02	1.87E-02	3.09E-03
6	3	4.63E-03	4.06E-02	5.12E-02	7.30E-02	5.76E-01	7.31E-01	8.34E-01	2.60E-01	1.97E-01	4.50E-02	1.28E-02	1.42E-03
6	4	4.49E-03	3.43E-02	7.79E-02	5.70E-02	5.24E-01	7.12E-01	7.45E-01	1.55E-01	1.04E-01	1.78E-02	4.30E-03	3.31E-04
6	5	3.16E-03	1.95E-02	7.37E-02	1.04E-01	2.77E-01	3.24E-01	2.86E-01	3.68E-02	2.18E-02	3.23E-03	1.88E-03	7.48E-05
7		2.98E-03	4.51E-02	1.01E-01	1.06E-01	7.95E-01	1.17E+00	1.61E+00	5.79E-01	4.22E-01	1.07E-01	3.66E-02	5.65E-03
7	0	1.60E-04	3.25E-03	1.05E-02	5.58E-03	2.00E-02	2.67E-02	4.22E-02	1.84E-02	1.16E-02	2.63E-03	9.63E-04	2.06E-04
7	1	5.58E-04	6.56E-03	1.75E-02	1.22E-02	7.51E-02	9.80E-02	1.11E-01	5.36E-02	4.40E-02	1.55E-02	8.48E-03	1.76E-03
7	2	3.79E-04	9.18E-03	1.15E-02	1.59E-02	1.72E-01	2.37E-01	3.24E-01	1.81E-01	1.29E-01	3.96E-02	1.31E-02	2.20E-03
7	3	3.20E-04	1.32E-02	1.85E-02	1.59E-02	2.05E-01	3.12E-01	4.65E-01	1.74E-01	1.31E-01	3.12E-02	8.18E-03	1.07E-03
7	4	4.49E-04	5.98E-03	2.02E-02	2.39E-02	1.84E-01	2.95E-01	4.26E-01	1.06E-01	7.88E-02	1.34E-02	3.71E-03	2.85E-04
7	5	5.44E-04	4.28E-03	1.23E-02	1.11E-02	1.14E-01	1.64E-01	1.99E-01	3.88E-02	2.42E-02	3.66E-03	1.61E-03	1.04E-04
7	6	5.70E-04	2.64E-03	1.06E-02	2.10E-02	2.47E-02	3.74E-02	4.39E-02	6.31E-03	3.05E-03	8.70E-04	6.14E-04	3.46E-05

Table continues on next page.

Data for B^{5+} + H(1s) (cont.)

		impact energies [kev/amu]											
		1.0	5.0	7.0	10.0	25.0	30.0	45.0	90.0	100.0	150.0	200.0	300.0
		state resolved CX cross sections [10^{-16} cm^2]											
n	ℓ												
8		5.85E-04	1.92E-02	4.81E-02	5.06E-02	3.80E-01	6.18E-01	1.04E+00	4.55E-01	3.23E-01	7.97E-02	3.05E-02	4.81E-03
8	0	5.07E-05	1.29E-03	4.63E-03	3.07E-03	1.11E-02	1.62E-02	3.02E-02	1.62E-02	1.01E-02	2.11E-03	8.04E-04	1.95E-04
8	1	9.83E-05	2.93E-03	8.16E-03	6.43E-03	4.21E-02	5.86E-02	7.87E-02	4.14E-02	3.73E-02	1.26E-02	5.87E-03	1.52E-03
8	2	7.46E-05	4.13E-03	5.88E-03	6.53E-03	7.93E-02	1.29E-01	2.04E-01	1.44E-01	9.87E-02	2.64E-02	1.12E-02	1.78E-03
8	3	6.05E-05	4.83E-03	8.94E-03	8.42E-03	9.39E-02	1.51E-01	3.00E-01	1.36E-01	9.03E-02	2.30E-02	7.05E-03	8.49E-04
8	4	1.06E-04	2.21E-03	7.87E-03	8.69E-03	8.21E-02	1.46E-01	2.50E-01	7.44E-02	5.89E-02	1.12E-02	2.31E-03	2.61E-04
8	5	9.12E-05	1.78E-03	5.75E-03	6.75E-03	5.25E-02	8.61E-02	1.26E-01	3.19E-02	2.23E-02	2.85E-03	1.10E-03	1.06E-04
8	6	6.32E-05	1.12E-03	4.26E-03	5.02E-03	1.69E-02	2.60E-02	3.75E-02	8.84E-03	4.58E-03	7.53E-04	1.98E-03	7.29E-05
8	7	3.99E-05	8.71E-04	2.61E-03	5.72E-03	2.36E-03	4.85E-03	9.42E-03	1.71E-03	5.36E-04	7.18E-04	1.86E-04	1.73E-05
9		2.16E-04	9.07E-03	2.68E-02	3.16E-02	2.07E-01	3.60E-01	7.26E-01	3.71E-01	2.61E-01	6.46E-02	2.58E-02	4.22E-03
9	0	5.59E-06	4.45E-04	2.20E-03	1.84E-03	7.62E-03	1.11E-02	2.26E-02	1.34E-02	9.85E-03	1.74E-03	5.99E-04	1.69E-04
9	1	2.28E-05	1.20E-03	4.23E-03	3.59E-03	2.35E-02	3.59E-02	5.91E-02	3.13E-02	3.22E-02	9.05E-03	3.92E-03	1.27E-03
9	2	3.33E-05	2.01E-03	3.29E-03	3.40E-03	4.53E-02	7.48E-02	1.34E-01	1.12E-01	8.18E-02	2.19E-02	9.56E-03	1.55E-03
9	3	2.71E-05	2.07E-03	4.02E-03	4.74E-03	4.78E-02	8.53E-02	2.21E-01	1.17E-01	6.69E-02	1.88E-02	6.58E-03	7.31E-04
9	4	1.50E-05	8.40E-04	3.88E-03	4.73E-03	4.35E-02	7.71E-02	1.64E-01	5.89E-02	4.43E-02	8.46E-03	2.16E-03	2.53E-04
9	5	3.77E-05	7.72E-04	3.47E-03	4.82E-03	2.63E-02	5.22E-02	8.51E-02	2.54E-02	1.97E-02	3.04E-03	9.10E-04	1.02E-04
9	6	3.69E-05	6.75E-04	2.92E-03	2.74E-03	9.74E-03	1.80E-02	2.94E-02	9.23E-03	5.37E-03	5.43E-04	8.40E-04	9.18E-05
9	7	2.13E-05	6.05E-04	1.52E-03	3.14E-03	2.51E-03	3.86E-03	6.57E-03	3.43E-03	1.17E-03	4.12E-04	8.15E-04	4.37E-05
9	8	1.61E-05	4.57E-04	1.26E-03	2.64E-03	1.05E-03	1.55E-03	4.59E-03	6.93E-04	1.28E-04	6.55E-04	3.62E-04	7.34E-06
10		1.51E-04	5.09E-03	1.54E-02	2.06E-02	1.37E-01	2.45E-01	5.54E-01	3.12E-01	2.29E-01	5.41E-02	1.98E-02	3.67E-03
10	0	1.06E-05	1.65E-04	9.15E-04	1.13E-03	5.38E-03	7.88E-03	1.92E-02	1.06E-02	1.06E-02	1.91E-03	4.58E-04	1.42E-04
10	1	1.54E-05	5.37E-04	2.13E-03	2.15E-03	1.67E-02	2.83E-02	5.22E-02	2.51E-02	2.97E-02	9.02E-03	3.07E-03	1.02E-03
10	2	1.10E-05	9.95E-04	1.87E-03	2.37E-03	2.76E-02	4.60E-02	9.79E-02	8.31E-02	7.40E-02	1.78E-02	6.56E-03	1.34E-03
10	3	1.45E-05	9.34E-04	2.02E-03	2.95E-03	3.13E-02	5.68E-02	1.62E-01	1.03E-01	5.49E-02	1.41E-02	4.85E-03	6.41E-04
10	4	1.62E-05	4.43E-04	1.96E-03	2.94E-03	2.76E-02	5.06E-02	1.23E-01	5.32E-02	3.42E-02	7.22E-03	1.87E-03	2.36E-04
10	5	1.08E-05	5.51E-04	2.20E-03	3.00E-03	1.68E-02	3.38E-02	6.15E-02	2.14E-02	1.74E-02	2.31E-03	4.97E-04	9.59E-05
10	6	1.08E-05	4.39E-04	1.98E-03	2.09E-03	6.94E-03	1.41E-02	2.45E-02	8.85E-03	5.93E-03	6.01E-04	4.17E-04	9.75E-05
10	7	1.14E-05	4.55E-04	9.55E-04	1.93E-03	2.21E-03	4.44E-03	8.35E-03	4.59E-03	1.81E-03	1.13E-04	1.60E-03	7.01E-05
10	8	1.93E-05	3.43E-04	8.71E-04	1.65E-03	1.04E-03	1.43E-03	4.46E-03	1.89E-03	3.92E-04	4.74E-04	2.94E-04	2.16E-05
10	9	3.13E-05	2.27E-04	4.68E-04	4.10E-04	1.18E-03	1.28E-03	1.54E-03	2.86E-04	3.27E-05	5.76E-04	2.11E-04	2.49E-06
11		8.42E-05	3.03E-03	9.17E-03	2.12E-02	1.47E-01	2.27E-01	5.26E-01	2.78E-01	2.27E-01	4.62E-02	1.61E-02	3.23E-03
11	0	1.93E-06	9.92E-05	4.27E-04	9.28E-04	8.39E-03	1.14E-02	2.24E-02	8.46E-03	1.27E-02	1.66E-03	4.58E-04	1.19E-04
11	1	2.74E-06	2.07E-04	1.13E-03	2.07E-03	2.26E-02	3.12E-02	6.72E-02	2.26E-02	3.10E-02	7.09E-03	2.79E-03	8.13E-04
11	2	6.57E-06	5.02E-04	1.28E-03	2.72E-03	3.22E-02	4.86E-02	9.24E-02	6.14E-02	7.48E-02	1.47E-02	4.68E-03	1.17E-03
11	3	5.37E-06	4.46E-04	1.13E-03	3.32E-03	3.32E-02	4.86E-02	1.44E-01	9.16E-02	5.21E-02	1.26E-02	3.61E-03	5.86E-04
11	4	8.76E-06	3.20E-04	9.68E-04	3.14E-03	2.52E-02	4.14E-02	1.12E-01	5.32E-02	2.87E-02	6.14E-03	1.75E-03	2.25E-04
11	5	6.53E-06	3.27E-04	1.16E-03	2.72E-03	1.50E-02	2.70E-02	5.29E-02	2.05E-02	1.66E-02	2.19E-03	5.51E-04	9.21E-05
11	6	5.70E-06	2.79E-04	1.12E-03	2.16E-03	6.14E-03	1.12E-02	2.19E-02	9.21E-03	7.09E-03	5.54E-04	2.48E-04	8.98E-05
11	7	8.00E-06	3.06E-04	6.78E-04	1.77E-03	2.69E-03	4.17E-03	6.71E-03	5.91E-03	2.70E-03	2.74E-04	7.13E-04	8.24E-05
11	8	1.03E-05	2.70E-04	5.91E-04	1.34E-03	7.63E-04	1.05E-03	1.86E-03	3.55E-03	8.05E-04	4.83E-05	6.05E-04	4.06E-05
11	9	1.29E-05	1.84E-04	5.05E-04	6.19E-04	3.40E-04	9.76E-04	4.14E-03	1.05E-03	1.22E-04	4.85E-04	7.01E-04	1.01E-05
11	10	1.55E-05	9.33E-05	1.77E-04	4.70E-04	8.89E-04	1.09E-03	4.10E-04	1.16E-04	7.72E-06	4.48E-04	3.16E-05	7.76E-07

Appendix D

Acronyms

AOCC atomic-orbital close-coupling

CX charge exchange

CXS charge exchange spectroscopy

ETF electron translational factors

ION ionization

JET Joint European Torus

MOCC molecular-orbital close-coupling

VSC-2 Vienna Scientific Cluster-2

ONC one-center coupling

TWC two-center coupling

Appendix E

Acknowledgements

First and foremost, I feel very privileged to have had Friedrich Aumayr as supervisor. I want to thank him for giving me the opportunity to work with him and for being available whenever I needed support. I really enjoyed the friendly, cooperative and respectful atmosphere within his working group. His door was always open for anybody having a problem and he was always ready to help find solutions. He also gave me the opportunity to participate in a summer school on plasma physics in Garching and to contribute a poster to an international conference.

I am extremely thankful to my co-supervisor Katharina Igenbergs, who supervised this thesis during her free time in addition to her normal job. We usually met once a week at night, after she had had an entire day of work. Still, she was always in a good mood and it was a lot of fun to sit in front of the screen together with her, wondering about strange behavior of the code or weird results and eating carrots and "Kornspitz with Liptauer". I admire her more or less subtle geekiness together with her profound knowledge of both physics and computers. I am really glad to have had her guidance during this entire time. This work would not have been possible without her.

I want to acknowledge and thank Markus Wallerberger for all his help during the entire project. In addition to writing his PhD thesis, he often took the time to discuss problems or results with me during lunch or over a beer at night. His bright mind and his incredible knowledge of theoretical physics are truly impressive and I am very thankful for all his input.

The computational results presented in this work have been achieved using the Vienna Scientific Cluster (VSC-2) and I therefore want to thank the entire team of the VSC for their continuing support during my calculations. They were extraordinarily friendly, really quick at answering questions per mail and very helpful.

I also want to thank Christoph Langer who, together with Club Mate, helped me through an entire night of debugging. Thank you for your time and effort, Christoph!

It is difficult to put in a few words, how much of my academic degree I owe to Tina and Sarah. Thank you both for your patience and all your help during our entire studies.

I want to thank Niki for all his input and our fruitful discussions while drinking coffee or

during lunch. It felt good to know that most of the times I was not the last person still in the "Freihaus" after midnight. You are not only a good friend but also a well-respected colleague.

Simone, thank you for accepting my frequent mental absence when weird results kept me puzzled even after I got home. You were an essential pillar for me during this entire thesis.

Last but not least, I can not even begin to explain how much I owe to my parents and how thankful I am for all that they have given me. Still, I want to emphasize my gratitude for their continuing support throughout my studies and for giving me the opportunity to spend several months in Sweden. This would not have been possible without their financial support. This work is dedicated to them.

List of Figures

List of Tables

Bibliography

[1] J. D. Lawson, "Some Criteria for a Power Producing Thermonuclear Reactor", Proceedings of the Physical Society. Section B 70 (1) (1957) 6.
URL http://stacks.iop.org/0370-1301/70/i=1/a=303

[2] BBC, "EU member states agree on ITER funding shortfall" (July 2010).
URL http://www.bbc.co.uk/news/10614542

[3] A. W. Kleyn, N. J. Lopes Cardozo, U. Samm, "Plasma-surface interaction in the context of ITER", Phys. Chem. Chem. Phys. 8 (2006) 1761–1774.
URL http://dx.doi.org/10.1039/B514367E

[4] A.M. Messiaen, J. Ongena et al., "Improved confinement with edge radiative cooling at high densities and high heating power in TEXTOR", Nuclear Fusion 34 (6) (1994) 825.
URL http://stacks.iop.org/0029-5515/34/i=6/a=I06

[5] R. C. Isler, "An overview of charge-exchange spectroscopy as a plasma diagnostic", Plasma Physics and Controlled Fusion 36 (2) (1994) 171.
URL http://stacks.iop.org/0741-3335/36/i=2/a=001

[6] D. R. Bates, R. McCarroll, "Electron Capture in Slow Collisions", Proc. Royal Soc. 245 (1958) 175–183.

[7] D R Bates, "Electron Capture in Fast Collisions", Proc. Royal Soc. 247 (1958) 294–301.

[8] K. Igenbergs, "Calculation of Cross Sections Relevant for Diagnostics of Hot Fusion Plasma", Ph.D. thesis, Technische Universität Wien (September 2011).

[9] B. H. Bransden, M. R. C. McDowell, "Charge Exchange and the Theory of Ion-Atom Collisions", Clarendon Press - Oxford, 1992.

[10] Wolfgang Fritsch and C.D Lin, "The semiclassical close-coupling description of atomic collisions: Recent developments and results", Physics Reports 202 (1-2) (1991) 1–97.
URL http://www.sciencedirect.com/science/article/pii/037015739190008A

[11] M. Wallerberger, "Effective calculations of fusion-relevant highly charged ion collisions", Master Thesis, Vienna University of Technology, 2011.

[12] Messiah, Albert, "Quantenmechanik, Band 1", 4. Auflage Edition, Walter de Gruyter, 1981.

[13] T. A. Green, "A proof of detailed balancing for the impact parameter method", Proceedings of the Physical Society 86 (5) (1965) 1017.
URL http://stacks.iop.org/0370-1328/86/i=5/a=312

[14] R Shakeshaft, "A note on the exchange integrals in the impact-parameter treatment of heavy-particle collisions", Journal of Physics B: Atomic and Molecular Physics 8 (8) (1975) L134.
URL `http://stacks.iop.org/0022-3700/8/i=8/a=003`

[15] M. Wallerberger, K. Igenbergs, J. Schweinzer, F. Aumayr, "Fast computation of close-coupling exchange integrals using polynomials in a tree representation", Comp. Phys. Comm. 182 (3) (2011) 775 – 778.

[16] Dahmen Wolfgang, Reusken Arnold, "Numerik für Ingenieure und Naturwissenschaftler", Springer-Lehrbuch, 2., korr. Aufl. 2008, XVIII.

[17] William H. Press, Brian P. Flannery, Saul A. Teukolsky and Williams T. Vetterling, "Numerical Recipes - The Art of Scientific Computing", Cambridge University Press, Cambridge, 1986.

[18] J. Schweinzer, "Ein- und Zweielektronenprozesse beim Einelektroneneinfag von Alkaliatomen in doppelt geladene Edelgasionen", Ph.D. thesis, Technische Universität Wien (January 1990).

[19] C. Illescas, A. Riera, "Classical study of single-electron capture and ionization processes in $A^{q+} + (\mathrm{H}, \mathrm{H}_2)$ collisions", Phys. Rev. A 60 (1999) 4546–4560.
URL `http://link.aps.org/doi/10.1103/PhysRevA.60.4546`

[20] H. F. Busnengo, S. E. Corchs, A. E. Martnez, R. D. Rivarola, "Single electron capture by impact of multicharged ions", Physica Scripta 1997 (T73) (1997) 242.
URL `http://stacks.iop.org/1402-4896/1997/i=T73/a=078`

[21] J. P. Hansen, A. Dubois, "Cross sections for electron capture in H-B^{q+} and He-B^{q+} (q=1,3,5) collisions", Physica Scripta 1996 (T62) (1996) 55.
URL `http://stacks.iop.org/1402-4896/1996/i=T62/a=009`

[22] W. Fritsch, C. D. Lin, "Atomic-orbital-expansion studies of electron transfer in bare-nucleus $Z(Z = 2, 4 - 8)$ - hydrogen-atom collisions", Phys. Rev. A 29 (1984) 3039–3051.
URL `http://link.aps.org/doi/10.1103/PhysRevA.29.3039`

[23] N. Toshima, "Ionization and charge transfer of atomic hydrogen in collision with multiply charged ions", Phys. Rev. A 50 (1994) 3940–3947.
URL `http://link.aps.org/doi/10.1103/PhysRevA.50.3940`

[24] C. Harel, H. Jouin, B. Pons, "Cross sections for electron capture from atomic Hydrogen by fully stripped ions on the 0.05 - 1.00 a.u. impact velocity range", Atomic Data and Nuclear Data Tables 68 (2) (1998) 279 – 302.
URL `http://www.sciencedirect.com/science/article/pii/S0092640X97907683`

[25] L. F. Errea, J. D. Gorfinkiel, C. Harel, H. Jouin, A. Macas, L. Mndez, B. Pons, A. Riera, "Total and partial cross-sections of electron transfer processes with hydrogen gas targets: Be^{4+} , B^{5+} + H(1s), H(2s)", Physica Scripta 1996 (T62) (1996) 27.
URL `http://stacks.iop.org/1402-4896/1996/i=T62/a=004`

[26] H. J. Ludde, R. M. Dreizler, "Electron capture with He^{2+} , Li^{3+} , Be^{4+} and B^{5+} projectiles from atomic hydrogen", Journal of Physics B: Atomic and Molecular Physics 15 (16) (1982) 2713.
URL http://stacks.iop.org/0022-3700/15/i=16/a=019

[27] H. Suzuki, N. Toshima, T. Watanabe, H. Ryufuku, "Exponential distorted-wave approximation for charge transfer in collisions of multicharged ions with atomic hydrogen", Phys. Rev. A 29 (1984) 529–535.
URL http://link.aps.org/doi/10.1103/PhysRevA.29.529

[28] D. H. Crandall, R. A. Phaneuf, F. W. Meyer, "Electron capture by slow multicharged ions in atomic and molecular hydrogen", Phys. Rev. A 19 (1979) 504–514.
URL http://link.aps.org/doi/10.1103/PhysRevA.19.504

[29] K. Igenbergs, J. Schweinzer, A. Veiter, L. Perneczky, E. Frühwirth, M. Wallerberger, R. E. Olson, F. Aumayr, "Charge exchange and ionization in N^{7+}, N^{6+}, C^{6+} - H(n=1,2) collisions studied systematically by theoretical approaches", Journal of Physics B: Atomic, Molecular and Optical Physics 45 (6) (2012) 065203.
URL http://stacks.iop.org/0953-4075/45/i=6/a=065203

[30] J. Burgdorfer, R. Morgenstern, A. Niehaus, "Angular momentum distribution in the classical over-barrier model for electron capture into highly charged slow projectiles", Journal of Physics B: Atomic and Molecular Physics 19 (14) (1986) L507.
URL http://stacks.iop.org/0022-3700/19/i=14/a=001

[31] H. Ryufuku, K. Sasaki, T. Watanabe, "Oscillatory behavior of charge transfer cross sections as a function of the charge of projectiles in low-energy collisions", Phys. Rev A 21 (1980) 745.

[32] L. Landau, E. M. Lifschitz, "Quantum Mechanics - Non-Relativistic Theory", second, revised and updated Edition, Pergamon Press, 1965.

[33] Demtroeder, Wolfgang, "Experimentalphysik 3", 4. Auflage Edition, Springer, 2010.