Variational Calculation of the Hydrogen Molecular Ion

by

Ye Ning

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in the Graduate Academic Unit of Physics Department

Supervisor: Zong-Chao Yan, Ph.D., Physics Department

Examining Board: Allan Adam, Ph.D., Chemistry Department, Chair
Igor Mastikhin, Ph.D., Physics Department
Dennis Tokaryk, Ph.D., Physics Department
Zong-Chao Yan, Ph.D., Physics Department

External Examiner: Anand K. Bhatia, Ph.D.,
NASA Goddard Space Flight Center

This dissertation is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

April, 2014

ABSTRACT

A high-precision study of the hydrogen molecular ion (HMI) H_2^+ is presented using the variational method in Hylleraas coordinates. Hylleraas coordinates with three non-linear parameters are adopted so that distance scales of the wave function may be described independently for all three radial coordinates r_1, r_2 and r_{12}. To achieve high precision, multi-precision arithmetic software QD (quad-double) is included in the code. To obtain high efficiency, the Message-Passing Interface (MPI) is used in computation. The ground state energy eigenvalue is calculated to a precision of 2 parts in 10^{34}, which represents a more than 3 orders of magnitude improvement over the best published values. High-precision variational calculations are also carried out for the first excited states of S and P symmetries, with the corresponding achieved precision of about 2 parts in 10^{31} and 3 parts in 10^{29}, respectively. Our results lay a sound foundation for studying high-order relativistic and quantum electrodynamic (QED) effects in this system and set a benchmark for developing other theoretical methods.

DEDICATION

To my parents and my family.

ACKNOWLEDGEMENTS

First I would like to thank my supervisor Dr. Zong-Chao Yan for his expertise on this subject, constant encouragement and financial support. I would also like to thank the Department of Physics and the School of Graduate Studies of the University of New Brunswick for providing me with teaching assistantship during my study. Special thanks to my coworkers Dr. Chun Li, Dr. Li-Ming Wang, Dr. Xiao-Feng Wang and Dr. Zhen-Xiang Zhong for useful discussion and their contribution to this work. Finally, I would like to express my thanks to my family and friends for their constant encouragement and help.

Table of Contents

List of Tables

List of Figures

Chapter One: Introduction

1.1 Background and Significance

High precision studies of simple atomic and molecular systems provide us with a powerful instrument to look into the most basic problems of physics [1] [2] and studies of fundamental physical constants is one of them [3]. The importance of fundamental physical constants is obvious for they are ubiquitous in scientific research. Sometimes, they can be used as landmarks for different domains of science. However, even if we are equipped with advanced precise techniques like optical, radio and NMR spectroscopy, etc., in general truly fundamental effects, which can be expressed in terms of basic physical laws and fundamental physical constants, can seldom be studied directly, unless we can find an appropriate system to study.

Over the years, people have found that few-body systems, including simple atoms and molecules, can be used as the best candidates for they have both experimental and theoretical advantages. Their simple structures enable them to be manipulated and measured in a relatively easier manner. For the same reason, they can be clearly described by quantum mechanics and quantum electrodynamics (QED) and calculated results can be interpreted in terms of fundamental physical constants. A concrete example is the measurement of the electron-proton mass ratio through rotational υ_{ro} and vibrational υ_{vib} frequencies of the hydrogen molecular ion H_2^+ (HMI) by using the following dependencies [4]:

$$\upsilon_{\mathrm{vib}} \sim \sqrt{\frac{m_e}{\mu}} R_\infty, \quad \upsilon_{\mathrm{ro}} \sim \frac{m_e}{\mu} R_\infty,$$

where m_e is the mass of electron, μ is the reduced mass of the two nuclei and, R_∞ is the Rydberg constant. This also provides us with the opportunity to determine other mass ratios such as m_e/m_d, m_e/m_t, m_p/m_d, m_p/m_t by measuring relevant rovibrational transition frequencies in D_2^+, HD^+, HT^+ combined with high-precision calculations, where, m_p, m_d and m_t are the masses of the proton, deuterium and tritium nuclei, respectively.

Besides the above, high precision calculation will help us to better understand simple atomic and molecular systems. The HMI itself is a good example. In the past four decades, calculations on the non-relativistic ground state of the HMI have improved remarkably from an early value of –0.597 139 062 5 from [5] to the present value of –0.597 139 063 123 405 074 834 134 096 026 189 9(1) obtained in this work (atomic units are used from here to the end of the thesis). This high precision energy determination implies a high precision wave function. Equipped with such a wave function, various higher order effects, such as relativistic and QED effects can be evaluated precisely. Meanwhile, the object of current study, the HMI, belongs to a large family named three-body Coulomb systems. This family contains not only traditional atomic systems like helium, which has been investigated from the beginning of quantum theory, but also some other exotic atomic system like H^- and Ps^-. Although different family members may have their unique features, such as different masses, charges, spins etc., from the perspective of computation, if all these factors have been taken into account, it one can readily to generalize calculation from one system to another. Furthermore, we note that the study of some exotic atomic systems is intriguing. For example, high precision energy levels for some few-body systems may help us discern the difference between matter and antimatter [6].

1.2 Survey of History

1.2.1 Brief History of High Precision Calculation

Schrödinger's equation can be regarded as one of the most important cornerstones of modern physics. In atomic physics, except for hydrogen, it is impossible for us to find an analytical solution for an atom with more than one electron. Even helium, with only one more electron than hydrogen, cannot be described analytically due to the existence of correlation between the two electrons. Therefore, it is necessary to develop a series of theoretical methods to pursue numerical solutions as precisely as possible. These methods include configuration interaction methods, perturbation methods, variational methods, self-consistent field and statistical methods (such as the Thomas-Fermi method) etc. In the practical computation, according to different situations and requirements, one of the method or another may be adopted to find an approximate solution to the Schrödinger's equation.

Among all those different methods, variational methods are unique due to their simplicity in principle and versatility in usage. From 1928 to 1929, Hylleraas introduced a new coordinate system which was later called "Hylleraas coordinates" [7]. He invented the "Hylleraas variational method" by combining his coordinates with the variational principle. This method has turned out to be a powerful one in dealing with atomic systems containing a few electrons, such as helium. For example, for the ground state of helium, in 1929, by using only 6 terms in his basis set, Hylleraas achieved the energy level of –2.903 24, which has 4 significant digits with the error of only 0.06%. This result was improved to –2.903 722 5 by Kinoshita [8] in 1957, by using 39 terms. With

the development of computer technology since the 1960s, the Hylleraas variational method has been applied successfully to several few body systems. The accuracy for the ground state energy of helium has now been calculated to 44 significant figures [9]. Such a high level of precision is sufficient for any practical applications.

Though it is called an approximate method, we can still obtain results of higher precision out of it, provided the method can be performed sufficiently effectively to overcome possible numerical problems that may occur. From this respect, to some extent, its solution can be regarded as an "exact" one. The underlying reason of its effectiveness is that Hylleraas coordinates include the distance of two electrons explicitly, and this inclusion means the consideration of correlation between the two electrons. If we were to expand $r_{12}^n = |\mathbf{r}_1 - \mathbf{r}_2|^n$ in terms of $\mathbf{r}_1$ and $\mathbf{r}_2$, we can see that a calculation using Hylleraas coordinates has essentially taken into account in an effective way of all possible angular momentum couplings between two single-electron states in the picture of configuration interaction.

1.2.2 History of Study on the HMI

The HMI was discovered by J. J. Thomson [10] in 1907. Since then, this object has been studied both theoretically and experimentally for more than one century, and people have made great progress on both sides. Fig. 1 presents the history of computational precision for the ground state energy level from 1975 to 2014. Using the result of the present work as a reference, we have determined the relative error for each calculation, which serves as the ordinate. From this figure we can see that the computational precision has been improved from a few parts in 10^5 to a few parts in 10^{34}.

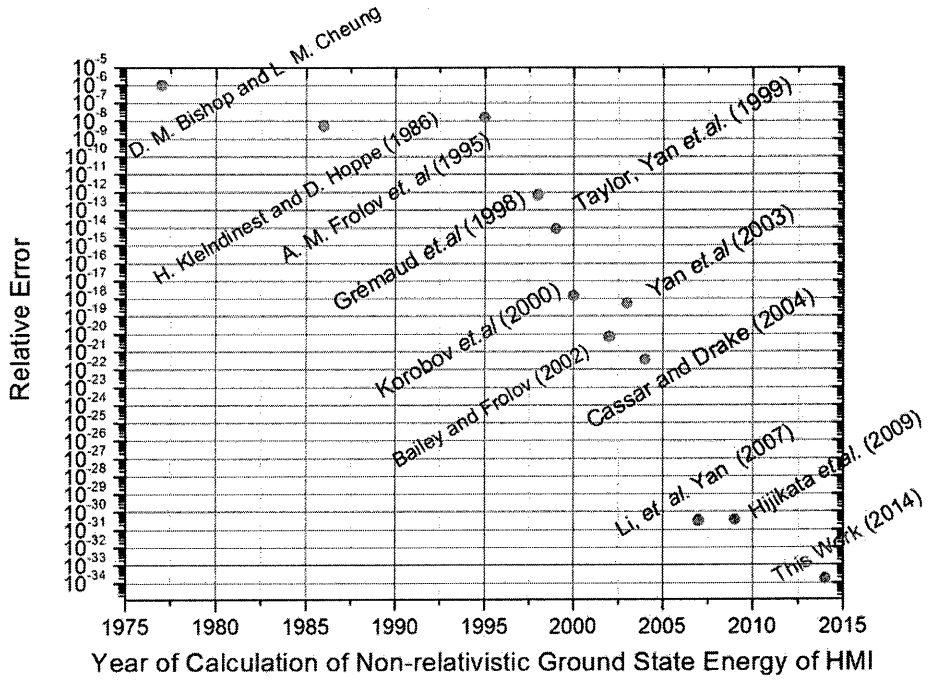

Figure 1 Advances of theoretical calculations of the ground state of the HMI over the years

For the theoretical side, in early years, the study of this simplest molecule demonstrated the power of quantum mechanics by explaining the existence of a stable ground state that was in agreement with experimental observation while classical theory failed to do so. At that time, the molecular orbital method, in the form of a linear combination of atomic orbital (LCAO-MO method), was proposed. This method has been summarized in some review articles such as the Carrington's work [11] and the references therein. In this review, they categorize the theoretical effort into two groups: those with the Born-Oppenheimer (BO) approximation and those without such an approximation. The BO approximation itself is a useful approach, especially when nuclei are heavy. In chemistry, the BO approximation, as well as the adiabatic approach, enables the

definition of potential energy surfaces on which the constituent nuclei undergo chemical reactions and molecular vibrations. Under the BO approximation, the HMI is the only molecular system that can be solved analytically [12] [13]. However, quantum effects due to nuclear motion are significant for light elements, which must be considered in dealing with many processes and properties, in order to reproduce experimental results to high accuracy. These include proton transfer and proton exchange processes in chemical and biological reactions, proton tunneling phenomena, and detailed analysis of the coupling among vibrational, rotational, and electronic motions. Therefore a desire for higher accurate wave functions beyond the BO approximation gave rise to the establishment of various non-adiabatic methods, including variational methods, the coupled-states method, the transformed Hamiltonian method, variation-perturbation method and free iterative complement interaction (ICI) method. For the purpose of comparison, we are going to briefly introduce some of them here, especially those developed in recent years.

In 2000, Korobov [14] performed a variational calculation for the ground state of the HMI, using basis functions with complex exponents

$$\Psi_0 = \sum_{i=1}^{\infty} \{U_i \Re[\exp(-\alpha_i r_1 - \beta_i r_2 - \gamma_i r_{12})] + W_i \Im[\exp(-\alpha_i r_1 - \beta_i r_2 - \gamma_i r_{12})]\} \mathcal{Y}_{l_1 l_2}^{LM}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2).$$

In the above expression, α_i, β_i and γ_i are chosen in a quasi-random manner according to

$$\alpha_i = \left\lfloor \frac{1}{2} i(i+1)\sqrt{p_\alpha} \right\rfloor [(A_2 - A_1) + A_1] + \iota \left\{ \left\lfloor \frac{1}{2} i(i+1)\sqrt{q_\alpha} \right\rfloor [(A_2' - A_1') + A_1'] \right\},$$

where $[x]$ designates the fractional part of x, ι is the imaginary unit, p_α and q_α are some prime numbers and the real variational parameters A_1, A_2 and A_1', A_2' are the end points of real intervals and need to be optimized. His variational calculation for the ground state energy, including 2200 terms in the wave function and using multiple precision package MPFUN [15], resulted in the value of –0.597 139 063 123 405 074 0.

By using perimetric coordinates, also in 2000, another variational calculation was performed by Hilico *et al* [16] using the wave functions

$$\Psi^{JM} = \sum_{T=-J}^{J} D_{M,T}^{J*}(\psi,\theta,\phi)\Phi_T^{JM}(R,\rho,\zeta),$$

where the angular functions are related to the matrix elements of the rotation operators $R_{M,T}^J$ [17]:

$$D_{M,T}^{J*}(\psi,\theta,\phi) = \sqrt{\frac{2J+1}{8\pi^2}} R_{M,T}^{J*}(\psi,\theta,\phi).$$

Their variational basis set is expressed in perimetric coordinates which takes full advantage of dynamical symmetries. Here Φ is taken as products of Laguerre polynomials in the perimetric coordinates. Energies were calculated for all $J = 0$ and 1 rovibrational states.

In the following year, Korobov [18], once again, used a variational method with a wave function taken in the form

$$\Psi_0 = \sum_{i=1}^{\infty} [C_i \cos(\upsilon_i R_{12}) + D_i \sin(\upsilon_i R_{12})] e^{-\alpha_i r_1 - \beta_i r_2 - \gamma_i R_{12}} + (\mathbf{r}_1 \leftrightarrow \mathbf{r}_2),$$

with the parameters v_i, α_i, β_i and γ_i being generated in a pseudo-random manner as mentioned above. $\mathbf{r}_1 \leftrightarrow \mathbf{r}_2$ means exchanging the position of the two protons. Truncating this expansion after 800 terms yielded an energy of –0.597 139 063 123 40 (1).

In 2002, Frolov [19] performed an exponential variational calculation in perimetric coordinates. The basis functions he used were written as

$$\psi_i = e^{-\alpha_i u_1 - \beta_i u_2 - \gamma_i u_3} e^{\iota\delta_1 u_1 + \iota\epsilon_i u_2 + \iota f_i u_3} Y_{LM}^{l_1 l_2}(r_{31}, r_{32});$$

where $Y_{LM}^{l_1 l_2}(r_{31}, r_{32})$ are bipolar harmonics, and $u_i = \frac{1}{2}(r_{ik} + r_{ij} - r_{jk})$ are perimetric coordinates. Using 72-116 decimal digit accuracy (using the MPFUN [15] package mentioned above) and 2600 terms, he calculated the ground state energy of the HMI and obtained –0.597 139 063 123 405 074 74. In the same year, Bailey and Frolov [20], using the same approach, extended the number of terms to 3500 and obtained the lower energy value –0.597 139 063 123 405 074 83.

In 2003, Yan, Zhang, and Li [21] carried out a variational calculation for the HMI. They used a basis set in Hylleraas coordinates in the form.

$$\phi_{ijk}(\mathbf{r}_1, \mathbf{r}_2) = r_1^i r_2^j r_{12}^k e^{-\alpha r_1 - \beta r_2} \mathcal{Y}_{l_1 l_2}^{LM}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2).$$

Instead of choosing the electron as the origin, they picked one of the protons as the origin. The reason for this is that they could then present the internuclear vibrational motion in an effectively way through the $r_2^j e^{-\beta r_2}$ portion of the wave function, with $j \geq j_{min} = 35$. The internuclear vibrational motion can be well described by using Gaussian-like function under the BO approximation. It was first discussed by Bhatia [22] and Bhatia and Drachman [23] on the possibility of using Hylleraas type basis to simulate

such a behavior without using the BO approximation. The basis is generated by including all the terms such that

$$i + j + k \leq \Omega$$

where Ω is an integer not smaller than $j_{\min}$. Furthermore, they split the generated basis into 8 blocks, each having its own nonlinear parameters α and β. The n-th block contains all the terms satisfying

$$g_1 \leq j_1 \leq g_2,$$

$$g_n < j_n \leq g_{n+1}, \quad n = 2, \ldots, 8$$

with

$$g_n = \text{int}\left[j_{\min} + \frac{\Omega - j_{\min}}{8}(n-1)\right],$$

where $\text{int}[x]$ means the integer part of x. A complete optimization over those eight sets of nonlinear parameters was performed. For the ground state energy, using 1330 terms in their basis set, they obtained a value of –0.597 139 063 123 405 074 5(4). They also calculated the energy eigenvalues for the lowest P states of hydrogen molecular ions.

In 2009, Hijikata suggested a method named the free iterative complement interaction (ICI) [24]. This approach is based on the idea that the exact function would be expressed by a function of the Hamiltonian $f(H)$ applied to some function ψ_0 as

$$\psi_{\text{exact}} = f(H)\psi_0.$$

In the Schrödinger's equation, the exact wave function, together with the corresponding energy eigenvalue, can be regarded as an output while the Hamiltonian is an input. Therefore, they thought the simplest ICI wave function can be chosen in an iterative way as

$$\psi_{n+1} = [1 + C_n g(H - E_n)]\psi_n$$

where g function was introduced to prevent the singularity difficulty intrinsic to the Coulombic systems. As iteration proceeds, this ICI is supposed to converge to the exact wave function. To accelerate convergence, they also collected all the linearly independent functions $\{\phi_i\}$, $i = 1,2,\ldots,M_n$ from the right-hand side of the above formula and designated each an independent coefficient to form

$$\psi_{n+1} = \sum_{j=1}^{M_n} c_i \phi_i,$$

where all the coefficients $\left\{c_i^{(n)}\right\}$ are variationally determined. Here, n (the iteration cycle) is the "order", and M_n (the number of independent functions) is the "dimension" of order n. For the ground state of the HMI, they stopped the calculation at $n = 21$ with $M_n = 19286$, and claimed a benchmark result –0.597 139 063 123 405 074 834 134 096 025 974 142. In 2012, the Schrödinger's equation for the HMI system was also solved in perimetric coordinates using the Lagrange-mesh method by Olivares Pilón and Daniel Baye [25]. Although, for the moment, this method doesn't provide the most accurate result (13 digits), it can give us a larger spectrum of energy eigenvalues than any other methods can. Meanwhile the wave functions generated by this method have great simplicity, which is convenient to be used to further study oscillator strength, transition probability, and dipole as well as quadrupole operators.

Table 1: Detailed Collection and Comparison of the Non-relativistic Energies for the Ground State of the HMI. The Proton to Electron Mass Ratio m_p/m_e=1836.152 701 is Used.

Author	Year	Ref.	Energy
Bishop and Cheung	1977	[5]	-0.597 138 460
Kleindienst and Hoppe	1986	[26]	–0.597 139 06(3)
Frolov et al.	1995	[27]	–0.597 139 053 69
Rebane and Filinsky	1997	[28]	–0.597 139 063 123 40
Saavedra *et al.*	1998	[29]	–0.597 139 063 123
Grémaud *et al.*	1998	[30]	–0.597 139 063 123(1)
Taylor *et al.*	1999	[31]	–0.597 138 063 123 9 (5)
Moss	1999	[32]	–0.597 138 063 123 4
Korobov	2000	[14]	–0.597 138 063 123 405 704
Hilico *et al.*	2000	[16]	–0.597 138 063 123 40(1)
Bailey and Frolov	2002	[20]	–0.597 138 063 123 405 074 83
Yan *et al.*	2003	[21]	–0.597 138 063 123 405 074 5(4)
Cassar and Drake	2004	[33]	–0.597 138 063 123 405 074 834 337 7(21)
Li *et al.*	2007	[34]	–0.597 139 063 123 405 074 834 134 096 026 (5)
Hijikata *et al.*	2009	[24]	–0.597 139 063 123 405 074 834 134 096 025 974 142
Pilón *et al.*	2012	[25]	–0.597 139 063 123 3
This work			–0.597 139 063 123 405 074 834 134 096 026 189 9 (1)

As for the experimental side, the invention of magneto-optical trap (MOT) [35] has given rise to many seminal measurements of atomic and molecular systems. By using this technique, combined with other techniques such as magnetically tunable Feshbach resonances or with photoassociation, people can produce translationally cooled molecules at a temperature as low as 10 mK. Such a cold object enables us to observe rotational and

vibrational as well as other properties of a molecule to a high degree of precision. However, since the chief concern of this thesis is the theoretical work, only the most accurate and up-to-date experimental apparatus [36] [37] will be briefly introduced below.

Roughly speaking, the whole experimental setup can be divided into three relatively independent parts. The first one is the source of ions. To produce ionized molecules, the most straightforward way is perhaps to leak the neutral molecular gas into a vacuum chamber and ionize them through an electron beam. There are also many other ways to achieve this goal, such as electro-spray ionization (ESI) [38] or chemical reaction. Different methods may be applied to different molecules according to molecular properties. After that, a charge-to-mass ratio quadrupole mass filter is employed to select the ions we need.

The second part, the ion trap, is the most vital one, because we have a special requirement for the trap—the cold temperature. As we know, for atomic or molecular systems, low temperature means less translational kinetic energy which will cause less collision broadening and Doppler Effect on the spectrum. In particular, such an extreme physical condition can ensure longer lifetime for molecular vibrational and rotational levels, which is advantageous to spectroscopic observation. Up to now, the lowest temperature around a few mK for the HMI has been reached by Schiller's group [39] using a method called sympathetic cooling. Their work enables an improvement of spectroscopic measurements by 4 orders of magnitude or more. In this method, a certain kind of atomic ions, such as Be^+ in their experiment, are cooled in advance by using laser cooling technique so that they can serve as a coolant. When the ensemble of atomic ions

is well "crystallized", molecular ions can be loaded in. The term "crystallized" refers to a cold ion ensemble with developed shell structure, with temperatures below 200 mK. After that, the loaded ions can be cooled via its Coulomb interaction with coolant ions.

The rest part of the apparatus is composed of the laser system, which is used to excite molecular ions, and an optical detection system, which is used to collect spectral information. This part is common in all the spectroscopic experiments.

Chapter 2 Theory

2.1. Transformation of Schrödinger Equation

As is well known, Schrödinger equation, which is suitable to describe a non-relativistic system, can be written in the form

$$H\Psi = E\Psi, \qquad (1)$$

where H is the non-relativistic Hamiltonian operator, E is an eigenvalue and Ψ is the corresponding eigenfunction.

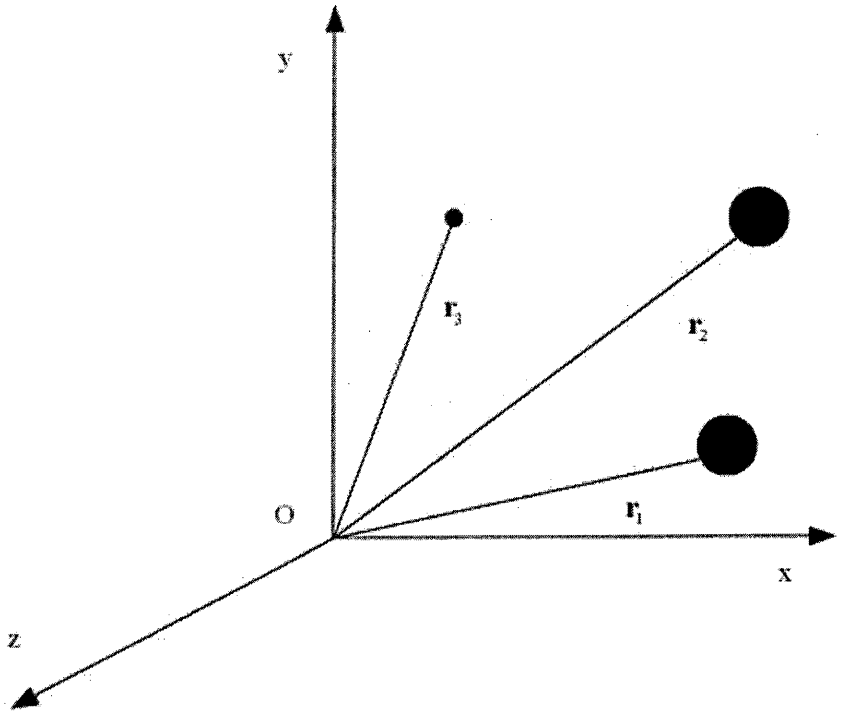

Figure 2 Fixed-origin coordinate system. Big dots are for the protons and small dot is for the electron.

If we choose an arbitrary point in space as the fixed origin of the coordinate system as shown in Fig.2, the Hamiltonian of a three-body system can be expressed as

$$H = -\frac{1}{2m_1}\nabla_1^2 - \frac{1}{2m_2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 + \frac{q_1q_2}{r_{12}} + \frac{q_1q_3}{r_{13}} + \frac{q_2q_3}{r_{23}}, \tag{2}$$

where

$$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|,\ (i,j = 1,2,3;\ i \neq j)$$

is the distance between particle i and j. In Cartesian coordinates $\mathbf{r_i} = (x_i, y_i, z_i)$ and

$$\nabla_i^2 = \frac{\partial^2}{\partial x_i^2} + \frac{\partial^2}{\partial y_i^2} + \frac{\partial^2}{\partial z_i^2}. \tag{3}$$

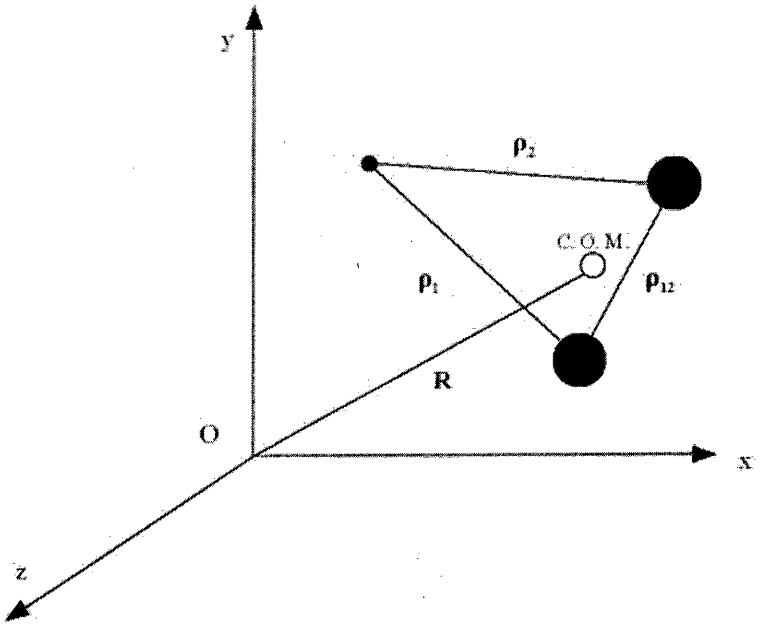

Figure 3 Internal and centre-of-mass coordinate system. Big dots are for the protons and small dot is for the electron.

On making the standard transformation to the center-of-mass as shown in Fig.3, and relative coordinates defined by

$$\begin{aligned} &\boldsymbol{\rho}_1 = \mathbf{r}_1 - \mathbf{r}_3 \quad \boldsymbol{\rho}_2 = \mathbf{r}_2 - \mathbf{r}_3 \\ &\mathbf{R} = \frac{m_1\mathbf{r}_1 + m_2\mathbf{r}_2 + m_3\mathbf{r}_3}{M}, \\ &M = m_1 + m_2 + m_3 \\ &\frac{1}{\mu_i} = \frac{1}{m_i} + \frac{1}{m_3}, \qquad i = 1, 2 \end{aligned} \tag{4}$$

the Hamiltonian can be rewritten as

$$\begin{aligned} H = &-\frac{1}{2\mu_1}\nabla_{\rho_1}^2 - \frac{1}{2\mu_2}\nabla_{\rho_2}^2 - \frac{1}{2M}\nabla_R^2 - \frac{1}{m_3}\nabla_{\rho_1}\cdot\nabla_{\rho_2} \\ &+ \frac{q_1q_3}{\rho_1} + \frac{q_2q_3}{\rho_2} + \frac{q_1q_2}{\rho_{12}}. \end{aligned} \tag{5}$$

Noticing that the potential energy terms in the Hamiltonian contain no **R**, the coordinates of center-of-mass, we can conclude that **R** can be treated as an ignorable coordinate [40], and the whole Hamiltonian can be further simplified into

$$\begin{aligned} H = &-\frac{1}{2\mu_1}\nabla_{\rho_1}^2 - \frac{1}{2\mu_2}\nabla_{\rho_2}^2 - \frac{1}{m_3}\nabla_{\rho_1}\cdot\nabla_{\rho_2} + \frac{q_1q_3}{\rho_1} \\ &+ \frac{q_2q_3}{\rho_2} + \frac{q_1q_2}{\rho_{12}} \end{aligned} \tag{6}$$

For the convenience of calculation, the next step is to remove the multiplicative constants by making a scale change which will put H in Z-scaled, reduced mass atomic units, that is,

$$r_i = \frac{Z_i}{a_1}\rho_i \Rightarrow \frac{\partial}{\partial\rho_i} = \frac{Z_i}{a_1}\frac{\partial}{\partial r_i}, \tag{7}$$

where

$$a_1 = \frac{\hbar^2}{\mu_1 e^2} = \frac{m_e}{\mu_1} a_0 \quad \text{and} \quad Z_i = \frac{q_i}{e}, \tag{8}$$

and a_0 is the Bohr radius. More specifically, for the HMI, there are two identical protons which can be labeled as 1 and 2 while the only electron can be labeled by 3. So we have the relationships $Z_1 = Z_2 = Z = 1$ and $Z_3 = -Z = -1$, which allows us to simplify the Hamiltonian further into

$$\frac{H}{Z^2 e^2/a_1} = -\frac{1}{2}\nabla_{r_1}^2 - \frac{1}{2}\nabla_{r_2}^2 - \frac{m_p}{1+m_p}\nabla_{r_1}\cdot\nabla_{r_2} - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{r_{12}}, \tag{9}$$

where m_p is the mass of proton. Therefore, the dimensionless Hamiltonian for the HMI is

$$\mathbb{H} = -\frac{1}{2}\nabla_{r_1}^2 - \frac{1}{2}\nabla_{r_2}^2 - \frac{m_p}{1+m_p}\nabla_{r_1}\cdot\nabla_{r_2} - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{r_{12}}, \tag{10}$$

where

$$\mathbb{H} = \frac{H}{Z^2 e^2/a_1} = H\left(1 + \frac{1}{m_p}\right),$$

and $\mu_1 = \mu_2$ has been used.

Thus, Eq. (1) can be transformed into a reduced form

$$\mathbb{H}\Psi = \mathbb{E}\Psi, \tag{11}$$

where $\mathbb{E}$ is related to the original energy eigenvalue of E by

$$\mathbb{E} = \frac{E}{Z^2 e^2/a_1} = E\left(1 + \frac{1}{m_p}\right).$$

2.2. Variational Calculation

In most cases, Schrödinger's equation cannot be solved *analytically*. As mentioned before, one often turns to the Rayleigh-Ritz variational method, which is a powerful method for finding out eigenvalues and eigenfunctions of a quantum system.

The basic idea of the Rayleigh-Ritz variational method is based on the following variational principle: an eigenvector of a Hamiltonian is a vector in Hilbert space that minimizes the expectation value

$$\langle H \rangle = E_\Psi = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}, \tag{12}$$

and E_Ψ thus becomes the corresponding eigenvalue. Using variational language, the variational principle can be expressed as

$$\delta E_\Psi = 0. \tag{13}$$

To achieve this purpose, we assume a trial wave function in the form

$$|\Psi_t\rangle = \sum_{i=1}^{N} c_i |\Phi_i\rangle, \tag{14}$$

where Φ_is are the basis functions and are not necessarily orthogonal to each other, and c_is are the linear coefficients, which can serve as variational parameters. Substituting $|\Psi_t\rangle$ into Eq. (12) yields

$$E_{\Psi_t} = \frac{\sum_{ij} c_i^* c_j H_{ij}}{\sum_{ij} c_i^* c_j O_{ij}}, \tag{15}$$

where $H_{ij} = \langle \Phi_i | H | \Phi_j \rangle$ and $O_{ij} = \langle \Phi_i | \Phi_j \rangle$. Performing partial differentiation of Eq. (15) with respect to those variational parameters, we have

$$\frac{\partial E_{\Psi_t}}{\partial c_k} = \frac{\sum_i c_i^* \left(H_{ik} - E_{\Psi_t} O_{ik}\right)}{\sum_{ij} c_i^* c_j O_{ij}}. \tag{16}$$

and

$$\frac{\partial E_{\Psi_t}}{\partial c_k^*} = \frac{\sum_j c_j \left(H_{kj} - E_{\Psi_t} O_{kj}\right)}{\sum_{ij} c_i^* c_j O_{ij}}. \tag{17}$$

From Eqs. (13), (16) and (17), we can see

$$\partial E_{\Psi_t} = \sum_k \frac{\partial E_{\Psi_t}}{\partial c_k} \delta c_k + \sum_k \frac{\partial E_{\Psi_t}}{\partial c_k^*} \delta c_k^* = 0. \tag{18}$$

Since δc_k and δc_k^* are arbitrary, the preceding equation implies

$$\frac{\partial E_{\Psi_t}}{\partial c_k} = \frac{\partial E_{\Psi_t}}{\partial c_k^*} = 0, (k = 1,2,3, \cdots, N), \tag{19}$$

which leads to

$$\sum_j c_j \left(H_{kj} - E_{\Psi_t} O_{kj}\right) = 0, (k = 1,2,3, \cdots, N). \tag{20}$$

Considering the Hamiltonian operator is Hermitian, we neglect the conjugate form of the above equation. And we simplify it further into the following matrix form

$$\mathcal{H}\mathcal{C} = E\mathcal{O}\mathcal{C}, \tag{21}$$

where $\mathcal{H}$ is the Hamiltonian matrix, $\mathcal{O}$ the overlap matrix, and $\mathcal{C}$ a column matrix formed by the expansion coefficients c_i. Explicitly, the above equation can also be written as

$$\begin{pmatrix} H_{11} & H_{12} & \dots & H_{1N} \\ H_{21} & H_{22} & \dots & H_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ H_{N1} & H_{N2} & \dots & H_{NN} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = E_{\Psi_t} \begin{pmatrix} O_{11} & O_{12} & \dots & O_{1N} \\ O_{21} & O_{22} & \dots & O_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ O_{N1} & O_{N2} & \dots & O_{NN} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} \tag{22}$$

Diagonalization of $\mathcal{H}$ will yield N eigenvalues $E_{\Psi_t}^j$ $(j = 1,2,3, \dots, N)$. The variational method is still powerful even if the trial wave function is NOT an exact eigenfunction of the Hamiltonian, for it can give us an upper bound to the corresponding exact energy eigenvalue. To clarify this point briefly, we need to expand the trial wave functions in terms of the exact normalized eigenfunctions of the Hamiltonian as the basis functions

$$|\Psi_t\rangle = \sum_{i=1}^{N} c_i |\varphi_i\rangle, \tag{23}$$

with $\langle\varphi_i|\varphi_i\rangle = 1$. Thus the energy E_{Ψ_t} can be expressed into

$$E_{\Psi_t} = \sum_i c_i^2 E_i. \tag{24}$$

Due to the facts that $E_i \geq E_0$ and $\sum_i c_i^2 = 1$, E_{Ψ_t} is always greater than the ground state energy $E_0 (E_{\Psi_t} \geq E_0)$. Consequently, E_{Ψ_t} is always an upper bound to the true state energy [41].

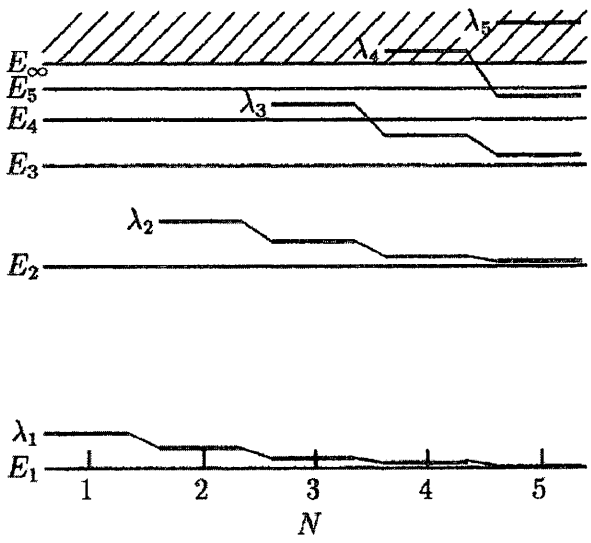

Figure 4 Illustration of the Hylleraas-Undheim-MacDonald Theorem

According to the matrix interleaving theorem, the higher variational eigenvalues are also upper bounds to the corresponding true eigenvalues. When an extra column and row are added to the formal $N \times N$ matrix, the N old eigenvalues lie between the $N + 1$ new ones. The new highest eigenvalue becomes higher than the previous highest one and the lowest becomes lower than the previous lowest one. Since the system is bounded from below, the variational eigenvalues become the exact eigenvalues in the limit $N \to \infty$. As a result, the N variational eigenvalues move progressively toward the exact values as N increases as shown in Fig. 4. This also implies that we can not only get an energy level for the ground state but also for other excited states of the system.

2.3. Solving the Matrix Equation

2.3.1. Power Method

From the preceding discussion, one can see we are now facing the problem of solving the matrix equation (21). There are many different ways of solving the so-called generalized eigenvalue problem [42] [43] [44]. However, if we are only interested in one specific energy eigenvalue and its corresponding eigenfunction, the power method [45] may be the best choice in terms of speed and robustness. This method is described below.

The basic idea of the power method is straightforward. Suppose $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the eigenvalues of $\mathcal{H}$ with the corresponding eigenfunctions $\Psi_1, \Psi_2, \cdots, \Psi_n$ which are linearly independent with each other. If $|\lambda_n|$ is the largest among these $|\lambda_i|$s, then $|\lambda_n|$ is called the dominant eigenvalue of $\mathcal{H}$ and its corresponding eigenvector Ψ_n the dominant eigenvector. Choosing an arbitrary vector χ_0 such as

$$\chi_0 = \sum_i^n a_i \Psi_i \,, \tag{25}$$

we may find a sequence

$$\begin{aligned} \mathcal{H}\chi_0 &= a_1\lambda_1\Psi_1 + a_2\lambda_2\Psi_2 + \cdots + a_n\lambda_n\Psi_n = \chi_1 \\ \mathcal{H}\chi_1 &= a_1\lambda_1^2\Psi_1 + a_2\lambda_2^2\Psi_2 + \cdots + a_n\lambda_n^2\Psi_n = \chi_2 \\ &\vdots \\ \mathcal{H}\chi_{k-1} &= a_1\lambda_1^k\Psi_1 + a_2\lambda_2^k\Psi_2 + \cdots + a_n\lambda_n^k\Psi_n = \chi_k \end{aligned} \tag{26}$$

As k becomes large, χ_k approaches the eigenvector of $\mathcal{H}$ with the largest eigenvalue:

$$\chi_k = a_1\lambda_1^k\Psi_1 + a_2\lambda_2^k\Psi_2 + \cdots + a_n\lambda_n^k\Psi_n,$$
$$\frac{\chi_k}{\lambda_n^k} = a_1\left(\frac{\lambda_1}{\lambda_n}\right)^k\Psi_1 + a_2\left(\frac{\lambda_2}{\lambda_n}\right)^k\Psi_2 + \cdots + a_n\Psi_n. \tag{27}$$

Considering $|\lambda_n|$ is dominant, and as $k \to \infty$, we are sure to get

$$\chi_k = a_n\lambda_n^k\Psi_n, (a_n \neq 0), \tag{28}$$

which fulfills $\mathcal{H}\chi_k = \lambda_n\chi_k$.

To find the i-th eigenvalue $\lambda_{i<n}$, we can first give a pre-guessed value λ_g and transform the above procedures as follows:

$$\mathcal{H}\chi = \lambda_i\mathcal{O}\chi$$
$$(\mathcal{H} - \lambda_g\mathcal{O})\chi = (\lambda_i - \lambda_g)\mathcal{O}\chi$$
$$(\mathcal{H} - \lambda_g\mathcal{O})^{-1}\mathcal{O}\chi = \frac{1}{\lambda_i - \lambda_g}\chi \tag{29}$$

By choosing λ_g close to any one of the eigenvalues λ_i will make $1/(\lambda_i - \lambda_g)$ much bigger than any others. We thus have the sequence

$$\begin{aligned}\chi_m &= \left[(\mathcal{H} - \lambda_g\mathcal{O})^{-1}\mathcal{O}\right]\chi_{m-1} \\ &= \left[(\mathcal{H} - \lambda_g\mathcal{O})^{-1}\mathcal{O}\right]^m\chi_0, \ (m = 1,2,\cdots).\end{aligned} \tag{30}$$

As m increases, χ_m becomes the dominant eigenvector of the operator $(\mathcal{H} - \lambda_g\mathcal{O})^{-1}\mathcal{O}$ with the corresponding eigenvalue λ_i'

$$\chi_{m+1} = \left[(\mathcal{H} - \lambda_g\mathcal{O})^{-1}\mathcal{O}\right]\chi_m = \lambda_i'\chi_m. \tag{31}$$

Multiplying the above equation by the transposed vector $\tilde{\chi}_{m+1}$, we can obtain

$$\lambda_i' = \frac{\tilde{\chi}_{m+1}\chi_{m+1}}{\tilde{\chi}_{m+1}\chi_m}, \tag{32}$$

Finally, we achieve

$$\lambda_i = \lambda_g + \lambda_i' = \lambda_g + \frac{\tilde{\chi}_{m+1}\chi_{m+1}}{\tilde{\chi}_{m+1}\chi_m}. \tag{33}$$

2.3.2. Square Root Method

In order to solve Eq. (31), we need to use the square-root method [45]. For this purpose, we rewrite Eq. (31) in the form

$$A\chi_n = F, \tag{34}$$

where $A = (\mathcal{H} - \lambda_g \mathcal{O})$ and $F = \mathcal{O}\chi_{n-1}$. Since A is symmetric, it can be converted into a multiplication of two triangular matrices

$$A = \tilde{S}S, \tag{35}$$

where S is the upper triangular matrix

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ 0 & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{nn} \end{pmatrix} \tag{36}$$

and $\tilde{S}$ is its transpose. From Eq. (35), we can get these nonzero elements of S

$$s_{11} = \sqrt{a_{11}},$$

$$s_{1j} = \frac{a_{1j}}{s_{11}},$$

$$s_{ii} = \sqrt{a_{ii} - \sum_{\ell=1}^{i-1} s_{\ell i}^2}, (i > 1)$$

$$s_{ij} = \frac{a_{ij} - \sum_{\ell=1}^{i-1} s_{\ell i} s_{\ell j}}{s_{ii}}, (j > i),$$

where a_{ij}s are the elements of A.

Eq. (35), combined with Eq. (34), is equivalent to the following two equations

$$\tilde{S}K = F, \tag{37}$$

$$S\chi_n = K. \tag{38}$$

Matrix elements k_i of K can be determined in terms of s_{ij} and matrix elements of f_i by the following recurrence formulas:

$$k_1 = \frac{f_1}{s_{11}}, \tag{39}$$

$$k_{i>1} = \frac{f_i - \sum_{\ell=1}^{i-1} s_{\ell i} k_\ell}{s_{ii}}. \tag{40}$$

Eventually, the solution for χ_n is given by

$$x_N = \frac{k_N}{s_{NN}},$$

$$x_{i<N} = \frac{k_i - \sum_{\ell=i+1}^{n} s_{i\ell} k_\ell}{s_{ii}}, \tag{41}$$

where N is the dimension of the original matrix $\mathcal{H}$ whose dominant eigenvector has components x_j.

2.4. Construction of Basis

As discussed in the preceding sections, the Rayleigh-Ritz variational method requires the trial wave function to be expanded in terms of a series of basis functions as shown in Eq. (23). The Hylleraas basis is used in this work, so the general term of $|\varphi\rangle$ in Eq. (23) takes the form of

$$|\varphi\rangle = r_1^i r_2^j r_{12}^k \exp(-\alpha r_1 - \beta r_2 - \gamma r_{12})\, \mathcal{Y}_{l_1 l_2 L}^{M}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2), \tag{42}$$

where $\mathcal{Y}_{l_1 l_2 L}^{M}$ is used to denote angular part of the basis, and α, β and γ contained in the exponential term are the non-linear parameters. Inclusion of γ term makes our calculation more involved and time-consuming than the basis functions used by Li *et al.* [34], although both of us use Hylleraas coordinates. But the advantage of our choice is that we can achieve much higher rate of convergence with fewer terms.

However, Eq. (42) is only a primitive form of the basis functions. In the real calculation, to achieve better convergence as well as higher numerical stability, we need to construct a more sophisticated form of basis. For S-states, a modified double basis set [33] [46] is employed. Its basic form is

$$\begin{aligned}&\Psi^{S}(\mathbf{r}_1, \mathbf{r}_2)\\ &= \sum_{p=1}^{2} \sum_{i,j=0}^{i+j\le\Omega_1^{(p)}} \sum_{k=-\Omega_{\text{low}}}^{\Omega_{\text{high}}} a_{ijk}^{(p)} r_1^{i} r_2^{j} r_{12}^{k} \exp\left(-\alpha^{(p)} r_1 \right. \\ &\left. - \beta^{(p)} r_2 - \gamma^{(p)} r_{12}\right) \pm (\text{exchange}),\end{aligned} \tag{43}$$

where the range of k is defined by

$$\Omega_{\text{low}} = N_p - \Omega_1^{(p)} + (i + j),$$

$$\Omega_{\text{high}} = N_p + \Omega_1^{(p)} - (i + j),$$

and $\Omega_1^{(p)}$ is used to control the range of i and j, with the integer $N_p > \Omega_1^{(p)}$ being an adjustable parameter. By the word 'exchange' we mean the interchanging of labels $1 \leftrightarrow 2$. To simulate the vibrational modes of this molecular system [23], we make $N_p \approx 2\gamma^{(p)}$ with $\gamma^{(p)}$ being between 14 and 20. In Cassar's work [33], he optimized this value for different energy levels. In our calculation, we simply pick a fixed value for $N_1 = N_2 =$

37 for all the energy levels. The sign '+' is for nuclear singlet states with the total spin of the two protons being zero while '–' is for triplet states with total spin being one. The way we used to construct a basis set makes each combination of powers i, j, k to be included twice with different nonlinear parameters. One may think that this may lead to a problem of linear dependency, but a complete optimization of the energy with respect to all those nonlinear parameters will end up with a well-defined and numerically stable energy value.

It is easy to conclude from Eq. (43) that as the $\Omega_1^{(p)}$ increases, the size of the basis will become larger, leading to a more and more converged result. However, numerical problems of near-linear dependency in the basis set become progressively severe as $\Omega_1^{(p)}$ increases. To avoid this, for S-states, all terms with $i > j$ are omitted. In addition to that, a truncation over index k is used by avoiding terms with $i + j + |N_p - k| - |l_1 - l_2| + |j - i| > \Omega_1^{(p)}$ [33]. Using these strategies, and by incorporating QD [47] (a package that supports both a double-double data type with about 32 decimal digits and a quad-double data type with about 64 decimal digits) into our code, we do not meet numerical problems until the basis is larger than 4000 terms.

The basis sets are more complicated for the case of higher total angular momentum $L > 0$ because of the multiplicity of possible one-particle angular momenta l_1 and l_2 for a given L. It can be expressed in the form

$$\Psi^{L>0}(\mathbf{r}_1, \mathbf{r}_2) = \sum_{l_1, l_2} \Psi^S(\mathbf{r}_1, \mathbf{r}_2)(l_1, l_2; LM), \tag{44}$$

where the notation

$$(l_1, l_2; LM)$$
$$= r_1^{l_1} r_2^{l_2} \sum_{m_1, m_2} \langle l_1 l_2 m_1 m_2 | LM \rangle Y_{l_1}^{m_1}(\hat{\mathbf{r}}_1) Y_{l_2}^{m_2}(\hat{\mathbf{r}}_2) \tag{45}$$

is used to describe the total angular momentum eigenstates made from vector coupled products of spherical harmonics. In our code, the configurations which are needed for completeness are

$$S \text{ states } (L = 0): \ (0,0;00)$$
$$P \text{ states } (L = 1): \ (0,1;1M)\ (1,0;1M)$$
$$D \text{ states } (L = 2): \ (0,2;2M)\ (1,1;2M)\ (2,0;2M)$$
$$\vdots$$

Here, the 'doubling' of the basis set for states with $L > 0$ is accomplished by including twice the dominant $(0, L;\ LM)$ angular term with different nonlinear parameters. That is to say, we need 3 sectors for *P* states, 4 for *D* states, etc.

2.5. Evaluation of Matrix Elements

2.5.1 Transformation of Operators

In order to solve the general eigenvalue equation presented by Eq. (22), we need to evaluate the matrix elements.

To evaluate the angular part, it is convenient for us to choose spherical coordinates. Accordingly, ∇_1^2 can be written into

$$\nabla_1^2 = \left\{ \frac{1}{r_1^2}[i(i+1) - l_1(l_1+1)] + \frac{k(k+1)}{r_{12}^2} \right.$$
$$- 2\left[\frac{\alpha(i+1)}{r_1} + \frac{\gamma(k+1)}{r_{12}}\right]$$
$$+ \frac{2(r_1 - r_2\cos\theta)}{r_1 r_{12}^2}(k - \gamma r_{12})(i - \alpha r_1) \qquad (46)$$
$$\left. - \frac{2(\hat{\nabla}_1^y \cdot \hat{r}_2) r_2}{r_1 r_{12}^2}(k - \gamma r_{12}) + \alpha^2 + \gamma^2 \right\},$$

where $\hat{\nabla}_1^y$ acts only on angular part. From this equation, ∇_2^2 can be easily obtained by interchanging $1 \leftrightarrow 2$, $i \leftrightarrow j$ and $\alpha \leftrightarrow \beta$.

Unlike the case of helium, where the mass polarization term can be treated as a perturbation due to the small electron-nuclear mass ratio, for the HMI,

$$-\frac{m_p}{1+m_p}\nabla_1 \cdot \nabla_2 \qquad (47)$$

must be calculated in an exact manner. As shown in appendix, this operator can be eventually expanded into

$$-\frac{m_p}{1+m_p}\nabla_1\cdot\nabla_2$$

$$=-\frac{m_p}{1+m_p}\left\{\frac{1}{r_1r_2}\left[ij\cos\theta+i\left(\hat{r}_1\cdot\hat{\nabla}_2^y\right)+j\left(\hat{r}_2\cdot\hat{\nabla}_1^y\right)+\hat{\nabla}_1^y\cdot\hat{\nabla}_2^y\right]\right.$$

$$+\frac{1}{r_1}\left[-\beta\left(i\cos\theta+\hat{r}_2\cdot\hat{\nabla}_1^y\right)\right]+\frac{1}{r_2}\left[-\alpha\left(j\cos\theta+\hat{r}_1\cdot\hat{\nabla}_2^y\right)\right]+\frac{1}{r_{12}}\left[\gamma(i+j+2k+2)\right]$$

$$-\frac{1}{r_{12}^2}\left[k(k+1)+jk+ik\right]+\frac{r_1}{r_{12}}\left[\gamma(\beta\cos\theta-\alpha)\right]+\frac{r_2}{r_{12}}\left[\gamma(\alpha\cos\theta-\beta)\right]$$

$$+\frac{r_2}{r_{12}^2}\left[\beta k-\alpha k\cos\theta\right]+\frac{r_1}{r_2r_{12}}\left[-\gamma\left(j\cos\theta+\hat{r}_2\cdot\hat{\nabla}_1^y\right)\right]+\frac{r_2}{r_1r_{12}}\left[-\gamma\left(i\cos\theta+\hat{r}_2\cdot\nabla_1^y\right)\right]$$

$$+\frac{r_2}{r_1r_{12}^2}\left[k\left(i\cos\theta+\hat{r}_2\cdot\hat{\nabla}_1^y\right)\right]+\frac{r_1}{r_2r_{12}^2}\left[k\left(j\cos\theta+\hat{r}_1\cdot\hat{\nabla}_2^y\right)\right]+\alpha\beta\cos\theta$$

$$\left.-\gamma^2\right\}. \qquad (48)$$

Now that we know the action of all the operators in the Hamiltonian onto various terms in the wave function, we need to evaluate the resulting integrals in order to construct the actual Hamiltonian and overlap matrices. For this purpose, we now focus on the required integrals.

2.5.2 General Integral

In 1978, Drake [48] obtained general expressions for the reduction of a wide class of two electron matrix elements involving Hylleraas coordinates to finite sums of radial integrals. In addition to presenting the main idea of his work, we will show more details on the deduction and more discussion on relations for radial integrals in the following section.

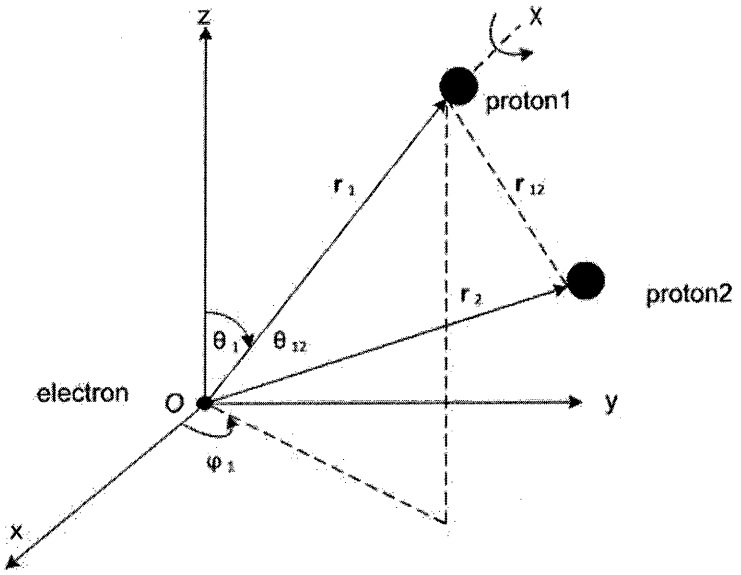

Figure 5 Hylleraas coordinates

Consider the basic form of integral

$$I = \langle \Phi_p' | O | \Phi_p \rangle$$
$$= \iint d\mathbf{r}_1 d\mathbf{r}_2 R_p^* \mathcal{Y}_{l_1' l_2' L'}^{M'}{}^* (\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) \mathcal{T}_{k_1 k_2 K}^{Q} \mathcal{Y}_{l_1 l_2 L}^{M} (\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) R_p, \tag{49}$$

where $\mathcal{Y}_{l_1' l_2' L'}^{M'}$, $\mathcal{Y}_{l_1 l_2 L}^{M}$, and $O = \mathcal{T}_{k_1 k_2 K}^{Q}$ are vector-coupled products of spherical harmonics given by

$$\mathcal{Y}_{l_1, l_2, L}^{M}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = \sum_{m_1 m_2} \langle l_1 l_2 m_1 m_2 | LM \rangle Y_{l_1}^{m_1}(\hat{\mathbf{r}}_1) Y_{l_2}^{m_2}(\hat{\mathbf{r}}_2),$$
$$\mathcal{T}_{k_1, k_2, K}^{Q}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = \sum_{q_1, q_2} \langle k_1 k_2 q_1 q_2 | KQ \rangle Y_{k_1}^{q_1}(\hat{\mathbf{r}}_1) Y_{k_2}^{q_2}(\hat{\mathbf{r}}_2). \tag{50}$$

The function $R_p = R_p(r_1, r_2, r_{12})$ is the correlated radial part of the integral defined by

$$R_p = r_1^i r_2^j r_{12}^k e^{-\alpha r_1 - \beta r_2 - \gamma r_{12}}. \tag{51}$$

The volume element is

$$d\mathbf{r}_1 d\mathbf{r}_2 = r_1 dr_1 r_2 dr_2 r dr \sin\theta_1 \, d\theta_1 d\varphi_1 d\chi, \tag{52}$$

where all the angles and coordinates are labeled in Fig. 5.

Using the formula

$$\begin{aligned} &\langle l_1 l_2 m_1 m_2 | LM \rangle \\ &= (-1)^{l_1 - l_2 + M} \sqrt{2L+1} \begin{pmatrix} l_1 & l_2 & L \\ m_1 & m_2 & -M \end{pmatrix}, \end{aligned} \tag{53}$$

we can expand Eq. (50) into

$$\begin{aligned} \mathcal{Y}^M_{l_1,l_2,L}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) &= \sum_{m_1 m_2} (-1)^{l_1 - l_2 + M} \sqrt{2L+1} \\ &\times \begin{pmatrix} l_1 & l_2 & L \\ m_1 & m_2 & -M \end{pmatrix} Y^{m_1}_{l_1}(\hat{\mathbf{r}}_1) Y^{m_2}_{l_2}(\hat{\mathbf{r}}_2), \\ \mathcal{T}^Q_{k_1,k_2,K}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) &= \sum_{q_1 q_2} (-1)^{k_1 - k_2 + Q} \sqrt{2K+1} \\ &\times \begin{pmatrix} k_1 & k_2 & K \\ q_1 & q_2 & -Q \end{pmatrix} Y^{q_1}_{k_1}(\hat{\mathbf{r}}_1) Y^{q_2}_{k_2}(\hat{\mathbf{r}}_2). \end{aligned} \tag{54}$$

Using $[L]$ to denote $(2L+1)$, and noticing $\mathcal{Y}^{M'}_{l_1' l_2' L'}{}^*(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2)$ is the complex conjugate of $\mathcal{Y}^M_{l_1,l_2,L}(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2)$, by substituting (54) into (49), we can get

$$\begin{aligned} I = &\iint d\mathbf{r}_1 d\mathbf{r}_2 R'^* R \sqrt{(L', K, L)} \\ &\times \sum_{\text{all } m} (-1)^{l_1' + k_1 + l_1 - l_2' - k_2 - l_2 + M' + Q + M} \\ &\times \begin{pmatrix} l_1' & l_2' & L' \\ m_1' & m_2' & -M' \end{pmatrix} \begin{pmatrix} k_1 & k_2 & K \\ q_1 & q_2 & -Q \end{pmatrix} \begin{pmatrix} l_1 & l_2 & L \\ m_1 & m_2 & -M \end{pmatrix} \\ &\times Y^{m_1'}_{l_1'}{}^*(\hat{\mathbf{r}}_1) Y^{m_2'}_{l_2'}{}^*(\hat{\mathbf{r}}_2) Y^{q_1}_{k_1}(\hat{\mathbf{r}}_1) Y^{q_2}_{k_2}(\hat{\mathbf{r}}_2) Y^{m_1}_{l_1}(\hat{\mathbf{r}}_1) Y^{m_2}_{l_2}(\hat{\mathbf{r}}_2). \end{aligned} \tag{55}$$

In (55), 'all m' means summation over all possible values of those magnetic quantum numbers labeled by lower-case letters, and such an abbreviation will be used throughout the following sections. Thus the basic integral can be decomposed into an integral over the radial part times an integral over the angular part. To simplify the angular part, we use

$$Y_l^{m*}(\hat{\mathbf{r}}) = (-1)^m Y_l^{-m}(\hat{\mathbf{r}}), \tag{56}$$

$$\begin{aligned} & Y_l^m(\hat{\mathbf{r}}) Y_{l'}^{m'}(\hat{\mathbf{r}}) \\ &= \sum_{LM} \frac{[l, l', L]^{\frac{1}{2}}}{\sqrt{4\pi}} \begin{pmatrix} l & l' & L \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l & l' & L \\ m & m' & -M \end{pmatrix} Y_L^M(\hat{\mathbf{r}}), \end{aligned} \tag{57}$$

to find out

$$\begin{aligned} & Y_{l_1'}^{m_1'*}(\hat{\mathbf{r}}_1) Y_{k_1}^{q_1}(\hat{\mathbf{r}}_1) Y_{l_1}^{m_1}(\hat{\mathbf{r}}_1) \\ &= \frac{1}{4\pi} \sum_{\lambda_1 t_1 L_1 M_1} (-1)^{-t_1 + m_1' - M} (2\lambda_1 + 1)[l_1, k_1, l_1', L_1]^{\frac{1}{2}} \\ &\times \begin{pmatrix} k_1 & l_1 & \lambda_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_1' & \lambda_1 & L_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} k_1 & l_1 & \lambda_1 \\ q_1 & m_1 & -t_1 \end{pmatrix} \begin{pmatrix} l_1' & \lambda_1 & L_1 \\ -m_1' & t_1 & -M_1 \end{pmatrix} \\ &\times Y_{L_1}^{M_1}(\hat{\mathbf{r}}_1). \end{aligned} \tag{58}$$

We can also find a similar expansion about $Y_{l_2'}^{m_2'*}(\hat{\mathbf{r}}_2) Y_{k_2}^{q_2}(\hat{\mathbf{r}}_2) Y_{l_2}^{m_2}(\hat{\mathbf{r}}_2)$ by setting up mapping relationships like $l_1 \to l_2, l_1' \to l_2'$, etc.

Up to now, the angular part contained in (54) can be expressed as

$$\begin{aligned} & \frac{1}{(4\pi)^2} \sum_{\lambda_1 \lambda_2 L_1 L_2} \sum_{\text{all } m} (-1)^{\text{phase factor}} [L', L, L_1, L_2, K, l_1, l_2, l_1', l_2', k_1, k_2]^{\frac{1}{2}} [\lambda_1, \lambda_2] \\ &\times \begin{pmatrix} k_1 & l_1 & \lambda_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_1' & \lambda_1 & L_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} k_2 & l_2 & \lambda_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_2' & \lambda_2 & L_2 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

$$\times \begin{pmatrix} k_1 & l_1 & \lambda_1 \\ q_1 & m_1 & -t_1 \end{pmatrix} \begin{pmatrix} l_1' & \lambda_1 & L_1 \\ -m_1' & t_1 & -M_1 \end{pmatrix} \begin{pmatrix} k_2 & l_2 & \lambda_2 \\ q_2 & m_2 & -t_2 \end{pmatrix} \begin{pmatrix} l_2' & \lambda_2 & L_2 \\ -m_2' & t_2 & -M_2 \end{pmatrix}$$

$$\times \begin{pmatrix} l_1' & l_2' & L' \\ m_1' & m_2' & -M' \end{pmatrix} \begin{pmatrix} k_1 & k_2 & K \\ q_1 & q_2 & -Q \end{pmatrix} \begin{pmatrix} l_1 & l_2 & L \\ m_1 & m_2 & -M \end{pmatrix}$$

$$\times Y_{L_1}^{M_1}(\hat{\mathbf{r}}_1) Y_{L_2}^{M_2}(\hat{\mathbf{r}}_2), \tag{59}$$

where the 'phase factor'$= l_1' + k + l_1 - l_2' - k_2 - l_2 + M' + Q + M - t_1 + m_1' - M_1 - t_2 + m_2' - M_2$. This can be further simplified by using the identities such as $q_1 + m_1 - t_1 = 0$ and $k_1 + l_1 + \lambda_1 =$ "even number" from the property of 3j-coefficients. However, as shown in (51), the integral over the angular part is not dependent upon solid angle $\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_2$, so we need more transformation by virtue of

$$Y_{L_2}^{M_2}(\theta_2, \varphi_2) = \sum_{M_2'} \mathcal{D}_{M_2 M_2'}^{L_2}{}^{*}(\varphi_1, \theta_1, \chi) Y_{L_2}^{M_2'}(\theta, \varphi), \tag{60}$$

where $\mathcal{D}$ is the matrix elements of the well-known Wigner D function which serves as a representation of SO_3 group [49]. All these angular variables are labeled in Fig 5. We further take into account the relationship between harmonic functions and D functions [50]:

$$Y_{L_1}^{M_1}(\hat{\mathbf{r}}_1) = Y_{L_1}^{M_1}(\theta_1, \varphi_1)$$

$$= (-1)^{M_1} \frac{[L_1]^{\frac{1}{2}}}{\sqrt{4\pi}} \mathcal{D}_{M_1,0}^{L_1}(\varphi_1, \theta_1, \chi), \tag{61}$$

$$\frac{1}{8\pi^2} \int_0^{2\pi} \int_0^{\pi} \int_0^{2\pi} \mathcal{D}_{m_1' m_1}^{j_1 *}(\alpha, \beta, \gamma) \mathcal{D}_{m_2', m_2}^{j_2}(\alpha, \beta, \gamma) d\alpha \sin\beta \, d\beta d\gamma$$

$$= \delta_{m_1', m_2'} \delta_{m_1 m_2} \delta_{j_1 j_2'} \frac{1}{[j_1]}, \tag{62}$$

After integration over the Euler angles, the angular part becomes

$$\sum_{\lambda_1\lambda_2 L_1}\sum_{\text{all } m}\frac{(-1)^{M_1}}{8\pi}[L',L,K,l_1,l_2,l_1',l_2',k_1,k_2]^{\frac{1}{2}}[\lambda_1,\lambda_2]$$

$$\times\begin{pmatrix}k_1 & l_1 & \lambda_1\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}l_1' & \lambda_1 & L_1\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}k_2 & l_2 & \lambda_2\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}l_2' & \lambda_2 & L_2\\ 0 & 0 & 0\end{pmatrix}$$

$$\times\begin{pmatrix}k_1 & l_1 & \lambda_1\\ q_1 & m_1 & -t_1\end{pmatrix}\begin{pmatrix}l_1' & \lambda_1 & L_1\\ -m_1' & t_1 & -M_1\end{pmatrix}\begin{pmatrix}k_2 & l_2 & \lambda_2\\ q_2 & m_2 & -t_2\end{pmatrix}\begin{pmatrix}l_2' & \lambda_2 & L_1\\ -m_2' & t_2 & -M_1\end{pmatrix}$$

$$\times\begin{pmatrix}l_1' & l_2' & L'\\ m_1' & m_2' & -M'\end{pmatrix}\begin{pmatrix}k_1 & k_2 & K\\ q_1 & q_2 & -Q\end{pmatrix}\begin{pmatrix}l_1 & l_2 & L\\ m_1 & m_2 & -M\end{pmatrix}P_{L_1}(\cos\theta), \qquad (63)$$

where $P_{L_1}(\cos\theta)$ is the Legendre polynomial and θ is the angle between vectors $\hat{r}_1$ and $\hat{r}_2$. If we use the orthogonality property of 3j-coefficients, or use the graphic method [51], the following expression

$$\sum_{\text{all } m}(-1)^{M_1}\begin{pmatrix}k_1 & l_1 & \lambda_1\\ q_1 & m_1 & -t_1\end{pmatrix}\begin{pmatrix}l_1' & \lambda_1 & L_1\\ -m_1' & t_1 & -M_1\end{pmatrix}\begin{pmatrix}k_2 & l_2 & \lambda_2\\ q_2 & m_2 & -t_2\end{pmatrix}\begin{pmatrix}l_2' & \lambda_2 & L_1\\ -m_2' & t_2 & -M_1\end{pmatrix}$$

$$\times\begin{pmatrix}l_1' & l_2' & L'\\ m_1' & m_2' & -M'\end{pmatrix}\begin{pmatrix}k_1 & k_2 & K\\ q_1 & q_2 & -Q\end{pmatrix}\begin{pmatrix}l_1 & l_2 & L\\ m_1 & m_2 & -M\end{pmatrix}$$

can be further simplified into

$$(-1)^{L'-M'}\begin{pmatrix}L' & K & L\\ -M' & Q & M\end{pmatrix}\begin{Bmatrix}L' & l_2' & l_1'\\ L_1 & \lambda_1 & \lambda_2\end{Bmatrix}\begin{Bmatrix}l_1 & l_2 & L\\ k_1 & k_2 & K\\ \lambda_1 & \lambda_2 & L'\end{Bmatrix}. \qquad (64)$$

Finally, the basic integral (48) can be transformed into a more manageable form:

$$I=(-1)^{L'-M'}\begin{pmatrix}L' & K & L\\ -M' & Q & M\end{pmatrix}\times\sum_{\lambda_1,\lambda_2,\Lambda}X_{\lambda_1,\lambda_2,\Lambda}D_{\lambda_1,\lambda_2,\Lambda}I_{\Lambda}(R'R), \qquad (65)$$

where we have replaced L_1 by Λ in order to obtain exactly the same result as Drake's [48]. Similarly, we also employ the same definition contained in (65):

$$I_\Lambda(R'R) = \int_0^\infty r_1 dr_1 \int_0^\infty r_2 dr_2 \int_{|r_1-r_2|}^{r_1+r_2} r dr R' R P_\Lambda(\cos\theta), \tag{66}$$

$$X_{\lambda_1,\lambda_2,\Lambda} = \frac{(-1)^{l_1'+l_2'+L'+\Lambda}}{8\pi}[\lambda_1,\lambda_2,\Lambda][L',L,K,l_1,l_2,l_1',l_2',k_1,k_2]^{\frac{1}{2}} \times \begin{pmatrix} k_1 & l_1 & \lambda_1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} l_1' & \lambda_1 & L_1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} k_2 & l_2 & \lambda_2 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} l_2' & \lambda_2 & L_2 \\ 0 & 0 & 0 \end{pmatrix}, \tag{67}$$

and

$$D_{\lambda_1,\lambda_2,\Lambda} = \begin{Bmatrix} L' & l_2' & l_1' \\ \Lambda & \lambda_1 & \lambda_2 \end{Bmatrix}\begin{Bmatrix} l_1 & l_2 & L \\ k_1 & k_2 & K \\ \lambda_1 & \lambda_2 & L' \end{Bmatrix}. \tag{68}$$

2.5.3 Radial Integral

Now, we will evaluate the integral defined by Eq. (66). Let us start with the easiest case when $\Lambda = 0$, where we can define a basic radial integral:

$$I_0(i,j,k) = \int_0^\infty r_1^{i+1} e^{-\alpha r_1} \mathrm{d}r_1 \int_0^\infty r_2^{j+1} e^{-\beta r_2} \mathrm{d}r_2 \int_{|r_1-r_2|}^{r_1+r_2} r_{12}^{k+1} e^{-\gamma r_{12}} \mathrm{d}r_{12}. \tag{69}$$

For the special case of $i = j = k = -1$, the integral reduces to

$$I_0(-1,-1,-1) = \int_0^\infty dr_1 \int_0^\infty dr_2 \int_{|r_1-r_2|}^{r_1+r_2} dr_{12} e^{-\alpha r_1 - \beta r_2 - \gamma r_{12}}, \tag{70}$$

which can be split into two parts:

$$I_0(-1,-1,-1)$$

$$= \int_0^\infty dr_1 \int_{r_1}^\infty dr_2 \int_{r_2-r_1}^{r_1+r_2} dr_{12} e^{-\alpha r_1 - \beta r_2 - \gamma r_{12}}$$

$$+ \int_{r_2}^\infty dr_1 \int_0^\infty dr_2 \int_{r_1-r_2}^{r_1+r_2} dr_{12} e^{-\alpha r_1 - \beta r_2 - \gamma r_{12}}.$$

This one can be easily evaluated to get the result

$$I_0(-1,-1,-1) = \frac{2}{(\alpha+\beta)(\alpha+\gamma)(\beta+\gamma)}. \tag{71}$$

It is obvious from (69) that the following expressions hold

$$\left(-\frac{\partial}{\partial\alpha}\right) I_0(i-1,j,k) = I(i,j,k),$$

$$\left(-\frac{\partial}{\partial\beta}\right) I_0(i,j-1,k) = I(i,j,k), \tag{72}$$

$$\left(-\frac{\partial}{\partial\gamma}\right) I_0(i,j,k-1) = I(i,j,k)$$

We can use (71) as the starting point to obtain higher-power integrals by repeated differentiation:

$$I_0(i,j,k)$$

$$= (-1)^{i+j+k+3} \frac{\partial^{i+1}}{\partial\alpha^{i+1}} \frac{\partial^{j+1}}{\partial\beta^{j+1}} \frac{\partial^{k+1}}{\partial\gamma^{k+1}} \left[\frac{2}{(\alpha+\beta)(\alpha+\gamma)(\beta+\gamma)}\right]. \tag{73}$$

Eventually, by defining $p = i+1, m = j+1$ and $n = k+1$, we will have

$$I_0(i,j,k) = I_0(p-1, m-1, n-1)$$

$$= \sum_{a=0}^{p}\sum_{b=0}^{m}\sum_{c=0}^{n} \frac{2p!\,m!\,n!}{a!\,b!\,c!\,(p-a)!\,(m-b)!\,(n-c)!} \tag{74}$$

$$\times \frac{(a+b)!\,(c+p-q)!\,(n+m-b-c)!}{(\alpha+\beta)^{a+b+1}(\alpha+\gamma)^{c+p-a+1}(\beta+\gamma)^{n+m-b-c+1}}.$$

However, as will be shown below, in order to calculate higher rank of radial integral and also to generate vectors for optimization, we may meet the case when $p = -1$ with positive m and n. In this case, the preceding discussion is not enough. This can be solved by virtue of Harris's work [52]. The basic radial integral we need in this case can be expressed as:

$$\begin{aligned} I_0(-p,m,n) = \left(\frac{1}{\beta+\gamma}\right)[&mI_0(-p,m-1,n) \\ &+ nI_0(-p,m,n-1) \\ &+ G(-p,m,n)], \end{aligned} \tag{75}$$

where

$$\begin{aligned} G(-p,m,n) = &\frac{2m!\,n!\,(m+n-p+1)!}{(m+n+1)!\,(\alpha+\beta)^{n+m-p+2}} \\ &\times {}_2F_1(n+1, m+n-p+2; m+n \\ &+ 1; \tau). \end{aligned} \tag{76}$$

Eq. (76) will be numerically stable provided the hypergeometric function ${}_2F_1$, with $\tau = (\beta-\gamma)/(\alpha+\beta)$, is evaluated in a suitable fashion. It is worthwhile to mention here that other singular cases can only appear in higher rank corrections to the energy, such as relativistic and QED effects, not in the calculation of non-relativistic energy.

To evaluate (66) with $\Lambda \neq 0$, recurrence relations must be studied. For this purpose, Drake generalized (66) into

$$I_\Lambda(g) = \int_0^\infty r_1 \mathrm{d}r_1 \int_0^\infty r_2 \mathrm{d}r_2 f(r_1, r_2) \times \int_{|r_1 - r_2|}^{r_1 + r_2} r \mathrm{d}r g(r) P_\Lambda(\cos\theta), \tag{77}$$

where $f(r_1, r_2)$ and $g(r)$ are any functions for which the integral exists. If $g(r) = r^{c+2}$ with $c \neq -2$, by using the recurrence relation of Legendre polynomial

$$P_\Lambda(x) = [P'_{\Lambda+1}(x) - P'_{\Lambda-1}(x)]/(2l+1), \tag{78}$$

Drake found that

$$I_{l+1}(r^c) = \frac{2l+1}{c+2} I_l(r_1^{-1} r_2^{-1} r^{c+2}) + I_{l-1}(r^c). \tag{79}$$

The values needed to start the recurrence relation are

$$I_1(r_1^a r_2^b r^c) = \frac{1}{2}\left[I_0(r_1^{a+1} r_2^{b-1} r^c) + I_0(r_1^{a-1} r_2^{b+1} r^c) - I_0(r_1^{a-1} r_2^{b-1} r^{c+2})\right]. \tag{80}$$

However, this idea cannot be applied here directly, because in our case $g(r) = r^{c+2} e^{-\gamma r}$. Differentiation of this $g(r)$ will lead to two terms, which will bring us a more complicated situation. Therefore, instead of using (78), we use

$$(l+1)P_{l+1}(x) - (2l+1)xP_l(x) + lP_{l-1}(x) = 0, \tag{81}$$

so that from which the recurrence relation we will use is

$$I_{l+1}(g) = \frac{2l+1}{2(l+1)}[I_l(r_1 r_2^{-1} g) + I_l(r_1^{-1} r_2 g) - I_l(r_1^{-1} r_2^{-1} r^2 g)] - \frac{l}{l+1} I_{l-1}(g). \quad (82)$$

3. Optimization

To achieve a better efficiency, it is necessary to have a complete optimization of the energy E with respect to all nonlinear parameters for each basis set size. For this purpose, the Newton-Raphson method is used in our code [53]. If the wave function Ψ has been normalized, the first-order derivatives we need can be calculated according to

$$\frac{\partial E}{\partial \alpha^{(p)}} = -2\langle \Psi | H | r_1 \psi(\mathbf{r}_1, \mathbf{r}_2; \alpha^{(p)}) \pm r_2 \psi(\mathbf{r}_2, \mathbf{r}_1; \alpha^{(p)}) \rangle,$$

$$\frac{\partial E}{\partial \beta^{(p)}} = -2\langle \Psi | H | r_1 \psi(\mathbf{r}_1, \mathbf{r}_2; \beta^{(p)}) \pm r_2 \psi(\mathbf{r}_2, \mathbf{r}_1; \beta^{(p)}) \rangle,$$

$$\frac{\partial E}{\partial \gamma^{(p)}} = -2\langle \Psi | H | r_1 \psi(\mathbf{r}_1, \mathbf{r}_2; \gamma^{(p)}) \pm r_2 \psi(\mathbf{r}_2, \mathbf{r}_1; \gamma^{(p)}) \rangle, \quad (83)$$

where $\psi(\mathbf{r}_1, \mathbf{r}_2; \alpha^{(p)})$ denotes a term in Ψ that depends explicitly on $\alpha^{(p)}$, and parameter p may have different ranges according to which configuration has been used as described in Section 4. These derivatives contain no contribution from an implicit dependence of the energy on the nonlinear parameters through the wave function coefficients $a_{ijk}^{(p)}$ as implied in Eq. (19). This indicates that the energy is stable with respect to first-order variations of the wave function coefficients.

The second-order derivatives with respect to these nonlinear parameters can be calculated approximately using finite differencing of these known first-order derivatives. The code will quit the iteration as long as a specified criterion is fulfilled.

Chapter 3 Details in Programming

3.1 Structure and New Feature of this Code

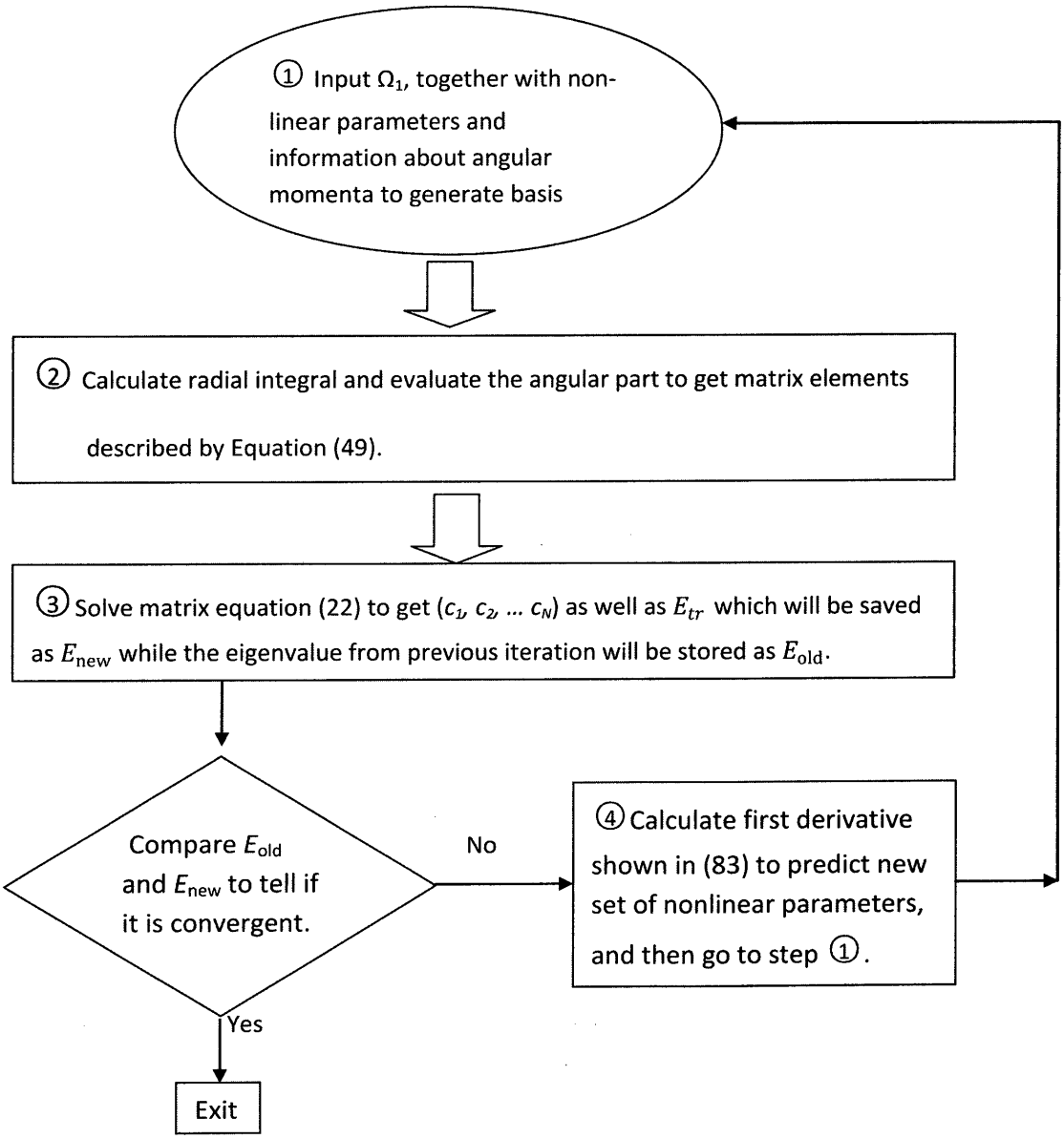

Figure 6 Basic structure of the code

Fig. 6 presents a skeleton of the computer program which is written in FORTRAN 90. The whole computational process is composed of four parts. The first step is to generate a basis according to the scheme described in Section 2.4. The second step is to calculate the necessary matrix elements described by (49). In this step, to avoid the vast repetition of the basic radial integral presented by Eq. (69), a strategy must be employed here. For this purpose, we declared a four-dimensional real array accompanied by a logical array of the same shape, and the logical array shall be initialized as 'false'. In these two types of arrays, the first three dimensions are used to hold the information of power indices and the fourth dimension is used to tell the different combination of nonlinear parameters. Once a certain basic radial integral is required, we retrieve information in the logical array first. If it is 'false', this integral is evaluated and the result is stored in the array while the logical value will be updated to be 'true'. Otherwise, the value needed can be retrieved directly from the real array used for storage. We refer to this strategy as '*store and retrieve*', and it helps to improve the computational efficiency dramatically. The third step is to solve the generalized eigenvalue problem to obtain the eigenvalues as well as the eigenvectors. The last step is to calculate the first-order derivative of energy with respect to those nonlinear parameters and perform the Newton-Raphson algorithm if it is necessary.

Some new features of this free-formed FORTRAN program, such as *Allocatable Array*, *Derived Data Type* and *Module* are widely used throughout the code [54] [55]. These techniques make the code more efficient, more readable and better organized. For example, in our code, we defined a new data type named BASE, which is:

```
type :: base
  integer :: index1
  integer :: index2
  integer :: index3
  integer :: L1
  integer :: L2
  real*16 :: alpha
  real*16 :: beta
  real*16 :: gamma
end type
```

Figure 7 Illustration of derived data type

From this figure, we can see all the information of a basis described by (41) has been included in this new type of data. The three integers index1, index2 and index3 are used to denote those power indices *i, j* and *k*, respectively; those three real numbers correspond to the non-linear parameters in the base; L1 and L2 are used to represent angular momenta of the two particles. By doing so, any procedure that needs this information can use a dummy variable of type (BASE) without writing a lengthy list of variables. Furthermore, an allocatable array can be set up with respect to this derived-type. Truncation criteria together with other schemes described in Section 4 can be used to figure out how long this derived-array will be, and then allocate an array of such length to hold all the information of the basis. This array should be released after usage to save memory. As also can be seen from the figure, the setup of user-defined data is bulky, so it is customary to combine this technique with another one —*Module*. All new data types and parameters can be claimed and sealed in one module, and the module should be given a name, say, **SHARE**. Therefore, other procedures can get access to these definitions and

parameters by writing a sentence '**USE SHARE**' at the beginning. This technique gives rise to a possibility of abolishing the traditional usage of '*COMMON BLOCK*' and thus provides a safer way of delivering data and convenience of future maintenance. Besides this, since '*COMMON BLOCK*' is not compatible with the '*Allocatable Array*', data stored in an allocatable array cannot be shared between different routines by virtue of '*COMMON BLOCK*'. It is worthwhile to end up this paragraph by quoting the comments on 'MEMORY ALLOCATION FOR DERIVED DATA TYPES' from [55]:

> "*When a FORTRAN compiler allocates memory for a variable of a derived data type, the compiler is NOT required to allocate the elements of the derived data type in successive memory locations. Instead, it is free to place them anywhere it wants, as long as the proper element order is preserved during I/O operations. This freedom was deliberately built into the FORTRAN 95 and FORTRAN 2003 standards to allow compilers on massively parallel computers to optimize memory allocations for the fastest possible performance.*"

'To pull a bigger wagon, it is easier to add more oxen than to grow a gigantic ox', that is one of the basic factors of stimulating the development of parallel computation. There are mainly two parallel computational models, one is OpenMP (Open Multi-Processing), and the other is MPI (Message-Passing Interface) [56]. A brief comparison of those two models is displayed in the following table [57]:

Table 2: Comparison between MPI and OpenMP

	OpenMp	MPI
Pros	1.easier to program and debug than MPI; 2.directives can be added incrementally--gradual parallelization; 3. can still run the program as a serial code; 4. serial code statements usually don't need modification; 5. code is easier to understand and maybe more easily maintained.	1. runs on either shared or distributed memory architectures; 2. can be used on a wider range of problems than OpenMP; 3. each process has its own local variables; 4. distributed memory computers are less expensive than large shared memory computers.
Cons	1. can only be run in shared memory computers; 2. requires a compiler that supports OpenMP; 3. mostly used for loop parallelization.	1. requires more programming changes to go from serial to parallel version; 2. can be harder to debug; 3. performance is limited by the communication network between the nodes.

Since OpenMP can only be used on shared memory computers, we adopt the MPI mode in our code for more flexibility. Two parts in our code need to be parallelized, one is the calculation of those matrix elements and the other is for solving the generalized eigenvalue problem. A straightforward idea for the first part is to divide the whole matrix into several parts of almost the same size, and distribute them by a '*master*' CPU to '*slave*' CPUs. Once the calculation is done, the results shall be collected by the '*master*' to form a complete matrix. However, in practice, in order to take full advantage of 'store and retrieve' strategy, each CPU should evaluate matrix elements with unique combination of nonlinear parameters since the memory is not shared in MPI mode. Another parallelism scheme, which is a little more complicated than the former one, has

also been performed upon this part. We noticed that the values of those basic radial integrals will be used twice, one is for the evaluation of matrix elements in Step 2, and the other is for the evaluation of the first-order derivatives in Step 4. According to the traditional way of programming, values obtained in Step 2 will be forgotten and cannot be used in Step 4, thus causing a waste of time and resources. Therefore we began to think about the possibility of calculating and storing those values in advance and parallelizing this calculation. However, the difficulty we noticed is that not all the possible combinations of (i, j, k) satisfying $i+j+k \leqslant \Omega$ or other criteria would be needed. This problem can be solved by generalizing the spirit of 'store and retrieve'. We can develop a new set of code that runs without real calculation to determine which basic integrals will be needed, and this information can be stored in a logical array for future judgment. Such a logical array for determination needs to be formed for only one time and can be used in the rest of the iterations. Armed with such a logical array, we need not worry about the waste of calculation any more.

3.2 More Details in Parallelization of the Power Method

Now, let's turn to the parallelization of the eigenvalue-equation-solving part. As described in Sections 3.1 and 3.2, the power method incorporating the Cholesky decomposition is employed here. It has been pointed out [58] that the CPU time for this part will be proportional to n^3, where n is the dimension of the matrix. When n reaches around 5000, the CPU time in this part is considerable. For this reason, parallel programming is necessary. At first glance, one might think it is impossible to parallelize this part since there is causality between the steps in the Cholesky decomposition.

However, a close investigation will show us a possibility of designing such a parallel algorithm. Let's do it on a simple case. Suppose we are going to use 4 CPUs, which will be labeled by $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$ and $\mathbb{D}$ individually, to decompose a 4×4 symmetric real matrix like:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{pmatrix}.$$

Without losing the generality, each CPU can take charge of a decomposition of one column. For simplicity, we presume that the information on matrix A can be shared by all the CPUs from the beginning.

According to the rules $s_{11} = \sqrt{a_{11}}$ and $s_{1j} = \frac{a_{1j}}{s_{11}}$, we can see once s_{11} is obtained by $\mathbb{A}$ and shared by the others in the form of message passing, s_{1j} with $2 \leq j \leq 4$ can be calculated by the others in a *parallel* manner, resulting in the decomposition of the first row. For the second step, by using $s_{22} = \sqrt{a_{22} - s_{12}^2}$, $\mathbb{B}$ can get s_{22} and then sends it to $\mathbb{C}$ and $\mathbb{D}$. After that, $\mathbb{C}$ and $\mathbb{D}$ can get s_{23} and s_{24} by using $s_{23} = \frac{a_{23} - s_{12}s_{13}}{s_{22}}$ and $s_{24} = \frac{a_{24} - s_{12}s_{14}}{s_{22}}$, respectively. At this step, $\mathbb{A}$ is not involved. Similar things happen in the third step, where $\mathbb{C}$ is to calculate principal element and $\mathbb{D}$ to calculate s_{34} while $\mathbb{A}$ and $\mathbb{B}$ are absent. For the last step, only $\mathbb{D}$ is used to calculate the last element. From the analysis of this heuristic example, we can confirm the possibility of parallelization as well as some features for future algorithm. Firstly, this decomposition will be performed progressively from the first row to the last row, and each new element depends only on the principal

element of its row and the results from previous steps. Secondly, those CPUs will quit the job one by one, so that the total CPU time is decided by the last one.

Based on the same principle outlined above, different versions have been developed according to different ways of allocating jobs and sharing information [59]. In our code, an MPI version [55] has been adopted. As is well-known, the Cholesky decomposition can only deal with a positive-defined matrix unless some modification is included [45]. For this reason, the introduction of this MPI algorithm should start with a modified serial edition.

Unlike the previous example, the following decomposition will be performed and stored on a lower-triangle matrix. In practice, in order to save the memory, we would like to update the matrix A step by step with the results of decomposition rather than declaring another individual matrix.

001 do $j = 1, n$
002 $A(j,j) = A(j,j) - SDOT(j, A_j, A_j)$
003 if $A(j,j) \geq 0, Sign(j) = +1$
004 else $A(j,j) < 0, Sign(j) = -1$
005 $A(j,j) = SQRT\left(ABS(A(j,j))\right)$
006 do $i = j + 1, n$
007 $A(i,j) = \left(A(i,j) - SDOT(j, A_j, A_i)\right)/A(j,j)$
008 end do
009 end do

Figure 8 Serial algorithm of Cholesky decomposition

In the above figure, $SDOT(k, A_i, A_j)$ denotes $\sum_{l=1}^{k-1} A(i,l) * A(j,l)$, $SQRT(x)$ means the square root of x and $ABS(x)$ is the absolute value of x. Modification of the traditional

Cholesky decomposition is embodied by the introduction of an integer array named $Sign(j)$ in the line 003 and 004. The element of the array will be set to -1 in case the principal element $A(j,j) < 0$; otherwise it will be +1. Information contained in this integer array is indispensable in the stage of back substitution; after that, it can be discarded.

Before we begin to discuss the parallel edition, we need to describe how the matrix is stored on the CPUs as well as some other conventions. Suppose the super-computer and its networks have a ring-shaped architecture with P_i $(i = 0, \cdots, p-1)$ to denote the ith CPU, as shown in Figure 9.

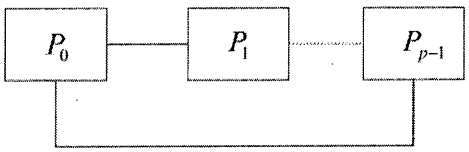

Figure 9 Data stored in the CPUs in a wrap manner

Data will be distributed upon these CPUs in a manner called *WRAP*; that is to say, A_i, which is the ith row in matrix A, will be stored in CPU $P_{i \bmod p}$, where "mod" is an operation used in computing to find the remainder of division of one number by another.

```
001  do i = 0, m − 1
002       k = i * p + myid; l = k − p + 1
003       isgm(k + 1) = 1
004  if myid ≠ master, then
005     Receive G and isgg from P(myid − 1)
006  end if
007  do j = 0, p − 2
```

```
008      A(i + 1, j + l + 1) = (A(i + 1, j + l + 1) −
009          − SDOT(j + l + 1, A_{i+1}, G_{j+1}))
010     F_{j+1} = G(j + 2)
011     isgm(j + l + 1) = isgg(j + 1)
012     isgf(j + 1) = isgg(j + 2)
013   end do
014       sum = SDOT(k, A_i, A_i, isgm)
015   if  sum > 0, A_{i+1} = SQRT(sum)
016   if  sum < 0, A_{i+1} = SQRT(−sum), isgm(k + 1) = −1
017       F_{p−1} = A_{i+1}
018       isgf(p − 1) = isgm(k + 1)
019       send F to P_{myid+1}
020   do t = i + 1, m − 1
021   do j = 0, p − 2
022       A(t + 1, j + l + 1) = (A(t + 1, j + l + 1) −
023         SDOT(j + l, A_{t+1}, G_{j+1}, isgm)/G(j, j + l + 1)
024       A(t + 1, k + 1) = (A(t + 1, k + 1) −
025         SDOT(k, A_{t+1}, A_{i+1}, isgm)/A(i + 1, k + 1)
026   end do
027   end do
```

Figure 10 The MPI parallel algorithm for Cholesky decomposition

In Fig. 10, on the basis of the previous serial algorithm, a parallel algorithm is developed. This one will be performed identically on the entire set of slave CPUs since master-slave mode is used. Here, $SDOT(j + l, A_{t+1}, G_{j+1}, isgm)$ denotes $\sum_{i=1}^{j+l-1} A(t + 1, i) * G(j + 1, i) * isgm(i)$, using $isgm(i)$ as the counterpart of $Sign(i)$ in the serial edition to hold the information of sign in the course of decomposition. However, $isgm(i)$ is introduced symbolically for the convenience of expression and discussion. In practice,

the functionality of $isgm(i)$ is carried out by $isgg(i)$ and $isgf(i)$, where $isgg(i)$ holds the information sent from other CPUs while $isgf(i)$ contains the information to be sent to others. More details on the problem of sign can be seen in the code in Appendix B. Similarly, we use G to denote the factors received from other CPUs and F to send factors to the others.

Chapter 4 Results and Discussion

A set of parallel Fortran 90 codes has been developed independently to calculate non-relativistic energy of the HMI. A benchmark calculation of its ground state energy has been achieved. The energy levels of states $H_2^+(1\ ^1S)$, $H_2^+(2\ ^1S)$, $H_2^+(2\ ^3P)$ were calculated and presented in Tables 3-5. In these tables, the "Ratio" is defined by

$$R(\Omega) = \frac{E(\Omega-2) - E(\Omega-1)}{E(\Omega-1) - E(\Omega)}, \tag{84}$$

which serves as an indicator for the rate of convergence, as the size of basis set increases progressively. Ideally, if $R(\Omega)$ were a constant, then the series would converge as a geometric series, to the value

$$E(\infty) = E(\Omega_{\max}) + \frac{E(\Omega_{\max}) - E(\Omega_{\max}-1)}{R-1}. \tag{85}$$

In practice, we can still get an extrapolated energy in the limit $\Omega \to \infty$ provided R varies smoothly enough with Ω [60]. In these tables, $\Omega = N_p + \Omega_l$, and N_p has been used as another parameter to optimize the base in Cassar's work [61]. However, in this work, since we have used QD with a bigger base than his, we did not try to optimize this parameter and merely used a fixed value N_p=37 throughout the calculations for all the states. The extrapolated results were obtained via Eq. (85) and displayed in the last line at the tables. All figures converged are highlighted in bold face. We can see from these tables that the energy eigenvalue of the ground state $1\ ^1S$ has been calculated to a precision of about 2 parts in 10^{34}. For the first excited state of $2\ ^1S$, our result for the energy eigenvalue is accurate to about 2 parts in 10^{31}. Finally, for the state of $2\ ^3P$, the

lowest state of P symmetry, the present value for the energy eigenvalue is accurate to about 3 parts in 10^{29}.

In Table 6, we present a comparison of our values for the $1\ ^1S$, $2\ ^1S$ and $2\ ^3P$ states with other published results. For the sake of consistency in comparison, we adopted the older value of the proton-electron mass ratio $m_p/m_e = 1836.152\ 701$. It should be mentioned that Table 6 only lists the results published in the last two decades for comparison.

For the $1\ ^1S$ state, we can see that the result of Cassar and Drake [33] seems to converge in a wrong direction, after the 21st digit, which may imply the existence of problem of numerical instability in their calculation. On the other hand, the results of Li *et al.* [34], Hijikata *et al.* [24], and the present work agree perfectly well with each other till the 29th decimal place. Although Hijikata *et al.* have claimed that their result for the ground state has 31 digits of precision (not including the digit "0" before decimal point), a comparison with the present calculation shows that their value is only accurate up to the 29th digit. Therefore, our result for the ground state energy has improved the value of Hijikata *et al.* by more than 3 orders of magnitude, which represents the most precise value for this state up to now.

As for the first excited state $2\ ^1S$, our calculation has improved the best previous value of Li *et al* [34], which is accurate to 6 parts in 10^{28}, by a factor of 3×10^3. The corresponding value of Hijikata *et al.* [24], which is accurate to about 1 part in 10^{26}, is absent from the table for a direct comparison because they used a different proton-electron mass ratio for this state.

Finally, for 2 ^3P, the present calculation has improved over the best previous value of Li *et al.* [34] by a factor of 2.5×10^2 and over the second best previous value of Hijikata *et al.* [24] by a factor of 5×10^3.

Table 3: Convergence pattern for the 1^1S state of the HMI.

Ω_1	N	Eigenvalues	Ratio
3	32	–0.**5971**389	
4	58	–0.**5971390**620	
5	94	–0.**5971390**630	81.45
6	144	–0.**597139063**122	20.93
7	208	–0.**5971390631**233	66.13
8	290	–0.**59713906312340**	35.20
9	390	–0.**59713906312340**506	36.00
10	512	–0.**597139063123405**745	56.33
11	656	–0.**59713906312340**7482	34.94
12	826	–0.**597139063123405**74833	29.68
13	1022	–0.**5971390631234050**7483412	31.52
14	1248	–0.**59713906312340**5074834133	73.58
15	1504	–0.**597139063123405**0748341340	15.17
16	1794	–0.**5971390631234050**7483413409	81.46
17	2118	–0.**59713906312340**50748341340960	66.59
18	2480	–0.**597139063123405**074834134096025	76.81

19	2880	–0.**597139063123405074834134096026**17	1.95
20	3322	–0.**5971390631234050748341340960261**8	27.81
21	3806	–0.**59713906312340507483413409602618**98	12.64
Extrap.	∞	–0.**597139063123405074834134096026189**9(1)	

Table 4: Convergence pattern for the 2^1S state of the HMI.

Ω_1	N	Eigenvalues	Ratio
2	16	–0.**587**111	
3	32	–0.**58715**515	
4	58	–0.**58715**558	99.67
5	94	–0.**5871556**785	4.86
6	144	–0.**58715567**919	130.9
7	208	–0.**58715567921**234	41.85
8	290	–0.**587155679212**728	42.86
9	390	–0.**58715567921274**649	21.39
10	512	–0.**587155679212746**804	58.43
11	656	–0.**58715567921274681**210	40.84
12	826	–0.**587155679212746812**208	72.53
13	1022	–0.**5871556792127468122**1176	36.04
14	1248	–0.**58715567921274681221**1876	24.73
15	1504	–0.**587155679212746812211881**04	24.94

16	1794	–0.**587155679212746812211881**85	34.19
17	2118	–0.**58715567921274681221188119**300	17.15
18	2480	–0.**587155679212746812211881193**129	61.54
19	2880	–0.**587155679212746812211881193**139	13.51
20	3322	–0.**587155679212746812211881193**1402	11.28
Extrap.	∞	–0. **587155679212746812211881193**1403(1)	

Table 5: Convergence pattern for the 2^3P state of the HMI.

Ω_1	N	Eigenvalues	Ratio
3	45	–0.**5968**705	
4	84	–0.**596873**730	
5	138	–0.**596873**738	377.2
6	213	–0.**59687373**881	40.84
7	309	–0.**5968737388**325	11.70
8	432	–0.**596873738832**749	99.88
9	582	–0.**59687373883276**441	11.59
10	765	–0.**596873738832764**724	11.59
11	981	–0.**59687373883276473**584	27.43
12	1236	–0.**596873738832764735**918	150.8
13	1530	–0.**5968737388327647359206**3	41.82
14	1869	–0.**59687373883276473592073**5	17.67

15	2253	−0.**596873738832764735920743**7	12.42
16	2688	−0.**596873738832764735920744**87	7.21
17	3174	−0.**596873738832764735920744**969	12.06
18	3717	−0.**596873738832764735920744983**9	6.45
19	4317	−0.**596873738832764735920744985**04	15.08
20	5706	−0.**596873738832764735920744985**09	19.22
Extrap.	∞	−0. **596873738832764735920744985**1(2)	

Table 6: Comparison of the non-relativistic energies of the HMI in the 1 ^1S, 2 ^{1}S and 2 ^3P states.

Author (Year)	Ref.	Energy
1 ^1S		
Taylor *et al.* (1999)	[31]	−0.597 139 063 123 9(5)
Korobov (2000)	[14]	−0.597 139 063 123 405 074
Hilico *et al.* (2000)	[16]	−0.597 139 063 123 40(1)
Bailey and Frolov (2002)	[20]	−0.597 139 063 123 405 074 83
Yan *et al.* (2003)	[21]	−0.597 139 063 123 405 074 5(4)
Cassar and Drake (2004)	[22]	−0.597 139 063 123 405 074 834 338(3)
Li *et al.* (2007)	[34]	−0.597 139 063 123 405 074 834 134 096 026(5)
Hijikata *et al.* (2009)	[24]	−0.597 139 063 123 405 074 834 134 096 025 974 142
This work		−0.597 139 063 123 405 074 834 134 096 026 189 9(1)
2 ^1S		
Taylor *et al.* (1999)	[31]	−0.587 155 679 213 6(5)
Hilico *et al.* (2000)	[16]	−0.587 155 679 212 75(1)
Cassar and Drake (2004)	[33]	−0.587 155 679 212 746 812 212(2)

Li *et al.*(2007)	[34]	−0.587 155 679 212 746 812 211 881 193 5(6)
This work		−0.587 155 679 212 746 812 211 881 193 140 3(1)
$2\,^3P$		
Taylor *et al.* (1999)	[31]	−0.596 873 738 832 8(5)
Hilico *et al.* (2000)	[16]	−0.596 873 738 83(1)
Yan *et al.* (2003)	[21]	−0.596 873 738 832 764 733 (1)
Cassar and Drake (2004)	[33]	−0.596 873 738 832 764 734 96(5)
Li *et al.* (2007)	[34]	−0.596 873 738 832 764 735 920 744 98(2)
Hijikata *et al.* (2009)	[24]	−0.596 873 738 832 764 735 920 744 893
This work		−0.596 873 738 832 764 735 920 744 985 11(2)

Chapter 5 Suggestions for Future Work

In this thesis work, we have developed a set of well tested MPI Fortran codes incorporated with the multi-precision software QD to perform high-precision calculations of the non-relativistic energy eigenvalues of the hydrogen molecular ion H_2^+ in some low-lying states. The established results for 1^1S, 2^1S and 2^3P states can serve as a benchmark for other theoretical methods. The current method can in principle be applied to calculate other excited states with different rovibrational quantum numbers. Furthermore, the extension of the current work to other isotopomers of H_2^+ can readily be made, with some slight modifications of basis parameters.

The present work has also laid a sound foundation for exploring relativistic and QED effects in H_2^+. Once we have high-precision wave functions, these effects can be evaluated using the perturbation method. Although some work in this direction has been done [62] [63], an independent check for previous calculations is always welcomed. At present, a complete study of higher-order relativistic and QED effects of $O(\alpha^4)$ and $O(\alpha^5)$ is underway by Korobov and coworkers [64] using quantum field theory. Once the corresponding operators in non-relativistic form that represent these effects have been derived, we can apply our high quality wave functions to perform numerical evaluation. It is typical that these higher order operators are singular in nature; therefore, high quality wave functions are necessary to guarantee numerically stable results.

Bibliography

[1] S. G. Karshenboim and V. B. Smionov Lect. Notes Phys **627** 1-12 (2003) (Springer-Verlag, Berlin Heidelberg 2003).

[2] S. G. Karshenboim *Physics Reports* **422** 1 – 63 (2005).

[3] J.-P. Uzan, *Rev. Mod. Phys.* **75** 403 (2003).

[4] B. Roth et al.: *Precision Spectroscopy of Molecular Hydrogen Ions*, Lect. Notes Phys. **745**, 205 - 232 (2008).

[5] D. M. Bishop and Lap M. Cheung *Phys. Rev. A.* **16** 640 (1977).

[6] R. S. Hayano et al. *Rep. Prog. Phys.* **70** 1995–2065 (2007).

[7] E. A. Hylleraas and B. Undheim, *Z. Phys.* **65**, 759 (1930).

[8] T. Kinoshita, *Phys. Rev.* **105**, 1490 (1957).

[9] C. Schwartz, *Int. J. Mod. Phys* **E15**; 877 (2006); arXiv:math-ph/0605018.

[10] J. J. Thomson, *Phil. Mag.* **13**, 561 (1907).

[11] A. Carrington, I. R. McNab, C. A. Montgomerie *J. Phys. B.* **22** 3551-3586 (1989).

[12] W. Demtröder *Atom, Molecules and Photons* (Springer ISBN-103-540-20631-0).

[13] J.-P. Grivet *J. Chem. Educ.* **79** 127 (2002).

[14] V. I. Korobov, *Phys. Rev. A* **61** 064503 (2000).

[15] D. H. Bailey, *ACM Trans. Math. Softw.* **19** 288 (1993); **21** 379 (1995).

[16] L. Hilico, N. Billy et al. *Eur. Phys. J. D* **12** 449 (2000).

[17] A. Messiah, *Quantum Mechanics* (Dunod, Paris, 1964), Vol. 2.

[18] V. I. Korobov, *Phys. Rev. A* **63**, 044501 (2001).

[19] A. M. Frolov, *J. Phys. B*, **35**, L331 (2002).

[20] D. H. Bailey, A. M. Frolov, *J. Phys. B* **35**, 4287 (2002).

[21] Z.-C. Yan, J.-Y. Zhang and Y. Li, *Phys. Rev. A* **67** 062504 (2003).

[22] A. K. Bhatia, *Phys. Rev. A* **58**, 2787 (1998).

[23] A. K. Bhatia and R. J. Drachman, *Phys. Rev. A* **59**, 205 (1999).

[24] Y. Hijikata *et al.*, *J. Chem. Phys.* **130** 024102 (2009).

[25] O. Pilón and D. Baye, *J. Phys. B* **45** 065101 (2012).

[26] H. Kleindienst, D. Hoppe, *Theor. Chim. Acta,* **70**, 221 (1986).

[27] A. M. Frolov, *et al.*, *J. Phys. B,* **28**, L449 (1995).

[28] T. K. Rebane, A. V. Filinsky, *Phys. At. Nucl.,* **60**, 1816 (1997).

[29] F. A. de Saavedra, *et al.*, *Eur. Phys. J. D,* **2**, 181 (1998).

[30] B. Grémaud, D. Delande, N. Billy, *J. Phys. B,* **31**, 383 (1998).

[31] J. M. Taylor, Z.-C. Yan, A. Dalgarno, J. F. Babb, *Mol. Phys.,* **97**, 25 (1999).

[32] R. E. Moss, *J. Phys. B,* **32**, L89 (1999).

[33] M. M. Cassar, G. W. F. Drake, *J. Phys. B,* **37**, 2485 (2004).

[34] H. Li, *et al*, *Phys. Rev. A,* **75**, 012504 (2007).

[35] S. Chu, *et al*, *Phys. Rev. Lett.,* **57**, 314 (1986).

[36] B. Roth and S. Schiller, in *Cold Molecules: Theory, Experiment, Applications*, edited

by R. V. Krems, W. C. Stwalley, B. Friedrich. (CRC Press Taylor & Francis Group ISBN 978-1-4200-5903-8).

[37] P. Blythe, *Phys. Rev. Lett,* **95**, 183002 (2005).

[38] A. Ostendorf, *SympathetischeKühlung von Molekülionen durch lasergekühlte Bariumionen in einer linearen Paulfalle,* Ph.D. thesis, Heinrich-Heine University Düsseldorf, 2006.

[39] S. Schiller, C. Lammerzahl, *Phy. Rev. A,* **68**, 053406 (2003).

[40] H. Goldstein, *Classical Mechanics*, (Addison-Wesley, Massachusetts, 1983), Second Edition.

[41] *Atomic, Molecular, and Optical Physics Handbook*, edited by G. W. F. Drake (AIP Press, New York, 1996).

[42] B. Ford, G. Hall, *Comp. Phys. Comm,* **8**, 337 (1974).

[43] T. Ericsson, A. Ruhe, *Mathematics of Computation,* **35**, 1251 (1980).

[44] A. V. Mitin, *J. Comp. Phys.,* **161**, 653 (2000).

[45] D. K, Faddeev and v. N. Faddeeva, *Computational Methods of Linear Algebra*, (W. H. Freeman, San Francisco, 1963).

[46] G. W. F. Drake, in *Long Range Casimir Forces: Theory and Recent Experiments on Atomic Systems*, edited by F. S. Levin and D. A. Micha (Plenum Press, New York, 1993).

[47] QD is available at http://crd.lbl.gov/$\backsim$dhbailey/mpdist.

[48] G. W. F. Drake, *Phys. Rev. A,* **18**, 820 (1978).

[49] W.-K. Tung, *Group Theory in Physics* (World Scientific, Philadelphia 1985).

[50] A. R. Edmonds, *Angular Momentum in Quantum Mechanics* (Princeton University Press, 1957).

[51] R. N. Zare, *Angular Momentum: Understanding Spatial Aspects in Chemistry and Physics* (Wiley-Interscience 2013).

[52] F. Harris, A. M. Frolov, V. H. Smith, *J. Chem. Phys.*, **121**, 6323 (2004).

[53] *Numerical Recipes*, edited by W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery (Press Syndicate of the University of Cambridge 2002).

[54] M. Metcalf, J. Reid and M. Cohen, *Fortran 95/2003 Explained* (Oxford University Press, 2004).

[55] S. J. Chapman, *Fortran95/2003 for Scientists and Engineers* (McGraw-Hill Primis, ISBN 0-390-91197-6).

[56] W. Gropp, E. Lusk and A. Skjellum *Using MPI second edition* (The MIT press, 1999).

[57] http://www.dartmouth.edu/~rc/classes/intro_mpi/parallel_prog_compare.html.

[58] L.-M. Wang, *High Precision Calculations for Low-lying States of Lithium Atom*, (Wuhan University, PhD thesis).

[59] X.-B. Chi, *Comput. Math.*, **3**, 289 (1993).

[60] Z.-C. Yan, G. W. F. Drake, *Phys. Rev. A*, **52**, 3711 (1995).

[61] M. M. Cassar, *High Precision Theoretical Study of* H_2^+, (University of Windsor, PhD thesis).

[62] V. I. Korobov, *Phys. Rev. A,* **74**, 052506 (2006).

[63] Z.-X. Zhong, *et al, Phys. Rev. A,* **79**, 064502 (2009).

[64] V. I. Korobov, L. Hilico, J.-Ph. Karr, *Phys. Rev. A,* **87**, 062506 (2013).

Appendix A: Reduction of Laplacian Operators ∇_1^2 , ∇_2^2 and $\nabla_1 \cdot \nabla_2$

To determine the explicit form of the operators ∇_1^2 and ∇_2^2, we may notice that

$$\nabla_i^2 = \nabla_i \cdot \nabla_i, \quad i = 1, 2. \tag{A.1}$$

For the purpose of simplicity, due to the symmetry between ∇_1^2 and ∇_2^2, we may concentrate on ∇_1^2 for the time being. The form for ∇_2^2 can easily be obtained by interchange of the indices $1 \leftrightarrow 2$.

Suppose the wave function to be acted on can be expressed in spherical polar coordinates

$$\begin{aligned} \Phi &= \mathcal{R}(r_1, r_2, r_{12})\Omega_{\ell_1,\ell_2} \\ &= \mathcal{R}(r_1, r_2, r_{12})Y_{\ell_1 m_1}(\hat{\mathbf{r}}_1)Y_{\ell_2 m_2}(\hat{\mathbf{r}}_2), \end{aligned} \tag{A.2}$$

by which we mean the function can be separated into a product of a radial part and an angular part. Accordingly, the operator can be decomposed in a similar way

$$\begin{aligned} \nabla_1\Phi &= (\nabla_1\mathcal{R})\Omega + \mathcal{R}(\nabla_1\Omega) \\ &= \left\{(\nabla_1 r_1)\frac{\partial\mathcal{R}}{\partial r_1} + (\nabla_1 r_{12})\frac{\partial\mathcal{R}}{\partial r_{12}}\right\}\Omega \\ &\quad + \mathcal{R}(\nabla_1^Y\Omega), \end{aligned} \tag{A.3}$$

where ∇_1^Y acts only on the angular part, and can be rewritten as

$$\nabla_1^Y = \frac{1}{r_1}\hat{e}_{\theta_1}\frac{\partial}{\partial\theta_1} + \frac{1}{r_1\sin\theta_1}\hat{e}_{\phi_1}\frac{\partial}{\partial\phi_1}. \tag{A.4}$$

To proceed with the deduction, we need the formulae

$$\nabla_1 r_1 = \frac{\mathbf{r}_1}{r_1} \quad \text{and} \quad \nabla_1 r_{12} = \frac{\mathbf{r}_{12}}{r_{12}}. \tag{A.5}$$

It is convenient to verify these two relationships in Cartesian coordinates

$$\nabla_1 r_1 = \left(\frac{\partial}{\partial x_1}\hat{\imath} + \frac{\partial}{\partial y_1}\hat{\jmath} + \frac{\partial}{\partial z_1}\hat{k}\right)\sqrt{x_1^2 + y_1^2 + z_1^2}$$

$$= \frac{1}{\sqrt{x_1^2 + y_1^2 + z_1^2}}\left(x_1\hat{\imath} + y_1\hat{\jmath} + z_1\hat{k}\right) = \frac{\mathbf{r}_1}{r_1}.$$

Thus, the gradient operator ∇_1 can be cast into

$$\nabla_1 = \frac{\mathbf{r}_1}{r_1}\frac{\partial}{\partial r_1} + \frac{\mathbf{r}_{12}}{r_{12}}\frac{\partial}{\partial r_{12}} + \nabla_1^Y. \tag{A.6}$$

Before we begin to expand ∇_1^2 according to the idea of (A.1), some more relationships need to be introduced here. They are

$$\hat{r}_1 \cdot \frac{\partial \hat{r}_1}{\partial r_1} = \frac{2}{r_1}, \tag{A.7}$$

$$\hat{r}_1 \cdot \frac{\partial}{\partial r_1}\frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} = \frac{3}{r_{12}}, \tag{A.8}$$

$$\mathbf{r}_1 \cdot \nabla_1^Y = 0, \tag{A.9}$$

$$(\nabla_1^Y)^2 = -\frac{\mathbf{l}_1^2}{r_1^2}. \tag{A.10}$$

(A.7) and (A.8) can be verified in Cartesian coordinates much the same way we derived (A.5). (A.9) is understandable if we know the orthogonality relationship between the unit vectors, $\hat{e}_{r_1} \cdot \hat{e}_{\theta_1} = \hat{e}_{r_1} \cdot \hat{e}_{\phi_1} = 0$. We can achieve (A. 10) by performing (A.4) on an object twice and by comparing with the square of the angular momentum vector in spherical polar coordinates:

$$\mathbf{l}_1^2 = -\left\{\frac{1}{\sin\theta_1}\frac{\partial}{\partial\theta_1}\left(\sin\theta_1\frac{\partial}{\partial\theta_1}\right) + \frac{1}{\sin^2\theta_1}\frac{\partial^2}{\partial\phi_1^2}\right\}. \tag{A.11}$$

Now, (A.1) can be expressed like

$$\nabla_1^2 = \nabla_1 \cdot \nabla_1$$

$$= \left\{\hat{r}_1 \frac{\partial}{\partial r_1} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\} \cdot \left\{\hat{r}_1 \frac{\partial}{\partial r_1} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\}$$

$$= \hat{r}_1 \cdot \frac{\partial}{\partial r_1}\left\{\hat{r}_1 \frac{\partial}{\partial r_1} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \frac{\partial}{\partial r_{12}}\left\{\hat{r}_1 \frac{\partial}{\partial r_1} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\}$$

$$+ \nabla_1^Y \cdot \left\{\hat{r}_1 \frac{\partial}{\partial r_1} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\}$$

$$= \hat{r}_1 \cdot \frac{\partial}{\partial r_1}\left\{\hat{r}_1 \frac{\partial}{\partial r_1}\right\} + \hat{r}_1 \cdot \frac{\partial}{\partial r_1}\left\{\frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}}\right\} + \hat{r}_1 \cdot \frac{\partial}{\partial r_1}\{\nabla_1^Y\} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \frac{\partial}{\partial r_{12}}\left\{\hat{r}_1 \frac{\partial}{\partial r_1}\right\}$$

$$+ \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \frac{\partial}{\partial r_{12}}\left\{\frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \frac{\partial}{\partial r_{12}}\right\} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \frac{\partial}{\partial r_{12}}\{\nabla_1^Y\} + \nabla_1^Y \cdot \left\{\hat{r}_1 \frac{\partial}{\partial r_1}\right\} + \nabla_1^Y$$

$$\cdot \left\{\frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial}{\partial r_{12}}\right\} + \nabla_1^Y \cdot \nabla_1^Y$$

$$= \hat{r}_1 \cdot \left\{\left(\frac{\partial \hat{r}_1}{\partial r_1}\right) \frac{\partial}{\partial r_1} + \hat{r}_1 \frac{\partial^2}{\partial r_1^2}\right\} + \hat{r}_1 \cdot \left\{\left[\frac{\partial}{\partial r_1} \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}}\right] \frac{\partial}{\partial r_{12}} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial^2}{\partial r_1 \partial r_{12}}\right\} + \hat{r}_1$$

$$\cdot \nabla_1^Y \frac{\partial}{\partial r_1} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \left\{\hat{r}_1 \frac{\partial^2}{\partial r_1 \partial r_{12}}\right\} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}}$$

$$\cdot \left\{(\mathbf{r}_1 - \mathbf{r}_2)\left(\frac{\partial}{\partial r_{12}} \frac{1}{r_{12}}\right) \frac{\partial}{\partial r_{12}} + \frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \frac{\partial^2}{\partial r_{12}^2}\right\} + 2\frac{\mathbf{r}_1 - \mathbf{r}_2}{r_{12}} \cdot \left\{\nabla_1^Y \frac{\partial}{\partial r_{12}}\right\}$$

$$+ (\nabla_1^Y \cdot \hat{r}_1) \frac{\partial}{\partial r_1} + \nabla_1^Y \cdot \nabla_1^Y.$$

Taking into account (A.7)-(A.10) as well as the triangule relation $r_{12}^2 = r_1^2 + r_2^2 - 2r_1 r_2 \cos\theta$, a further simplification of the above expression will eventually lead to

$$\nabla_1^2 = \frac{1}{r_1^2}\frac{\partial}{\partial r_1}\left(r_1^2\frac{\partial}{\partial r_1}\right) + \frac{1}{r_{12}^2}\frac{\partial}{\partial r_{12}}\left(r_{12}^2\frac{\partial}{\partial r_{12}}\right)$$

$$+ 2(r_1 - r_2\cos\theta)\frac{1}{r_{12}}\frac{\partial^2}{\partial r_1 \partial r_{12}} \qquad \text{(A.12)}$$

$$- \frac{2(\mathbf{r}_2 \cdot \nabla_1^Y)}{r_{12}}\frac{\partial}{\partial r_{12}} + \frac{\ell_1(\ell_1+1)}{r_1^2}.$$

Reduction of the mass polarization operator $\nabla_1 \cdot \nabla_2$ is similar:

$$\nabla_1 \cdot \nabla_2 = \left\{\frac{\mathbf{r}_1}{r_1}\frac{\partial}{\partial r_1} + \frac{\mathbf{r}_{12}}{r_{12}}\frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\} \cdot \left\{\frac{\mathbf{r}_1}{r_1}\frac{\partial}{\partial r_1} + \frac{\mathbf{r}_{12}}{r_{12}}\frac{\partial}{\partial r_{12}} + \nabla_1^Y\right\}$$

$$= \frac{\mathbf{r}_1 \cdot \mathbf{r}_2}{r_1 r_2}\frac{\partial}{\partial r_1 \partial r_2} - \frac{\mathbf{r}_1 \cdot \mathbf{r}_{12}}{r_1 r_{12}}\frac{\partial}{\partial r_1 \partial r_{12}} + \frac{\mathbf{r}_1 \cdot \nabla_2^Y}{r_1}\frac{\partial}{\partial r_1} + \frac{\mathbf{r}_2 \cdot \mathbf{r}_{12}}{r_2 r_{12}}\frac{\partial^2}{\partial r_2 \partial r_{12}} - \frac{\partial^2}{\partial r_{12}^2} - \frac{\mathbf{r}_{12}}{r_{12}}$$

$$\cdot\left(\frac{\partial}{\partial r_{12}}\frac{\mathbf{r}_{12}}{r_{12}}\right)\frac{\partial}{\partial r_{12}} + \frac{\mathbf{r}_{12} \cdot \nabla_2^Y}{r_{12}}\frac{\partial}{\partial r_{12}} + \frac{\mathbf{r}_2 \cdot \nabla_1^Y}{r_2}\frac{\partial}{\partial r_2} - \frac{\mathbf{r}_{12} \cdot \nabla_1^Y}{r_{12}}\frac{\partial}{\partial r_{12}} + \nabla_1^Y \cdot \nabla_2^Y.$$

Then using the formula

$$\mathbf{r}_1 \cdot \mathbf{r}_2 = \frac{r_1^2 + r_2^2 - r_{12}^2}{2},$$

and

$$\mathbf{r}_1 \cdot \left(\nabla_1^Y Y_{\ell_1 m_1}(\hat{\mathbf{r}}_1)\right) = 0,$$

together with the definition $\hat{\nabla}_i^Y = r_i \nabla_i^Y$, we can get

$$\nabla_1 \cdot \nabla_2 = -\left(\frac{\partial^2}{\partial r_{12}^2} + \frac{2}{r_{12}}\frac{\partial}{\partial r_{12}}\right) + \frac{r_1^2 + r_2^2 - r_{12}^2}{2r_1 r_2}\frac{\partial^2}{\partial r_1 \partial r_2} + \frac{r_2^2 - r_1^2 - r_{12}^2}{2r_1 r_{12}}\frac{\partial^2}{\partial r_1 \partial r_{12}}$$

$$+ \frac{r_2^2 - r_1^2 - r_{12}^2}{2r_2 r_{12}}\frac{\partial^2}{\partial r_2 \partial r_{12}} + \frac{r_1}{r_2}\left(\frac{1}{r_1}\frac{\partial}{\partial r_1} + \frac{1}{r_{12}}\frac{\partial}{\partial r_{12}}\right)\left(\hat{r}_1 \cdot \hat{\nabla}_2^Y\right) + \frac{1}{r_1 r_2}\left(\hat{\nabla}_1^Y \cdot \hat{\nabla}_2^Y\right)$$

$$+ \frac{r_2}{r_1}\left(\frac{1}{r_2}\frac{\partial}{\partial r_2} + \frac{1}{r_{12}}\frac{\partial}{\partial r_{12}}\right)\left(\hat{r}_2 \cdot \hat{\nabla}_1^Y\right). \qquad \text{(A.13)}$$

Appendix B: Parallel Cholesky Decomposition Code

The following code is written in FORTRAN 90, for the purpose of decomposing a real symmetric matrix. For the convenience of programming, we control the dimension of the matrix, *nr*, by controlling the number of base or the number of CPUs so that *nr* can be divided into the number of CPUs.

```
01  Program  Cholesky-Decomposition
02  implicit  none
03  Include  'mpif.h'
04  Integer   i, j, k, l, q,t
05  Integer, parameter:: nr=10000,pd=25, md=nr/pd
06  Integer:: isg(nr), isgm(nr), isgg(pd-1), isgf(pd-1)
07  Integer  myid, master, ierr, numprocs, status(MPI_STATUS_SIZE)
08  Real(kind=16):: sum, xp(nr), a(nr,nr)
09  Real(kind=16):: am(md, nr), G(pd-1,nr), F(pd-1,nr)
10  !nr is the dimension of the matrix, pd is the number of CPUs
11  !Initialize MPI
12  Call MPI_INIT(ierr)
13  Call MPI_COMM_RANK(MPI_COMM_WORLD,myid,ierr)
14  Call MPI_COMM_SIZE(MPI_COMM_WORLD,numprocs,ierr)
15  Master=0
16  !Input the matrix to be decomposed, only the lower-triangle part is needed
17  If(myid==master)then
18  Open(unit=40, file='a.txt')
19  Do i=1,nr
20  Do j=i,nr
21  Read(40,*) a(j,i)
22  End do
```

```
23  End do
24  Close(40)
25  End if
26  ! Allocate the matrix onto those CPUs in the manner of WRAP
27  Do i=1,pd-1
28  If(myid==master) then
29  Do j=1, md
30  Am(j,1:nr)=a(1+i+pd*(j-1),1:nr)
31  End do
32  Call MPI_SEND(am(1,1),md*nr,mpi_real16,i,i,mpi_comm_world,ierr)
33  Endif
34  If(myid==i) then
35  Call MPI_RECV(am(1,1),md*nr,mpi_real16,master,i, mpi_comm_world,  &
    status,ierr)
36  End if
37  End do
38  If(myid==master)then
39  Do i=1,md
40  Am(i,1:nr)=a(1+pd*(i-1),1:nr)
41  End do
42  End if
43  ! Allocation completed
44  ! Start the Cholesky decomposition of each rank
45  Do i=0,md-1
46  K=i*pd+myid
47  L=k-pd+1
48  Isgm(k+1)=1
49  If(myid/=master) then
```

```
50  Call MPI_RECV(G(1,1),(pd-1)*nr,MPI_REAL16,myid-1,                &
    i,MPI_COMM_WORLD,status, ierr)
51  Call MPI_RECV(isgg(1),(pd-1),MPI_INTEGER2,myid-1,                &
    md+i,MPI_COMM_WORLD, STATUS,ierr)
52  Endif
53  Do j=0,pd-2
54  Sum=0.Q0
55  Do q=0,j+l-1
56  If(q<=l-1)then
57  Sum=sum+am(i+1,q+1)*G(j+1,q+1)*isgm(q+1)
58  Else
59  Sum=sum+am(i+1,q+1)*G(j+1,q+1)*isgg(q-l+1)
60  End if
61  Enddo
62  Am(i+1,j+l+1)=(am(i+1,j+l+1)-sum)/G(j+1,j+l+1)
63  F(j+1, 1:nr)=G(j+2, 1:nr)
64  Isgm(j+l+1)=isgg(j+1)
65  Isgf(j+1)=isgg(j+2)
66  Enddo
67  Sum=0.Q0
68  Do q=0,k-1
69  If(q<=l-1)then
70  Sum=sum+am(i+1,q+1)*am(i+1,q+1)*isgm(q+1)
71  Else
72  Sum=sum+am(i+1,q+1)*am(i+1,q+1)*isgg(q-l+1)
73  End if
74  End do
75  Sum=am(i+1, k+1)-sum
```

```
76  If(sum>0.Q0) then
77  Isgm(k+1)=-1
78  Am(i+1,k+1)=sqrt(-sum)
79  Else
80  Am(i+1,k+1)=sqrt(sum)
81  End if
82  F(pd-1, 1:nr)=am(i+1,1:nr)
83  Isgf(pd-1)=isgm(k+1)
84  If(myid/=pd-1)then
85  Call  MPI_SEND(F(1,1),(pd-1)*nr,                                    &
    MPI_REAL16,master,i,MPI_COMM_WORLD, IERR)
86  End if
87  If(myid==pd-1)then
88  Call  MPI_SEND(F(1,1),(pd-1)*nr,                                    &
    MPI_REAL16,master,i,MPI_COMM_WORLD, IERR)
89  Call  MPI_SEND(isgf(1),(pd-1),                                      &
    MP_INTEGER2,master,md+i,MPI_COMM_WORLD,IERR)
90  End if
91  Do t=i+1,md-1
92  Do j=0,pd-2
93  Sum=0.Q0
94  Do q=0,j+l-1
95  If(q<=l-1)then
96  Sum=sum+am(t+1,q+1)*G(j+1,q+1)*isgm(q+1)
97  Else
98  Sum=sum+am(t+1,q+1)*G(j+1,q+1)*isgg(q-l+1)
99  End if
100 End do
```

```
101 Am(t+1,j+l+1)=(am(t+1,j+l+1)-sum)/G(j+1,j+l+1)
102 End do
103 Sum=0.Q0
104 Do q=0,k-1
105 If(q<=l-1)then
106 Sum=sum+am(t+1,q+1)*am(i+1,q+1)*isgm(q+1)
107 Else
108 Sum=sum+am(t+1,q+1)*am(i+1,q+1)*isgg(q-l+1)
109 End if
110 End do
111 Am(t+1,k+1)=(am(t+1,k+1)-sum)/am(i+1,k+1)
112 End do
113 If(myid==master) then
114 Call MPI_RECV(G(1,1),(pd-1)*nr,MPI_REAL16,pd-1,i,                &
    MPI_COMM_WORLD,status,ierr)
115 Call MPI_RECV(isgg(1),(pd-1),MPI_INTEGER2,pd-1,                 &
    md+i,MPI_COMM_WORLD,STATUS,IERR)
116 End if
117 End do
118 ! End of decomposition
119 ! Collecting data from each rank
120 If(myid==master) then
121 Do i=1,md
122 A(1+pd*(i-1),1:nr)=am(i,1:nr)
123 End do
124 End if
125 Do i=1,pd-1
126 If(myid==i) then
```

```
127 Call    MP-_SEND(am(1,1),                                    &
    md*nr,mpi_real16,master,mpi_comm_world, ierr)
128 End if
129 If(myid==master)then
130 Call  MPI_RECV(am(1,1),md*nr,mpi_real16,i,i,                  &
    mpi_comm_world,status,ierr)
131 Do j=1,md
132 A(1+i+pd*(j-1),1:nr)=am(j,1:nr)
133 End do
134 End if
135 End do
136 If(myid==master) then
137 Call  MPI_RECV(isgm(nr-pd+2),pd-1,mpi_integer,pd-1,1,          &
    mpi_comm_world,status,ierr)
138 Else if(myid==pd-1)then
139 Call  MPI_SEND(isgm(nr-pd+2),pd+1,                            &
    mpi_integer,master,1,mpi_comm_world,ierr)
140 End if
141 ! Complete collection
142 ! Finish decomposition, quit parallel mode
143 Call MPI_FINALIZE(IERR)
144 Stop
145 End
```

Curriculum Vitae

Candidate's full name: Ye Ning

Universities attended (with dates and degrees obtained):

Institution of Applied Physics and Computational Mathematics, Beijing, China,

September 1st, 2002—July 1st, 2005. M.Sc.

Publications:

1. "Benchmark Variational Calculation of Hydrogen Molecular Ion"
   Ye Ning and Z.-C. Yan
   Submitted to Physical Review A

Conference Presentations:

1. Ye Ning and Z.-C. Yan, "High precision variational calculations of non-relativistic energy levels of the hydrogen molecular ion", the DAMOP14 Conference of the American Physical Society, Madison, Wisconsin, USA, June 2-6, 2014.

2. Ye Ning, Z.-C. Yan and Y. K. Ho, "An investigation of resonances in $e^+ - H$ scattering in Debye plasmas", the DAMOP14 Conference of the American Physical Society, Madison, Wisconsin, USA, June 2-6, 2014.

3. Y. K. Ho, Ye Ning and Z.-C. Yan, "Triply-excited auto-dissociating resonant states in the positron-helium system", the DAMOP14 Conference of the American Physical Society, Madison, Wisconsin, USA, June 2-6, 2014.