Appendix A

Numerical and Algorithmic Details

In this appendix, the computer code is described which was used to produce the numerical results presented in this thesis. It also comprises the details of the computational methods and explicitly some formulas which have been omitted in the main part of this work. Furthermore, improvements of the code, which have not been implemented yet, are proposed. The main aim of such changes must be the reduction of the compute time, thereby making larger bases feasible.

The present program is the first of its kind which is capable of performing numerical computations in various different relativistic frames of reference. In *any* previously existing computer code that numerically solves the relativistic coupled channel equations a particular frame of reference was chosen, namely the reference frame where the initial electronic state was at rest. Moving initial configurations have not been considered. Not only coupled channel codes, but also other numerical approaches to solve the two-centre Dirac equation (like the momentum space approach, finite element and finite difference calculations) have only considered initial configurations at rest in the frame of the computation.

For coding the C/C++ programming languages have been used predominantly [Str97, CSC+97]. For the convenience of readers who are interested in reading the source code (although not reproduced here), we quote identifiers occasionally.

A.1 General definitions

At the time of writing, the program is capable of solving the coupled channel equations in those frames of reference where both nuclei, A and B, move along straight line trajectories parallel to the $\boldsymbol{e}_3$-axis of the coordinate system. The impact parameter plane is chosen to be the $\boldsymbol{e}_1$-$\boldsymbol{e}_3$-plane and the time of closest approach of the nuclei is $t = 0$. In the subsequent discussion the trajectories are assumed to be of the form,

$$\boldsymbol{R}_{\mathrm{A}}(t) = -\frac{b}{2}\,\boldsymbol{e}_1 + tv_{\mathrm{A}}\,\boldsymbol{e}_3, \qquad \boldsymbol{R}_{\mathrm{B}}(t) = \frac{b}{2}\,\boldsymbol{e}_1 + tv_{\mathrm{B}}\,\boldsymbol{e}_3, \tag{A.1}$$

with

$$-\infty < b < \infty, \qquad -1 < v_\Gamma < 1, \text{ with } \Gamma = \mathrm{A,B}, \qquad \text{and} \qquad -\infty < t < \infty.$$

However, recall that any translation of the origin of the spatial coordinate system in the $\boldsymbol{e}_1$-$\boldsymbol{e}_2$-plane will yield an identical set of coupled equations (cf. section 5.1). The input parameters of the computer program that determine these trajectories are listed in table A.1. In terms of the rapidities $\chi^{\mathrm{c}}_{\mathrm{B}}$ and $\chi^{\mathrm{c}}_{\mathrm{frame}}$ the velocities v_{A} and v_{B} are determined according to:

$$\begin{aligned} v_{\mathrm{A}} &= \tanh(-\chi^{\mathrm{c}}_{\mathrm{B}} - \chi^{\mathrm{c}}_{\mathrm{frame}}), \\ v_{\mathrm{B}} &= \tanh(\chi^{\mathrm{c}}_{\mathrm{B}} - \chi^{\mathrm{c}}_{\mathrm{frame}}). \end{aligned}$$

TABLE A.1. Command line parameters of the main program.

`chi`	Rapidity $\chi_{\mathrm{B}}^{\mathrm{c}}$ of nucleus B in the collider frame.
`chiframe`	Rapidity $\chi_{\mathrm{frame}}^{\mathrm{c}}$ of the frame of reference with respect to the collider frame.
`b`	Impact parameter b in relativistic units.
`zA zB`	Charge number of projectiles A and B respectively.
`nmaxA nmaxB`	Maximum principal quantum numbers of bound-state basis functions of centre A and B respectively.
`kappaA kappaB`	Maximum absolute values of the spin-orbit quantum numbers κ of free-particle basis functions.
`ti tf dt tinnr dtinnr`	Variables $t_{\mathrm{f}}, t_{\mathrm{i}}, \Delta_t, t_{\mathrm{inner}}$ and $\Delta_{t,\mathrm{inner}}$ that determine the time grid (cf. section A.4), in relativistic units.
`infile`	Name of a checkpoint file. Necessary for the continuation of a previously interrupted computation.
`id`	Identification string of a coupled channel calculation.

The Lorentz factor γ specifying the collision energy (i.e. the kinetic energy of one of the projectiles measured in the rest frame of the other projectile) is then given in terms of the rapidity $\chi_{\mathrm{B}}^{\mathrm{c}}$ by:

$$\gamma = \cosh(2\chi_{\mathrm{B}}^{\mathrm{c}}).$$

Half the distance between the centres A and B at time t in the unprimed frame, denoted by f in the following, is equal to:

$$f = \frac{1}{2}\sqrt{b^2 + t^2(v_{\mathrm{B}} - v_{\mathrm{A}})^2} \tag{A.2}$$

In the present context, primed rest-frame coordinates (t', x', y', z') of centre A and doubly primed rest-frame coordinates (t'', x'', y'', z'') of centre B are defined as,

$$t' = \gamma_{\mathrm{A}}(t - v_{\mathrm{A}}z), \qquad x' = x + \frac{b}{2}, \qquad y' = y, \qquad z' = \gamma_{\mathrm{A}}(z - v_{\mathrm{A}}t) \tag{A.3}$$

$$t'' = \gamma_{\mathrm{B}}(t - v_{\mathrm{B}}z), \qquad x'' = x - \frac{b}{2}, \qquad y'' = y, \qquad z'' = \gamma_{\mathrm{B}}(z - v_{\mathrm{B}}t). \tag{A.4}$$

These definitions are emphasised, because the definitions of the basis functions below explicitly refer to the direction of the z'- and z''-axis. Note that the directions have been chosen to be equal to the direction of the z-axis of the computational frame of reference. In the program radial coordinates $(r_{\mathrm{A}}, \vartheta_{\mathrm{A}}, \varphi_{\mathrm{A}})$ for the rest frame of A are

defined in terms of these Cartesian coordinates according to, [1]

$$r_{\mathrm{A}} = \sqrt{x'^2 + y'^2 + z'^2}, \qquad \cos\vartheta_{\mathrm{A}} = \frac{z'}{r_{\mathrm{A}}}, \qquad \varphi_{\mathrm{A}} = \arg\left(x' + \mathrm{i}y'\right), \tag{A.5}$$

The spatial radial coordinates $(r_{\mathrm{B}}, \vartheta_{\mathrm{B}}, \varphi_{\mathrm{B}})$ in the rest frame of B are given similarly in terms of the doubly primed Cartesian coordinates (x'', y'', z'').

A.2 Basis functions

As described in sections 5.2 and 5.3, in the present impementation of the coupled channel method the basis functions $\Phi_{\Gamma,i}(t, \boldsymbol{x})$ are eigenfunctions of the spin-orbit operator and the third component of angular momentum in their respective rest frames. In this section we give the precise forms of the eigenfunctions $\phi_{\mathrm{A},i}(\boldsymbol{x}')$ and $\phi_{\mathrm{B},i}(\boldsymbol{x}'')$, referred to in sections 5.3 and 5.5. The following presentation covers the eigenfunctions $\phi_{\mathrm{A},i}(\boldsymbol{x}')$ referring to the rest frame of centre A. Everything stated in the following applies in an analogous way to eigenfunctions $\phi_{\mathrm{B},i}(\boldsymbol{x}'')$ in the rest frame of centre B.

A.2.1 Spin-angular functions. Numerical calculations make use of the standard Dirac-Pauli representation of the Dirac matrices (see equation (C.2) of appendix C). For this representation, the spin-orbit operator K' defined in equation (5.5) is block-diagonal:

$$K' = \begin{pmatrix} -\frac{1}{2}\boldsymbol{\sigma}\cdot\boldsymbol{L}' - 1 & 0 \\ 0 & \frac{1}{2}\boldsymbol{\sigma}\cdot\boldsymbol{L}' + 1 \end{pmatrix}.$$

In this case, the third component ${J'}^3$ of the angular momentum operator $\boldsymbol{J}'$ is a diagonal block-matrix as well:

$${J'}^3 = -\mathrm{i}\,\frac{\partial}{\partial\varphi_{\mathrm{A}}} + \frac{1}{2}\begin{pmatrix} \sigma_3 & 0 \\ 0 & -\sigma_3 \end{pmatrix}.$$

Both K' and ${J'}^3$ only contain differentiation operators with respect to the angular variables ϑ_{A} and φ_{A}. Therefore, a simultaneous eigenstate $\phi_{\mathrm{A}}(\boldsymbol{x}')$ of K' and ${J'}^3$, with eigenvalues κ and m respectively, is of the form:

$$\phi_{\mathrm{A}}(\boldsymbol{x}') = \frac{1}{r_{\mathrm{A}}}\begin{pmatrix} \mathrm{i}P(r_{\mathrm{A}})\;\chi_\kappa^m(\vartheta_{\mathrm{A}}, \varphi_{\mathrm{A}}) \\ Q(r_{\mathrm{A}})\;\chi_{-\kappa}^m(\vartheta_{\mathrm{A}}, \varphi_{\mathrm{A}}) \end{pmatrix}, \tag{A.6}$$

with complex-valued radial functions $P(r_{\mathrm{A}})$ and $Q(r_{\mathrm{A}})$. The imaginary constant has been added for later convenience. The two-spinors $\chi_\kappa^m(\vartheta, \varphi)$ must satisfy the following eigenvalue equations:

$$\begin{aligned} \left[\tfrac{1}{2}\boldsymbol{\sigma}\cdot\boldsymbol{L}' + 1\right]\chi_\kappa^m(\vartheta,\varphi) &= -\kappa\,\chi_\kappa^m(\vartheta,\varphi), \\ \left[-\mathrm{i}\partial_{\varphi_{\mathrm{A}}} + \tfrac{1}{2}\sigma_3\right]\chi_\kappa^m(\vartheta,\varphi) &= m\,\chi_\kappa^m(\vartheta,\varphi). \end{aligned}$$

[1] This notation represents a minor inconsistency with previous notation: In this appendix r_{A} is not primed, although it is identical to the Lorentz scalar $r'_{\mathrm{A}}(t', \boldsymbol{x}')$ as defined in section 2.1.

A suitable choice for these spin-angular functions $\chi_\kappa^m(\vartheta,\varphi)$ is given by the following definition: [2]

$$\chi_\kappa^m(\vartheta,\varphi) = \frac{1}{\sqrt{2l+1}} \begin{pmatrix} -\sqrt{l-m+\frac{1}{2}}\, Y_l^{m-\frac{1}{2}}(\vartheta,\varphi) \\ \sqrt{l+m+\frac{1}{2}}\, Y_l^{m+\frac{1}{2}}(\vartheta,\varphi) \end{pmatrix} \qquad \text{for } \kappa > 0, \text{with } l = \kappa,$$

$$\chi_\kappa^m(\vartheta,\varphi) = \frac{1}{\sqrt{2l+1}} \begin{pmatrix} \sqrt{l+m+\frac{1}{2}}\, Y_l^{m-\frac{1}{2}}(\vartheta,\varphi) \\ \sqrt{l-m+\frac{1}{2}}\, Y_l^{m+\frac{1}{2}}(\vartheta,\varphi) \end{pmatrix} \qquad \text{for } \kappa < 0, \text{with } l = |\kappa| - 1.$$

In this work we refer to the following phase convention for spherical harmonic functions $Y_l^p(\vartheta,\varphi)$ and associated Legendre polynomials $P_l^p(\cos\vartheta)$:

$$\begin{aligned} Y_l^p(\vartheta,\varphi) &= \mathrm{e}^{\mathrm{i}p\varphi} \sqrt{\frac{2l+1}{4\pi}\frac{(l-p)!}{(l+p)!}}\, \underbrace{\frac{(-1)^{l+p}}{2^l\, l!}\, (\sin\vartheta)^p \left[\frac{\mathrm{d}}{\mathrm{d}(\cos\vartheta)}\right]^{l+p} (\sin\vartheta)^{2l}} \\ &= \mathrm{e}^{\mathrm{i}p\varphi} \sqrt{\frac{2l+1}{4\pi}\frac{(l-p)!}{(l+p)!}} \qquad\qquad\qquad\qquad P_l^p(\cos\vartheta) \end{aligned}$$

The spin-angular functions $\chi_\kappa^m(\vartheta,\varphi)$ constitute a complete set of orthonormal functions on the unit sphere. For the numerical evaluation of the spherical harmonics the formula,

$$Y_l^p(\vartheta,\varphi) = \mathrm{e}^{\mathrm{i}p\varphi} \sqrt{\frac{2l+1}{4\pi}\frac{(l-|p|)!}{(l+|p|)!}}\, P_l^{|p|}(\cos\vartheta) \times \begin{cases} (-1)^p & \text{if } p < 0, \\ 1 & \text{if } p > 0, \end{cases}$$

is most suitable, because numerically stable recursion relations exist, which allow for the determination of associated Legendre polynomials P_l^p of positive order p [PTVF92]. Owing to the presently adopted phase conventions, the spin-angular functions $\chi_\kappa^m(\vartheta,\varphi)$ also satisfy the relation [Ros61, eq. (1.65')],

$$\boldsymbol{e}_r \cdot \boldsymbol{\sigma}\, \chi_\kappa^m(\vartheta,\varphi) = -\chi_{-\kappa}^m(\vartheta,\varphi), \tag{A.7}$$

with $\boldsymbol{e}_r = (\cos\varphi\sin\vartheta, \sin\varphi\sin\vartheta, \cos\vartheta)$.

A.2.2 Radial Dirac equation. As explained in chapter 5, basis functions attributed to centre A are constructed either from eigenstates of the channel Hamiltonian $H'_{\mathrm{A}} = -\mathrm{i}\boldsymbol{\alpha}\cdot\nabla' + \beta - eV_{\mathrm{A}}(r_{\mathrm{A}})$ (for bound states) or as superpositions of continuum eigenfunctions of H'_{A} (in the case of wave packets). A simultaneous eigenfunction

[2] Spin-angular functions $\chi_\kappa^m(\vartheta,\varphi)$ are also known as central field spinors [Ros61], spinor spherical harmonics [Sch95], or spherical spinors [SFVW95b]. In the literature *many* different phase conventions for spin-angular functions, spherical harmonics, and associated Legendre polynomials can be found. For example, the present choice is in agreement with [AS65, PTVF92, Jac99] regarding associated Legendre polynomials (but disagreeing with [Edm57]), in agreement with [Edm57, Ros61, Dav65, PTVF92, Jac99] regarding spherical harmonics (but disagreeing with [Sch55], [BS77], as well as [LL86, BLP82]), and finally in agreement with [Ros61] regarding the Clebsch-Gordon coefficients which determine the spherical spinors. In consequence of different phase conventions, equation (A.7) does not hold for spin-angular functions as defined, e.g., in [BD66] or [BLP82]. However, similar equations are given by the latter authors. In the present work, the numerical algorithm of [PTVF92, sec. 6.8] is used to compute associated Legendre functions, in combination with the definition of spherical spinors $\chi_\kappa^m(\vartheta,\varphi)$ as in [Ros61, eq. (1.60')].

$\phi_{\mathrm{A}}(\boldsymbol{x}')$ of the operators H'_{A}, K' and ${J'}^3$, with eigenvalues ϵ, κ and m respectively, is of the form (A.6). Owing to equation (A.7) it can be verified that the eigenvalue equation,

$$[H'_{\mathrm{A}} - \epsilon]\ \phi_{\mathrm{A}}(\boldsymbol{x}') = 0,$$

is equivalent to the following radial Dirac equation for the radial wave functions $P(r_{\mathrm{A}})$ and $Q(r_{\mathrm{A}})$:

$$\frac{\mathrm{d}}{\mathrm{d}r_{\mathrm{A}}} \begin{pmatrix} P(r_{\mathrm{A}}) \\ Q(r_{\mathrm{A}}) \end{pmatrix} = \begin{pmatrix} -\dfrac{\kappa}{r_{\mathrm{A}}} & -\epsilon - eV_{\mathrm{A}}(r_{\mathrm{A}}) - 1 \\ \epsilon + eV_{\mathrm{A}}(r_{\mathrm{A}}) - 1 & \dfrac{\kappa}{r_{\mathrm{A}}} \end{pmatrix} \begin{pmatrix} P(r_{\mathrm{A}}) \\ Q(r_{\mathrm{A}}) \end{pmatrix}. \qquad \text{(A.8)}$$

Note that the imaginary factor in equation (A.6) leads to the present form (A.8) of the radial Dirac equation, which allows for real-valued solutions $P(r_{\mathrm{A}})$ and $Q(r_{\mathrm{A}})$.

An algorithm for an accurate numerical solution of the radial differential equation (A.8) has been published by Salvat et al. [SM91, SFVW95A, SFVW95B]. It assumes that a singularity of the radial potential $V_{\mathrm{A}}(r)$ at $r = 0$ is at most Coulomb-like, more precisely,

$$\lim_{r\to 0} V_{\mathrm{A}}(r) r < \infty,$$

is assumed. Furthermore, it is assumed that the potential $V_{\mathrm{A}}(r)$ vanishes as r tends to infinity and that the limit

$$\lim_{r\to\infty} V_{\mathrm{A}}(r) r = \tilde{Z}_{\mathrm{A}}$$

exists. The algorithm of Salvat et al. allows for the computation of the radial wave functions of normalised bound states and determines their eigenvalues at the same time. Furthermore, radial wave functions corresponding to continuum eigenfunctions with positive energy $\epsilon > 1$ can be determined. The latter radial functions are normalised by the code of Savat et al. such that the upper component $P(r)$ is oscillating with unit amplitude as $r \to \infty$. Only *regular* solutions of the radial Dirac equation are computed, which are distinguished by their property of square integrability at the boundary $r = 0$. For this work the code of Salvat et al. has been ported to the C programming language and extended to allow for the computation of radial wave functions of negative energy $\epsilon < -1$.

The normalisation of continuum eigenfunctions $\phi_{\mathrm{A},\epsilon}(\boldsymbol{x}')$ on the energy scale is obtained by a multiplication of their radial wave functions $P(r_{\mathrm{A}})$ and $Q(r_{\mathrm{A}})$ by the factor,

$$\frac{1}{\sqrt{\pi}} \left(\frac{\epsilon+1}{\epsilon-1}\right)^{1/4}.$$

That normalisation of continuum wave functions provides that radial wave packets of the form,

$$\frac{1}{\sqrt{\Delta_\epsilon}} \int_{\epsilon-\Delta_\epsilon/2}^{\epsilon+\Delta_\epsilon/2} \phi_{\mathrm{A},\epsilon}(\boldsymbol{x}')\, \mathrm{d}\epsilon,$$

are normalised in the primed frame.

For the numerical calculations presented in this work the Coulomb potential $V_\Gamma(r_\Gamma) = eZ_\Gamma/r_\Gamma$ has been considered for both centres $\Gamma = \mathrm{A}, \mathrm{B}$. However, due

to the fact that the radial wave functions of the basis functions are determined by a numerical integration of the radial Dirac equation (A.8), it is straightforward to extend the existing code to allow for a numerical coupled channel solution of a two-centre Dirac equation with more general moving charge distributions. This possibility has motivated the preference of the present determination of the radial wave functions over other numerical methods to compute Coulomb-Dirac wave functions or the related Whittaker functions [AS65, MRG73, GMR85, CEL].

A.2.3 Implementation details. For bound states, the radial wave functions are computed at the beginning of a coupled channel calculation and tabulated for a few thousand radii for later linear interpolation. The size of the radial grid in relativistic units was chosen according to $r_{\max} = (2n^2 + Cn^{1.2})/(e^2Z)$, with a constant C taking a value between 15 and 20. Here Z is the charge number of the respective centre and n the principal quantum number of the bound state. Wave functions of wave packets are integrated using a Gaussian quadrature formula for the energy integration and tabulated in the same way as bound state radial wave functions, to allow for the later evaluation of the radial functions by linear interpolation. For the wave packets as described in section 6.6 a radial grid of size 200 r.u. was used. To check the code, radial wave functions obtained by the present program have been compared to corresponding plots published in [BS85].

For numerical calculations, a finite coupled channel basis has to be specified. In the present program this can be done partly by command line parameters, denoted by `nmaxA`, `nmaxB`, `kappaA` and `kappaB` (cf. table A.1). The first two parameters determine the maximum principal quantum number of bound-state basis functions of centres A and B respectively. The second two parameters fix the maximum absolute values of the spin-orbit quantum numbers κ of free-particle basis functions. The mean energies $\bar{\epsilon}$ and the widths Δ_ϵ of the energy interval of wave packets cannot be chosen on the command line presently. In the source code tabulated radial Dirac wave functions attributed to the same centre are subsumed by an object of the class `DiracRadialBasis`. It is the constructor of this class that evaluates radial wave functions according to the algorithm of Salvat et al..

A.3 Quadrature formulas

The three-dimensional integrals presented in section 5.3 have to be evaluated fully numerically. This evaluation takes the major part of the computing time and is, hence, the reason for the numerical complexity of the computational task of solving the relativistic coupled channel equations (4.8). Efficient quadrature formulas for these integrals will take into account the distance between the centres as well as the fact whether the two states occurring in the integrand are located at the same or a at different centres. Therefore, two different quadrature schemes are used in the program. Both procedures introduce some curved spatial coordinates in the computational, unprimed frame of reference. The three-volume integrals (5.10), (5.11), (5.15) and (5.16) are then rewritten as a sequence of three nested one-dimensional integrals over these curved coordinates. The latter are evaluated successively by means

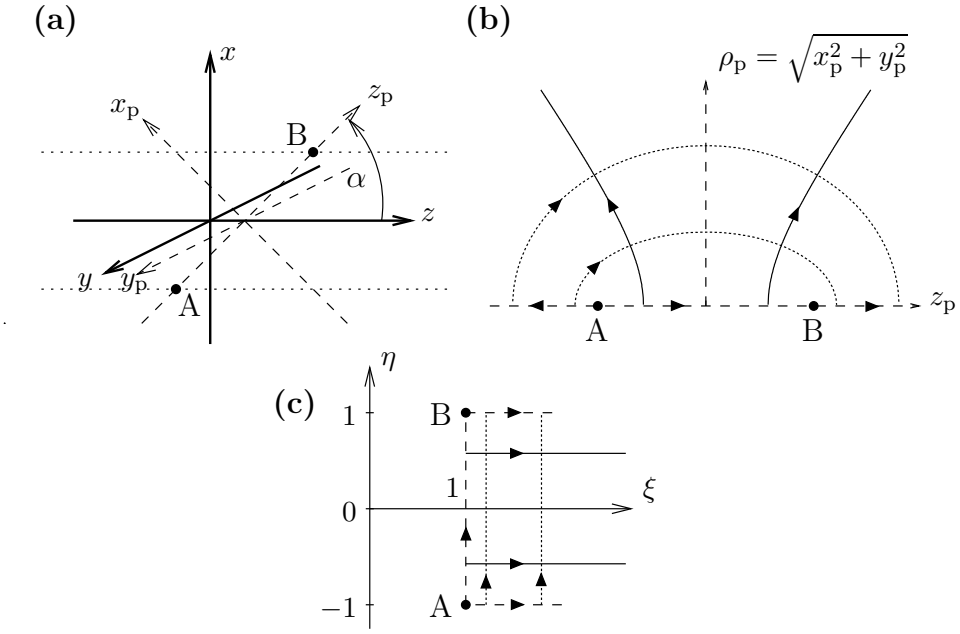

FIGURE A.1. Prolate spheroidal coordinates. (a) The centres A and B are moving in the x-z-plane. The straight line going through the centres, from A to B, defines the z_p-coordinate axis. (b) A half-plane is rotated around the z_p-axis. (c) The half-plane is parameterised by elliptical coordinates.

of one-dimensional quadrature formulas. The first quadrature method described below is based on prolate spheroidal coordinates, the second on contracted spherical coordinates around one of the centres.

A.3.1 Quadrature using prolate spheroidal coordinates. If the distance of the centres A and B is small compared to the extension of the integrand and in the case of scalar products between states located at different centres, a quadrature method based on prolate spheroidal coordinates is employed. These coordinates are elliptical coordinates of the half-plane which are rotated into the third spatial dimension in order to obtain coordinates of the whole three-dimensional space [AS65, MS88]. Their definition in the present context is depicted in figure A.1.

Elliptical coordinates are rotated around the z_p-axis which is defined as the coordinate axis passing through the centres from A to B at some fixed time t. Let $\rho_\mathrm{p} = \sqrt{x_\mathrm{p}^2 + y_\mathrm{p}^2}$ denote the distance of some point (x, y, z) from the z_p-axis. Elliptical coordinates (ξ, η) of the $(z_\mathrm{p}, \rho_\mathrm{p})$-half-plane can be defined in terms of a conformal mapping, namely the principal branch of the complex arcsin-function [AS65, FL92], as follows:

$$\begin{aligned} \xi &= \cosh\left(\Im \arcsin \frac{z_\mathrm{p} + \mathrm{i}\rho_\mathrm{p}}{f}\right), \\ \eta &= \sin\left(\Re \arcsin \frac{z_\mathrm{p} + \mathrm{i}\rho_\mathrm{p}}{f}\right). \end{aligned}$$

By this mapping the $(z_\mathrm{p}, \rho_\mathrm{p})$-half-plane is mapped one to one and onto the strip,

$$1 \le \xi < \infty \qquad \text{and} \qquad -1 \le \eta \le 1.$$

The inverse mapping is simply given by:

$$z_\mathrm{p} = f\xi\eta, \qquad \rho_\mathrm{p} = f\sqrt{(\xi^2-1)(1-\eta^2)}. \tag{A.9}$$

Therefore, while passing over from Cartesian coordinates (x, y, z) to prolate spheroidal coordinates (ξ, η, φ), where φ is the angle of rotation around the z_p-axis, the volume element transforms like:

$$\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \mathrm{d}\varphi\,\rho_\mathrm{p}\,\mathrm{d}\rho_\mathrm{p}\,\mathrm{d}z_\mathrm{p} = f^3(\xi^2-\eta^2)\,\mathrm{d}\varphi\,\mathrm{d}\eta\,\mathrm{d}\xi.$$

For large ξ the quantity $f\xi$ is approximately equal to the spatial distance of some point with the coordinates (ξ, η, φ) from the origin of the $(x_\mathrm{p}, y_\mathrm{p}, z_\mathrm{p})$-coordinate system.

For their numerical evaluation the infinite-volume integrals (5.10), (5.11), (5.15) and (5.16) are approximated by nested one-dimensional integrals over finite intervals,

$$\int_1^{\xi_\mathrm{max}} \mathrm{d}\xi \int_{-1}^{1} \mathrm{d}\eta \int_0^{2\pi} \mathrm{d}\varphi\, f^3(\xi^2-\eta^2)\,\ldots\,,$$

since the relevant integrands are expected to give negligible contributions to the infinite integral outside some sufficiently large ellipsoid characterised by $\xi < \xi_\mathrm{max}$. The ξ- and η-integrals are then computed by Gauß-Legendre quadrature formulas which corresponds to a polynomial interpolation of the integrand as a function of ξ and η respectively, see e.g. [AS65, PTVF92, DH93]. The φ-integration is carried out using the extended trapezoidal rule which is more simple than the Gauß-Legendre quadrature. Nonetheless, it is appropriate because the integrand is periodic in φ and the n-point extended trapezoidal rule is an exact quadrature formula for all trigonometric polynomials up to the order $n-1$. The n-point extended trapezoidal rule for the φ-integral corresponds to an integral over the interpolation of the integrand using a trigonometric polynomial of the order $n-1$. Contrary to the Gauß-Legendre quadrature formula the trapezoidal rule takes into account the periodicity of the integrand in φ [HH89, DR75, DH93].

In order to evaluate the integrand for some space-time point with coordinates (t, ξ, η, φ) the space-time coordinates $(t', r_\mathrm{A}, \vartheta_\mathrm{A}, \varphi_\mathrm{A})$ and $(t'', r_\mathrm{B}, \vartheta_\mathrm{B}, \varphi_\mathrm{B})$ of this event have to be computed. For the sake of completeness the transformations performed by the program will be described here. Starting from prolate spheroidal coordinates (ξ, η, φ) the spatial Cartesian coordinates $(x_\mathrm{p}, y_\mathrm{p}, z_\mathrm{p})$ are determined according to equation (A.9) and the following relations:

$$x_\mathrm{p} = \rho_\mathrm{p}\cos\varphi, \qquad y_\mathrm{p} = \rho_\mathrm{p}\sin\varphi.$$

The Cartesian coordinates (x, y, z) are then obtained by a rotation around the y_p-axis followed by a translation in z-direction:

$$x = x_\mathrm{p}\cos\alpha + z_\mathrm{p}\sin\alpha, \qquad y = y_\mathrm{p}, \qquad z = -x_\mathrm{p}\sin\alpha + z_\mathrm{p}\cos\alpha + \frac{t(v_\mathrm{B}+v_\mathrm{A})}{2}.$$

Here the coefficients of the rotation matrix are given by:

$$\sin\alpha = \frac{b}{2f}, \qquad \cos\alpha = \frac{t(v_\mathrm{B}-v_\mathrm{A})}{2f}.$$

Finally the space-time coordinates $(t', r_\mathrm{A}, \vartheta_\mathrm{A}, \varphi_\mathrm{A})$ and $(t'', r_\mathrm{B}, \vartheta_\mathrm{B}, \varphi_\mathrm{B})$ are computed from (t, x, y, z) by Lorentz boosts according to equations (A.3), (A.4) and (A.5).

It is important to note that the times t' and t'' in the rest frames of the respective centres, are functions of both t and z. For some particular time t they, therefore, have to be evaluated for each spatial coordinate (ξ, η, φ) separately. For the purpose of computational efficiency there are different functions implementing the coordinate transformation for computations in the collider frame, where $\chi_{\mathrm{A}} = -\chi_{\mathrm{B}}$, for computations in the rest frame of centre A, where $\chi_{\mathrm{A}} = 0$, and for arbitrary frames of reference.

Quadrature methods based on prolate spheroidal coordinates have also been used in nonrelativistic coupled channel calculations [FRI] and by Toshima and Eichler for their relativistic calculations of excitation and charge transfer in the target frame [TE88A].

A.3.2 Quadrature using contracted and translated spherical coordinates. For scalar products at large times t, between states which are located at the same projectile Γ (single centre integrals), a quadrature method is used that is based on the radial coordinates $(r_\Gamma, \vartheta_\Gamma, \varphi_\Gamma)$ defined in equation (A.5). A short calculation shows that the volume element $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$ in the unprimed frame is given in terms of the radial coordinates of the boosted frame at time t by:

$$\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \frac{r_\Gamma^2}{\gamma_\Gamma}\,\mathrm{d}r_\Gamma\,\mathrm{d}\cos\vartheta_\Gamma\,\mathrm{d}\varphi_\Gamma.$$

In order to verify this relation, it must be remembered that the Cartesian coordinate z at time t in the unprimed frame may be written as,

$$z = \frac{r_\Gamma \cos\vartheta_\Gamma}{\gamma_\Gamma} + v_\Gamma t, \tag{A.10}$$

due to the uniform motion of centre Γ. For a numerical evaluation, the matrix elements (5.11) and (5.16) are approximated by nested one-dimensional integrals of the form,

$$\int_0^{r_{\max}} \mathrm{d}r_\Gamma \int_{-1}^{1} \mathrm{d}\cos\vartheta_\Gamma \int_0^{2\pi} \mathrm{d}\varphi_\Gamma\, \frac{r_\Gamma^2}{\gamma_\Gamma}\, \ldots\,,$$

similarly to the previous subsection. Gauß-Legendre quadrature formulas are used for the computation of the r_Γ- and $\cos\vartheta_\Gamma$-integrals and the extended trapezoidal rule for the φ_Γ-integral for the same reasons as described above.

In order to evaluate the integrand of an interaction matrix element between two basis functions located both, e.g., at centre A (cf. equations (5.11) and (5.16)) it is necessary to calculate times t' and t'', the distance d_{B}, and the radius r_{B}. Here the radius r_{B} must correspond to a space-time point with spatial coordinates $(r_{\mathrm{A}}, \cos\vartheta_{\mathrm{A}}, \varphi_{\mathrm{A}})$ in rest frame A, but with time t in the unprimed coordinate system.

The times t' and t'', of the rest frames of centre A and B respectively, are obtained by first determining z, as in equation (A.10), and then using the relations $t' = \gamma_{\mathrm{A}}(t - v_{\mathrm{A}} z)$ and $t'' = \gamma_{\mathrm{B}}(t - v_{\mathrm{B}} z)$. The radius r_{B} must be computed according to,

$$r_{\mathrm{B}} = \Big[(r_{\mathrm{A}} \sin\vartheta_{\mathrm{A}})^2 + b^2 + z''^2 - 2b\, r_{\mathrm{A}} \sin\vartheta_{\mathrm{A}} \cos\varphi_{\mathrm{A}}\Big]^{\frac{1}{2}},$$

where $z'' = \gamma_B(z - v_B t)$, as already defined above, and $\sin\vartheta_A = \sqrt{1-(\cos\vartheta_A)^2}$. The distance d_B, as given in equation (5.3), can be expressed simply in terms of t''.

An analogous procedure has been implemented for the numerical determination of matrix elements between two basis functions that are both located at centre B.

A.3.3 Discussion and improvements. The evaluation of the elements of the interaction and overlap matrices, $N(t)$ and $V(t)$, is the main reason for the computational complexity of relativistic coupled-channel calculations as presented in this thesis. In [RSG93, BRBW94] *single*-centre coupled channel calculations of heavy-ion collisions using much larger bases have been reported. As the coupled channel basis in that approach is comprised of eigenstates (and wave packets) of one centre only, there are no two-centre matrix elements. However, in the present two-centre calculations the number of two-centre integrals that have to be evaluated is nearly twice as large as the number of single-centre integrals, showing that the major effort is to compute the former.

In [RSG93, BRBW94] Coulomb boundary conditions have not been used and calculations have been carried out in the rest frame of the basis functions (target frame). As described in [RSG93] the external field of a point-like projectile can be decomposed into a multipole series (see also [EM95]). Although, compared to the nonrelativistic case, this expansion is complicated slightly by the Lorentz contraction of the projectile potential, angular momentum algebra can be used to reduce the computational effort of determination of the matrix elements. Three-dimensional integrals for the interaction matrix elements $V_{ij}(t)$ can be reduced to infinite sums over two one-dimensional integrals, one of which does not need to be evaluated for every pair of indices (i, j) separately, because it mainly depends on the spin-angular quantum numbers of the corresponding basis functions [RSG93]. This integration method is not applicable if coupled channel calculations with phase-distorted basis functions are considered, as done in this work. If coupled channel calculations not satisfying Coulomb boundary conditions are an option, the program can be optimised by evaluating the single-centre integrals according to the method described in [RSG93].

We conclude this section with a few remarks about further ideas to optimise the numerical quadrature. Two-dimensional quadrature formulas for the integration on the unit sphere have been developed for numerical calculations of quantum chemistry [Del96]. These might turn out to be more efficient than the present integration method for the unit sphere using nested one-dimensional quadrature formulas. Another feature of the presently adopted quadrature formulas is the use of a fixed number of points, independent of, e.g., the distance between the centres. One should expect, however, that the convergence of the quadrature formulas with respect to the number of points at which the integrands must be evaluated does depend on the distance of the centres and the different types of wave functions. Therefore, adaptive quadrature methods based on extrapolation techniques (Romberg quadrature) might be more powerful [PTVF92, DH93]. On the other hand, the Romberg quadrature method requires sufficient smoothness of the integrand, which is not provided by the present evaluation of radial wave functions by linear interpolation of tabulated data. Finally, there are alternatives to the use of prolate spheroidal coordinates for the

quadrature of two-centre integrals. Different multi-centre integration schemes have been developed for calculations in quantum chemistry. E.g., the decomposition of multi-centre integrals into a sum of weighted single-centre integrals, as described in [Bec88, PJB94], is applicable in principal also in the present context. In conclusion, we have proposed improvements of the present quadrature schemes. It must be remembered, however, that all these proposals require careful and time-consuming implementation, testing and evaluation before they can replace the present methods. They should be object of future work.

A.4 Integration of the coupled channel equations

A.4.1 Matrix computations. The program integrates the coupled differential equations (4.8) for undistorted and phase-distorted bases simultaneously. For the integration of these differential equations the coefficient matrices,

$$-\mathrm{i}N(t)^{-1}V(t),$$

must be determined for both choices of the basis functions. This is achieved by means of an LU-decomposition [PTVF92, DH93] of the respective overlap matrix $N(t)$. For matrix computations the 'Template Numerical Toolkit' (TNT) [Poz00] has been used and slightly extended to allow for the LU-decomposition of complex-valued matrices.

Singular value decompositions of the fundamental solution matrices $F(t, t_\mathrm{i})$, as described in subsection 4.2.1 and presented in figures 5.3 and 5.4, have been performed by interfacing the TNT package to the Fortran library for linear algebra LAPACK [ABB+99]. The latter is optimised and available in binary format for many computing architectures. It provides the Fortran routines `ZGESVD` and `CGESVD`, which implement the singular value decomposition for complex-valued matrices.

A.4.2 Time integration. The overlap and interaction matrices $N(t)$ and $V(t)$ are numerically evaluated only for the times of a time grid. This time grid has upper and lower boundaries t_f and t_i, usually chosen in a symmetrical manner as $t_\mathrm{i} = -t_\mathrm{f}$. The spacing of the time grid is chosen to be equidistant, with a time step Δ_t, except for an inner time-interval. In this inner time-interval, ranging from $-t_\mathrm{inner}$ to t_inner, the equidistant spacing can be made smaller, using the time step $\Delta_{t,\mathrm{inner}}$. These five parameters defining the time grid are controlled by command line parameters of the program, which are summarised in table A.1.

For times t not part of the time grid, the matrix $-\mathrm{i}N(t)^{-1}V(t)$ of coefficients of the linear differential equation (4.8) is linearly interpolated. In the program this is achieved by means of the class `MatrixInterpolationTNT`. The integration of the coupled channel equations is, therefore, based on this linear interpolation function of the coefficient matrices.

For the integration of the differential equation over time intervals of the time grid, a sophisticated algorithm described in [PTVF92, ch. 16] is used. The powerful Burlisch–Stoer extrapolation technique is combined with a stepsize-control algorithm proposed by Deuflhard. In the present numerical code this integration method has

been implemented for differential equations with complex-valued coefficients as a class called `GBSDIntegrationTNT`.

The integration method works as follows: For a given stepsize H the so-called modified midpoint method is used to integrate the fundamental solution $F(t, t_\mathrm{i})$ from some time t to the time $t + H$. This method subdivides the stepsize H into n substeps each of size $h = H/n$. The modified midpoint rule is applied several times with increasing n, yielding several results for the same integration step from t to $t+H$. These different results are then extrapolated to the limit of vanishing substep-size, $h = 0$, using a polynomial interpolation (Richardson's deferred approach to the limit). If the (estimated) numerical error of this extrapolation is larger than a given error bound the stepsize H may be reduced. The stepsize adjustment strategy proposed by Deuflhard, described in [PTVF92, sec. 16.4], has been used with minor modifications for the present program.

A time grid has been used for the integration of the coupled channel equations also in [TE88A]. The effect of the Burlisch–Stoer–Deuflhard integration method in the present context is that the accuracy of the numerically evaluated fundamental solution $F(t, t_\mathrm{i})$ is practically determined by the choice of the time grid only. In order to control this accuracy the asymptotic unitarity of the fundamental solution can be verified. As described in section 4.2 asymptotic unitarity is provided, if the singular values of the fundamental solution are all equal to one at large times t. The smaller grid spacings in the inner time-interval $[-t_\mathrm{inner}, t_\mathrm{inner}]$ of the time grid allows for a rough adaption: At small times t the coefficients of the differential equations grow strongly as t tends to zero, due to maximum interaction at the closest approach of the centres (see e.g. figures 5.1 to 5.4).

A.4.3 Improvements. It may be attempted to apply the Burlisch–Stoer–Deuflhard integration method directly, without using a time grid and interpolating the coefficient matrices. However, it must be remembered that the numerical calculation of the overlap and interaction matrices by three-dimensional quadrature is very time-consuming and the use of directly evaluated coefficient matrices may actually increase the computing time. Furthermore extrapolation techniques, as the integration method described, require sufficient smoothness of the numerically evaluated coefficients as a function of time [PTVF92]. This may not be provided due to numerical inaccuracies of the quadrature formulas described above, but it is clearly true for the linear interpolation used for numerical calculations of this work. Finally note that using cubic splines for the interpolation of the coefficient matrices is presumably a valuable improvement of the present code, allowing for a larger spacing of the time grid and providing the necessary smoothness of the coefficients necessary for the Burlisch–Stoer–Deuflhard method.

A.5 Distributed computations

The numerical code has to perform two different tasks: On one hand, the numerical evaluation of the matrix elements, on the other hand the time integration of the coupled channel equations. In principle these two tasks can be separated from each other. They may be implemented in different programs, provided the matrix elements

can be passed from one program to the other. The latter could be achieved by storing the matrix elements, or the coefficient matrices of the differential equation, on mass storage devices. Another possibility is to write software that is able to exchange data using communication links.

The latter approach has been implemented. In this approach the process integrating the differential equations (master process) sends requests for matrix elements to a process only evaluating matrix elements (server process). A single master process is able to communicate with several different server processes. In the present program, a request of the master process sent to a server process contains the values of the following parameters:

$$\chi_{\mathrm{B}}^{\mathrm{c}}, \chi_{\mathrm{frame}}^{\mathrm{c}}, b, t, i \text{ and } j. \tag{A.11}$$

Here i and j are the row and column indices of the requested matrix elements. After the computation, of overlap and matrix elements for undistorted as well as phase-distorted bases, the server process returns four complex numbers to the master, namely the values of the matrix elements:

$$N_{ij}^{\mathrm{u}}(t),\ V_{ij}^{\mathrm{u}}(t),\ N_{ij}^{\mathrm{p}}(t) \text{ and } V_{ij}^{\mathrm{p}}(t).$$

Here the superscripts u and p refer to undistorted and phase-distorted bases respectively. As soon as the master process receives these results from a particular server process the former sends a new request to the server. The master process communicates with all server processes simultaneously, assembles the overlap and interaction matrices, determines the matrix of coefficients $-\mathrm{i}N(t)^{-1}V(t)$, integrates the differential equations and writes the results to several output files.

The program is constructed such that it can either take the role of a master or a server process. Typically, several processes are started simultaneously on different processors or networked computers. These processes are numbered and the first process automatically takes the role of the master process, establishing communication links to the other processes, which automatically start their operation in the server mode. If only a single process is started, it carries out both tasks, evaluating matrix elements and integrating the differential equations.

The principal advantage of the multiple-processor over single-processor computations is the much increased computing speed. E.g., a computation taking several days on a single-processor workstation could be executed on a network of workstations and personal computers during several hours. In many numerical calculations of this work multiple-processor computations have been necessary in order to get acceptable compute times. In addition, the parallelisation of the code allowed for the use of massively parallel processor systems, providing access to such powerful computing machinery.

A.5.1 Implementation details. Two different kinds of computing facilities have been used for the present work. On one hand, a heterogeneous network of workstations and personal computers connected via standard ethernet hardware, on the other hand, massively parallel processor systems of type Cray T3E. The communication between processes was realised by means of the massage passing interface [MPI96, GLS99]. Due to the fact that an implementation of the message passing interface is available both for TCP/IP-connected workstations [MPI00] and for the

proprietary hardware of the Cray T3E, it was technically feasible to write a *portable* code that could be used on both types of hardware.

We would like to close this appendix with a few remarks about technical aspects. The network of workstations and personal computers, connected via ethernet, included several different computing architectures, namely: Compaq True64 Unix running on workstations with Alpha processors, Sun Solaris for Sun Sparc computers, Hewlett-Packard Ultrix on HP-RISC computers, and Linux for personal computers with Intel processors. For all these systems (and additionally the Cray T3E architecture) the program sources had to be compiled and linked using different, partially incompatible compilers and linkers. This fact required adapting of sources to the various environments, selectable by preprocessor directives [Reg96] and compiler options. Having made available binaries of the program for the different computing architectures, the latter have been used simultaneously for a single, distributed coupled channel calculation. Typical distributed computations lasted several hours.

Finally, it should be emphasised that priority has always been given to numerical accuracy leading to long compute times. The optimisation of both accuracy and computing speed at the same time must be an object of future work. The potential for such improvements has been partly described in this appendix.