Relativistic ground state of diatomic molecules from
the numerical solution of the Dirac equation on
parallel computers

F. Fillion-Gourdeau¹, Emmanuel Lorin de la Grandmaison²,¹, André
D. Bandrauk³,¹
¹ Centre de Recherches Mathématiques, Université de Montréal, Montréal, Canada, H3T 1J4
² School of Mathematics and Statistics, Carleton University, Ottawa, Canada, K1S 5B6
³ Laboratoire de chimie théorique, Faculté des Sciences, Université de Sherbrooke, Sherbrooke,
Canada, J1K 2R1
E-mail: filliong@CRM.UMontreal.ca

Abstract.
The parallel implementation of a variational method for the numerical solution of the time-
dependent Dirac equation is presented. This is utilized in the calculation of diatomic molecule
relativistic wave functions and ground state energy, in the Born-Oppenheimer approximation.
Following [1], a functional equation predicting the same spectrum as the Dirac equation is
discretized over a set of basis functions. We find that the combination of B-spline basis functions
and prolate spheroidal coordinate systems facilitate the vectorization of the technique using
domain decomposition. This procedure results in a non-linear eigenvalue problem which is solved
by iteration using Brent’s method. We validate our methodology by comparing with existing
results and show that it yields very accurate ones. We also analyze the parallelization and
demonstrate that our implementation has a good scaling performance. Finally, the utilization
of this method for laser-matter interaction and other physical applications will be discussed.

1. Introduction
The advent of new experimental facilities with lasers reaching intensities above $10^{19}$ W·cm$^{-2}$
brings new challenges and opportunities for scientists. In this intensity regime, the motion and
energy of an electron subjected to such electromagnetic fields approaches the relativistic domain.
Thus, the Schroedinger equation loses its validity and should be replaced by its relativistic
counterpart: the Dirac wave equation. The numerical solution of this equation is known to be
a challenging task due to the presence of negative energy levels, which forbids the utilization
of the Rayleigh-Ritz variational method [2], and the appearance of spurious fermionic states,
which occurs in the discretization process [3]. To solve these issues, a new numerical technique
has recently been developed for the time-independent case by Esteban et al. [1, 4–6]. Their
method is based on a variational min-max principle and allows the computation of relativistic
eigenstates of any bound scalar potentials.

In this work, we present our implementation of this method on a parallel computer
architecture and use it to compute the ground state energy of diatomic molecules. Note here
that similar calculations were performed in [1, 7–9] but our implementation uses different basis

functions and/or coordinates and is parallelized. This work is separated as in the following. In Section 2, the numerical method is presented which includes the variational formulation of the Dirac equation and the choice of basis functions. Then, in Section 3, the details of the implementation of this method on a parallel computer architecture are shown. The results can be found in Section 4 where the scalability of the algorithm and the bound state of diatomic molecules are investigated.

2. The numerical method
The Dirac equation describes the relativistic dynamics of spin-\(\frac{1}{2}\) particles (fermions) like the electron. Because the fermions are ubiquitous in nature, this equation is very important in many areas of physics and chemistry. In this analysis, we consider specifically the static Dirac equation without vector potential given by

\[ \hat{H} \psi(x) = E \psi(x) \quad \text{with} \quad \hat{H} = -ic(\alpha \cdot \nabla) + mc^2 \beta + V(x) \|, \]

where \(\hat{H}\) is the Hamiltonian operator, \(m\) is the electron mass, \(E\) is the electron energy and \(\psi(x) \in L^2(\mathbb{R}^3) \otimes \mathbb{C}^4\) is a four-spinor. The Dirac matrices are given by

\[ \alpha = \begin{bmatrix} 0 & \sigma \\ \sigma & 0 \end{bmatrix} \quad \text{and} \quad \beta = \begin{bmatrix} \mathbb{I}_2 & 0 \\ 0 & -\mathbb{I}_2 \end{bmatrix} \]

written in terms of the usual Pauli matrices defined as

\[ \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{and} \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \]

This equation gives a consistent description of electrons bound in molecules in the Born-Oppenheimer approximation, i.e. when the atoms are fixed in space. This is a valid approximation when the mass of the atom is much larger than the mass of the atom, which will always be the case for the systems considered in this study.

The main goal of this work is to calculate approximate solutions of Eq. (1). To achieve this, it is convenient to write the four-spinor as \(\psi(x) \equiv \begin{bmatrix} \phi(x) \\ \chi(x) \end{bmatrix}\) where \(\phi(x), \chi(x) \in L^2(\mathbb{R}^3) \otimes \mathbb{C}^2\) are two bi-spinors [1, 4, 5]. The Dirac static equation can then be decomposed into two coupled equations as

\[ \hat{R}_\chi = (E - mc^2 - V)\phi \quad \text{and} \quad \hat{R}_\phi = (E + mc^2 - V)\chi \]

where the operator \(\hat{R} \equiv -ic(\sigma \cdot \nabla)\). These two equations can be decoupled by substituting the solution of the second one into the first one, so that we get

\[ \chi = (E + mc^2 - V)^{-1} \hat{R}_\phi \quad \text{and} \quad \hat{R} \left( \frac{\hat{R}_\phi}{E + mc^2 - V} \right) = (E - mc^2 - V)\phi. \]

Multiplying the latter by \(\phi^\dagger\) on the left and integrating on space (using integration by parts) yields a functional equation given by [1, 4, 5]

\[ A[E, \phi] \equiv \int d^3x \left[ \left( \frac{|\hat{R}_\phi|^2}{E + mc^2 - V} - (E - mc^2 - V)|\phi|^2 \right) \right] = 0 \]

\(^1\) All the calculations will be performed in atomic units (a.u.) where \(m = 1, \hbar = 1\) and \(c = 1/\alpha\) where \(\alpha \approx 1/137\) is the fine structure constant. In all the equations however, we are keeping the mass explicitly, allowing to switch easily from atomic to natural units.
for any $\phi(x) \in L^2(\mathbb{R}^3) \otimes \mathbb{C}^2$. This is obtained by using the divergence theorem and by assuming that the wave function vanishes faster than $\sim \frac{1}{r^2}$ at infinity. This is the case when $V(x)$ is a Coulomb-like potential because the corresponding wave function vanishes as $\phi \sim e^{-r}$ when $r \to \infty$ [10].

It can be shown that the last equation predicts the same spectrum in the mass gap $[-m c^2, m c^2]$ as that of the Dirac equation [1, 4, 5, 11, 12]. Moreover, this formulation allows discretization schemes that use basis functions without the “variational collapse”. The latter is related to the fact that the Dirac operator is not bounded from below because of the existence of the negative energy continuum. Therefore, the usual variational formulations which approximate the energy by minimizing a functional will fall into this lower part of the spectrum and will not be able to find the bound states, unless complicated projection operator techniques are used [2]. The method presented in this work avoid these complications and allows the calculation of bound states in the mass gap without the negative energy continuum contribution. Thus, as shown in the following, it is possible to develop a numerical method that solves Eq. (6) by discretizing over a set of basis functions.

### 2.1. Formulation in cylindrically symmetric system

For a diatomic molecule, the case considered in this study, the static potential is

$$V(x) = -\frac{Z_1}{|x + R\hat{z}|} - \frac{Z_2}{|x - R\hat{z}|}. \quad (7)$$

where $Z_{1,2}$ are the atomic electric charges, $2R$ is the inter-atomic distance and $\hat{z}$ is a unit vector in the $z$-coordinate direction. This potential is axially symmetric so it is possible to reduce by one the number of dimensions; the azimuthal coordinate dependence can be treated analytically by factorization. Thus, the four-spinor in cylindrical symmetry reads [7, 13]

$$\psi(x) = \begin{bmatrix} \phi(\xi, \eta, \theta) \\ \chi(\xi, \eta, \theta) \end{bmatrix} \begin{bmatrix} \phi_1(\xi, \eta)e^{i(j_z-1/2)\theta} \\ \phi_2(\xi, \eta)e^{i(j_z+1/2)\theta} \\ i\chi_1(\xi, \eta)e^{i(j_z-1/2)\theta} \\ i\chi_2(\xi, \eta)e^{i(j_z+1/2)\theta} \end{bmatrix}. \quad (8)$$

where $j_z$ is the angular momentum projection on the $z$-axis and $\eta, \xi$ are prolate spheroidal coordinates. As will be discussed later, this choice of coordinate system is very convenient for the numerical implementation. They are defined as

$$x = R \left[ (\xi^2 - 1)(1 - \eta^2) \right]^{1/2} \cos \theta, \quad (9)$$
$$y = R \left[ (\xi^2 - 1)(1 - \eta^2) \right]^{1/2} \sin \theta, \quad (10)$$
$$z = R\xi \eta \quad (11)$$

where $\xi \in [1, \infty], \eta \in [-1, 1]$ and $\theta = [0, 2\pi]$ (azimuthal angle).

The discretization of Eq. (6) with the potential Eq. (7) proceeds by expanding the wave function over a set of basis functions, in the same spirit as in Galerkin’s Methods. Thus, the bi-spinor can be written as

$$\phi_{1,2}(\xi, \eta) = \sum_{n=1}^{N} a_n^{(1,2)} B_n^{(1,2)}(\xi, \eta) \quad (12)$$

where $a_n^{(1,2)}$ are the coefficient of the basis expansion and $B_n^{(1,2)}(\xi, \eta)$ are the basis functions (to be defined later), for components 1 and 2 respectively. The explicit expression of Eq. (6)....
in discretized form depends on the potential considered, on the coordinate choice and is a complicated functional of basis functions (some examples can be found in [1, 5]). For cylindrically symmetric systems expressed in prolate spheroidal coordinates, the case considered in this study, the expression can be found in Appendix A.

Generally, the equation $A[E, \phi] = 0$, once discretized, have the form

$$\sum_{i,j=1}^{2N} a_{ij} A_{ij}(E) a_j = \mu(E)$$

(13)

where $\mu(E)$ is the eigenvalue of the $A$ matrix and where

$$a_i = \begin{cases} a_{i}^{(1)} & \text{for } i \leq N \\ a_{i-N}^{(2)} & \text{for } i > N \end{cases}$$

(14)

Therefore, solving the last non-linear eigenvalue problem with $\mu(E) = 0$ gives an approximation of the energy $E$ and eigenfunctions: the energy is a solution of $\mu(E) = 0$ where $\mu$ is an eigenvalue of $A$ while wave function coefficients in the basis expansion are the $A$ matrix eigenvector coefficients.

2.2. Basis functions

Throughout this work, the B-spline basis functions are used (a description of these functions can be found in [14]). This choice is favored over other techniques because it can be easily implemented and because the B-spline basis has compact support, leading to a sparse matrix structure. This allows the use of powerful numerical routines for the eigenvalue determination.

The B-splines are fully determined from their order $k_{\xi,\eta}$ and knot vector using the iterative formula [14, 15]

$$b_{ki}^{k}(x) = \frac{x - t_i}{t_{i+k-1} - t_i} b_{ki}^{k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} b_{ki}^{k-1}(x)$$

(15)

where $t_i$’s are knots coordinates chosen arbitrarily in the domain under consideration and $x = \eta, \xi$ is the coordinate. The number of knots at a given coordinates essentially determines the continuity conditions at that point. Therefore, the number of knots points should be maximal at singular points (at the Coulomb singularity position for instance) to allow for a stiff behavior. Throughout this work, the knot vectors are given by the sequences

$$1 = \xi_1 = \ldots = \xi_{k_{\xi}} < \xi_{k_{\xi}+1} < \ldots < \xi_{n_{\xi}+1} = \ldots = \xi_{n_{\xi}+k_{\xi}} = \xi_{\max}$$

$$-1 = \eta_1 = \ldots = \eta_{k_{\eta}} < \eta_{k_{\eta}+1} < \ldots < \eta_{n_{\eta}+1} = \ldots = \eta_{n_{\eta}+k_{\eta}} = 1$$

(16)

(17)

Here, $n_{\xi,\eta}$ are the number of spline functions in $\xi$ and $\eta$ coordinates respectively. The coordinates of the knot vectors can be chosen arbitrarily in the domain under consideration. However, to improve accuracy, an exponential sequence with smaller intervals close to the singularities is used in this study. The knot sequences and domain structure for diatomic molecules are depicted in Fig. 1.

The basis function can then be written as the tensor product of B-spline functions. To handle the singularity of the Coulomb potential, this is multiplied by a global function $G^{(1,2)}(\xi, \eta)$ that accounts for the singular behavior, which results in

$$B_n^{(1,2)}(\xi, \eta) = G^{(1,2)}(\xi, \eta) b_{ki}^{k_{\xi}}(\xi) b_{kj}^{k_{\eta}}(\eta)$$

(18)
where \( n = [i, j] \in \mathbb{Z}^2 \), \( i \in [1, n_\xi] \) and \( j \in [1, n_\eta] \). The overall factor is defined as [7, 16]

\[
G^{(1,2)}(\xi, \eta) = G^{(1,2)}_1(\xi, \eta)G_2(\xi, \eta)
\]

(19)

where

\[
G^{(1,2)}_1(\xi, \eta) = [(\xi^2 - 1)(1 - \eta^2)]^{m_1(1,2)} \text{ with } m_1(1) = j_z - \frac{1}{2} \text{ and } m_1(2) = j_z + \frac{1}{2}
\]

(20)

accounts for the angular momentum dependence (\( j_z \) is the angular momentum projection on the \( z \)-axis). The second factor deals with the singular behavior and is given by

\[
G_2(\xi, \eta) = r_1^{-1+\gamma_1}r_2^{-1+\gamma_2}
\]

(21)

with

\[
r_1 = (\xi + \eta)R, \ n_2 = (\xi - \eta)R \text{ and } \gamma_{1,2} = \sqrt{\left(j_z + \frac{1}{2}\right)^2 - \alpha^2Z_{1,2}^2}.
\]

(22)

This form of the overall factor is based on the analytical solution of the single atom: as one gets closer to one of the atom in the molecule, the potential tends to the single atom potential and the wave function of the molecule and its constituent ion coincide in that region.

The boundary conditions are chosen as \( \phi(\xi_{\text{max}}, \eta) = 0 \). Using B-splines, this can be implemented easily by setting \( b_n(\xi) = 0 \) and by considering only \( n_\xi - 1 \) B-spline functions in \( \xi \) coordinates. The other boundaries are free. It was shown in [1] that the choice of boundary conditions at \( \xi = \xi_{\text{max}} \) has negligible effect on the final result if \( \xi_{\text{max}} \) is large enough. Thus, we give \( \xi_{\text{max}} \) the values already determined in [1, 7].

### 3. Parallel Implementation

The code implementing the previously described numerical method for the solution of the time-independent Dirac equation is written in the C++ programming language. The parallelization is performed by using a domain decomposition strategy and the inter-processor communications are done with the Message Passing Interface (MPI). The details of the parallelization and algorithms are presented in the following.

The algorithm can be separated into four main steps:

(i) Guess an initial value for the bound state energy \( E \).
(ii) Build the \( A \) matrix.
(iii) Diagonalize \( A \) to evaluate \( \mu(E) \).
(iv) Solve the equation \( \mu(E) = 0 \).

The initial guess for the energy is given by \( E_{\text{init}} = E_1 + E_2 \) where \( E_{1,2} \) are the energies of the single atoms, which have well-known analytical expressions [10].

The next step is the construction of the \( A \) matrix and is the more demanding in terms of computation time. It consists in the numerical evaluation of Eq. (A.1) which is an integral equation. This part of the resolution is an embarrassingly parallel problem: each region can be integrated independently. Each element is sent to a different processor and the PPPack library (written in FORTRAN 90) [15] is used to evaluate all the non-zero B-splines in that region. The integration is computed by the rectangle rule; the elements are subdivided into smaller domains as shown in Fig. 1 and the functions are evaluated at the center of these subregions. The result of the integration on each processor is added on the master processor to get the sparse \( A \) matrix.

The data flow between processors during this step is shown in Fig. 2.
The latter is diagonalized using the PRIMME software [17] which is an efficient preconditioned eigensolver for symmetric (and hermitian) matrices. This allows to compute the eigenvalues and eigenvectors of $A$ and consequently, to evaluate $\mu(E)$. This step is performed sequentially because the matrix is not too large (approximately $400 \times 400$) for the needed accuracy and the diagonalization is very efficient.

The root-finding algorithm of the GNU Scientific Library (GSL) based on Brent’s method [18] is used to solve the equation $\mu(E) = 0$. This method is a combination of the bisection and interpolation methods. Its main advantages is that it is guaranteed to converge once the root is bracketed, it does not necessitate the evaluation of function gradients and it has a better order of convergence than the bisection method [18].

4. Results

In this section, the results of our implementation are presented. We consider first the parallel performance to test the efficiency of our implementation. Then, we apply our method to the computation of the ground state of diatomic molecules.

4.1. Scaling properties

The parallel performance of the implementation is evaluated by looking at the efficiency defined as $e \equiv \frac{T_1}{N_p T_{N_p}}$ where $T_{N_p}$ is the computation time on $N_p$ processors and $T_1$ is the computation time on one processor. As a benchmark problem, we compute the ground state energy of diuranium $^{232+}_{Z_1,2}U_2$ with a semi inter-atomic distance of $R = \frac{1}{92} \approx 0.010869565$ a.u.. We split the domain into $12 \times 12$ elements and consider B-splines of order 7. We then consider two calculation parameters: in the first one, each element is subdivided into $25 \times 25$ regions.
Figure 2. Data flow between processors during the calculation of the $A$ matrix. Here, $N$ is the number of processors while $n$ is the number of regions. In this example, we have $n > N$. During step 1 (on the left), regions 1 to $N$ of the physical domain are sent to different processors where their $A$ matrix contributions are computed (continuous arrows). Then, the results are sent to the master processor where they are added together (dashed arrows). The same procedure is done in step 2 for regions $N + 1$ to $2N$. This continues until all region contributions are calculated.

while in the second one, we have $50 \times 50$ regions. The bound state energy obtained in this way is $E_{\text{bound}} \approx -9965.1878$ a.u. for the first one and $E_{\text{bound}} \approx -9964.8066$ a.u. for the second one. All the computation were performed on the MAMMOUTH cluster (which has Intel Xeon 64bit 3.6 GHz CPUs and an Infiniband non-blocking (800 MB/sec) network).

The results for the parallel efficiency in these computations are shown in Fig. 3. For low number of processors ($\lesssim 32$ for the first one and $\lesssim 64$ for the second one), the efficiency is slightly above 1. The most probable explanation for this result is the cache effect: as the number of processors is increased, the percentage of the problem fitted in cache memory is also enhanced. Because the cache memory is faster than normal memory, the computation time is reduced. Moreover, the more demanding part of the computation is embarrassingly parallel, which results in a very good parallelization performance. Above these number of processors however, the efficiency diminishes due to inter-communication between different processors. This diminution is more important in the smaller problem considered, suggesting that larger problems can be parallelized more efficiently on a large number of processors. This happens because the code spends more time to perform the embarrassingly parallel part (the construction of the $A$ matrix) in larger problems.

4.2. Diatomic molecules

In this section, we are investigating the convergence of our numerical method. More specifically, we study and calculate the ground state of di-thorium (Th$_2^{79+}$ which have $Z_{1,2} = 90$) and dihydrogen (H$_2^+$ which have $Z_{1,2} = 1$)). Note that both have only one electron. The semi inter atomic distance is set to $R = 1.90 \approx 0.011111$ a.u. for dithorium and to $R \approx 1.000$ a.u. for dihydrogen. The results for the calculation of the ground state binding energy for different simulation parameters are shown in Table 1.
Figure 3. Efficiency as a function of the number of processor for the diuranium molecule computation.

The results presented in this table show the convergence of the method as the number of elements and B-spline order is increased. In Table 2, a comparison with results obtained in the literature is presented. This shows that the results obtained are very accurate, although there is a small difference (≈ 0.03%) between our results and the results presented in [7] and [8]. This difference can be explained by a different choice of boundary conditions and element formulation.

Table 1. Results of the numerical computation for the ground state of H$_2^+$ and Th$_{179}^+$ for different calculation parameters.

| Order | $n_\xi$ | $n_\eta$ | # elements/cell | # of cells | $E_{H_2^+}$ (a.u.) | $E_{Th_{179}^+}$ (a.u.) |
|-------|---------|---------|----------------|-----------|-----------------|------------------|
| 5     | 8       | 8       | 625            | 16        | -1.1071293      | -9493.2409       |
| 3     | 10      | 10      | 625            | 64        | -1.1100043      | -9496.9237       |
| 5     | 12      | 12      | 625            | 64        | -1.1024476      | -9503.4105       |
| 7     | 14      | 14      | 625            | 64        | -1.1025155      | -9503.6125       |
| 5     | 16      | 16      | 625            | 144       | **-1.1026076**  | **-9504.2743**   |

5. Conclusion
In this work, we presented a numerical method to solve the time-independent Dirac equation based on a variational principle and using a B-spline discretization. The Coulomb singularity was dealt with by adding an overall factor to the basis functions. We applied this method to the computation of diatomic molecule ground state energies. Because of their axial symmetry, it was convenient to use prolate spheroidal coordinates. The method was implemented in parallel and the code showed an efficient scaling with the number of processors. It was used to compute...
the ground state energy of dihydrogen, dithorium and diuranium. A comparison with results in the literature demonstrated that our method yields very accurate results.

This method can then be utilized in many applications. Among others, this can be used to investigate relativistic laser-matter interaction: the solution obtained from this code can be used as an initial condition for the solution of the time-dependent Dirac equation. This will be the subject of future investigations.

Appendix A. Explicit expression of Eq. (6)

The explicit expression of Eq. (6) is obtained by using the Dirac equation in cylindrical coordinates given in [13]. Then, combining the ansatz in Eq. (8) with the basis function expansion, we get

\[
A(E) = \sum_{i=1}^{N} \sum_{j=1}^{N} \int d^3x \left\{ \right. \\
\left. a_i^{(1)} a_j^{(1)} \left[ (\partial_t B_i^{(1)})(\partial_r B_j^{(1)}) + (\partial_r B_i^{(1)})(\partial_t B_j^{(1)}) + \frac{(j_z - \frac{1}{2})}{r^2} B_i^{(1)} B_j^{(1)} \right] - \frac{(j_z - \frac{1}{2})}{r} B_i^{(1)} \left( \partial_r B_j^{(1)} \right) - \frac{(j_z - \frac{1}{2})}{r} \left( \partial_r B_i^{(1)} \right) B_j^{(1)} \right] \frac{c^2}{E + mc^2 - V} \\
+ a_i^{(2)} a_j^{(2)} \left[ (\partial_t B_i^{(2)})(\partial_r B_j^{(2)}) + (\partial_r B_i^{(2)})(\partial_t B_j^{(2)}) + \frac{(j_z + \frac{1}{2})}{r^2} B_i^{(2)} B_j^{(2)} \right] + \frac{(j_z + \frac{1}{2})}{r} B_i^{(2)} \left( \partial_r B_j^{(2)} \right) + \frac{(j_z + \frac{1}{2})}{r} \left( \partial_r B_i^{(2)} \right) B_j^{(2)} \left[ \right] \frac{c^2}{E + mc^2 - V} \\
+ a_i^{(1)} a_j^{(2)} \left[ (\partial_t B_i^{(1)})(\partial_r B_j^{(2)}) + \frac{(j_z - \frac{1}{2})}{r^2} B_i^{(1)} B_j^{(2)} \right] - \left( \partial_r B_i^{(1)} \right)(\partial_r B_j^{(2)}) + \frac{\left( j_z + \frac{1}{2} \right)}{r} \left( \partial_r B_i^{(1)} \right) B_j^{(2)} \left] \right. \right. \\
+ a_i^{(2)} a_j^{(1)} \left[ (\partial_t B_i^{(2)})(\partial_r B_j^{(1)}) + \frac{(j_z - \frac{1}{2})}{r^2} B_i^{(2)} B_j^{(1)} \right] - \left( \partial_r B_i^{(2)} \right)(\partial_r B_j^{(1)}) + \frac{\left( j_z + \frac{1}{2} \right)}{r} \left( \partial_r B_i^{(2)} \right) B_j^{(1)} \left] \right. \right. \\
+ a_i^{(1)} a_j^{(1)} (V + mc^2 - E) + a_i^{(2)} a_j^{(2)} (V + mc^2 - E) \\
\equiv \sum_{i=1}^{2N} \sum_{j=1}^{2N} a_i A_{i,j}(E) a_j
\]

(A.1)
The last expression can then be expressed in prolate spheroidal coordinates by using

\[
\partial_r = \frac{\sqrt{(\xi^2 - 1)(1 - \eta^2)}}{R(\xi^2 - \eta^2)} [\xi \partial_\xi - \eta \partial_\eta] \quad \text{(A.3)}
\]

\[
\partial_z = \frac{(\xi^2 - 1)}{R(\xi^2 - \eta^2)} \eta \partial_\xi + \frac{(1 - \eta^2)}{R(\xi^2 - \eta^2)} \xi \partial_\eta \quad \text{(A.4)}
\]

and the integration measure is given by

\[
d^3 x = R^3(\xi^2 - \eta^2)d\xi d\eta d\theta. \quad \text{(A.5)}
\]

References
[1] Dolbeault J, Esteban M J and Séré E 2003 International Journal of Quantum Chemistry 93 149 – 155
[2] Datta S N 1980 Chemical Physics Letters 74 568 – 572
[3] Stacey R 1982 Phys. Rev. D 26 468–472
[4] Dolbeault J, Esteban M J and Séré E 2000 Journal of Functional Analysis 174 208 – 226
[5] Dolbeault J, Esteban M J, Séré E and Vanbreugel M 2000 Phys. Rev. Lett. 85 4020–4023
[6] Bandrauk A, Delfour M and Bris C (eds) 2007 High-dimensional partial differential equations in science and engineering (CRM proceedings & lecture notes vol 41) (American Mathematical Society)
[7] Kullie O, Kolb D and Rutkowski A 2004 Chemical Physics Letters 383 215 – 221
[8] Artemyev A N, Surzhykov A, Indelicato P, Plunien G and Stohlker T 2010 Journal of Physics B: Atomic, Molecular and Optical Physics 43 235207
[9] Sundholm D, Pyykko P and Laaksonen L 1987 Physica Scripta 36 400
[10] Itzykson C and Zuber J B 1980 Quantum Field Theory (Mcgraw-hill)
[11] Dolbeault J, Esteban M J and Séré E 2000 Calculus of Variations and Partial Differential Equations 10(4) 321–347
[12] Dolbeault J, Esteban M J, Loss M and Vega L 2004 Journal of Functional Analysis 216 1 – 21
[13] Schütz P, Wietschorke K H and Greiner W 1983 Journal of Physics A: Mathematical and General 16 1999
[14] Bachau H, Cormier E, Decleva P, Hansen J E and Martin F 2001 Reports on Progress in Physics 64 1815
[15] De Boor C 1978 A practical guide to splines (Springer-Verlag, New York :)
[16] Kullie O, Dusterhoff C and Kolb D 1999 Chemical Physics Letters 314 307 – 310
[17] Stathopoulos A and McCombs J R 2010 ACM Trans. Math. Softw. 37(2) 21:1–21:30
[18] Brent R 1973 Algorithms for Minimization without Derivatives (Prentice-Hall)