Computational Chemistry in Parallel

by

Dylan Hennessey

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science.

Department of Chemistry
University of Alberta

Abstract

General Purpose computing with Graphics Processing Units (GPGPU) is a method of computation that allows for calculations to be performed in parallel. Graphics Processing Units (GPU) allow a single basic computer to perform as well as a super computer cluster composed of only Central Processing Units (CPU), at a fraction of the cost. As quantum computational chemistry scales very poorly with the size of systems, the use of GPGPU has been a very effective tool for quantum chemists.

This thesis presents a program called cudaDFRATOM. This program, written in CUDA Fortran, allows for the calculation and optimization of Well-Tempered Basis Sets (WTBS) at the four-component relativistic level of theory. Except for a few small initial calculations and control flow, all work is performed entirely on the GPU.

When calculating the two-electron integrals, cudaDFRATOM uses a very effective method of utilizing as much of the GPU's resources as possible and can calculate these integrals approximately 20 times faster than a similar program running on only a CPU.

Two different algorithms for forming the $\mathbf{P}$ and $\mathbf{Q}$ supermatrices are presented. The relatively small size of these matrices and the need to use double-precision floating-point operations stunts the speedup of these algorithms such that they only attain a speedup of about 2 times over the CPU program. An overall speedup of about 4 times is achieved by cudaDFRATOM.

In addition, new WTBS are calculated for non-relativistic levels of theory using the rWTBS program and at the four-component relativistic level of theory using cudaDFRATOM. In both cases, a novel algorithm for automating the optimization of these basis sets was used. The algorithm finds a basis set that is as small a possible while still retaining sufficient accuracy to numerical calculations.

Preface

This thesis is my own original work. No part of this thesis has been previously published. Experiments and code were planned, performed, and written by myself with the help of Mariusz Klobukowski.

In addition, I have also worked on a project with Mariusz which is not in this thesis. The project investigated how the FHF^- molecule behaves in a confining potential. The results were published in a paper titled "Hydrogen bonding and autoionization in confined bifluoride system FHF^-" in *The Journal of Physics B: Atomic, Molecular and Optical Physics* vol. 50 page 155101 in the year 2017.

For Jessie, who puts up with more than she needs to

Acknowledgements

I would first like to acknowledge my supervisor, Mariusz Klobukowski. Few grad students are lucky enough to have a mentor who is as kind, helpful, and a source of as much interesting conversation as he. I would not have as easily discovered my love of programming if it weren't for him, and for that I am truly grateful. He dealt with my ... sporadic writing schedule with the patience and grace of a saint which made this whole process of writing a thesis that much more bearable. I really cannot say enough how much his guidance has meant to me.

I would like to thank one of my course instructors, Paul Lu. He took a chance with me when no one else would and gave me the opportunity to more fully explore my computational potential. I don't think I could have made it without his help, so again, thank you.

I would also like to thank my friends and family. As much as I love it, the science can't be the reason to wake up *every* morning, so thank you to those who helped keep my spirits up, especially when it seemed like I would be writing this forever.

I should also thank the money. The work in this thesis could not have happened without the funding from NSERC and FGSR, and without the computer resources of WestGrid and Compute Canada.

Finally, I would like to thank my girlfriend Jessie Hartmann. We've been through Hell and back together, and there is no one else I would have rather had at my side than her. For some reason, she continues to love me, even as a make her watch a video on sorting algorithms for the tenth time, so for you Jessie I give my appreciation, affection, and love.

Contents

List of Tables

List of Figures

List of Symbols

Some symbols are reused. The exact meaning will need to obtained from context.

Symbol	Description
$\hat{H}$	Hamiltonian operator
Ψ	Wavefunction
E	Energy
∇	Gradient of a function
$\hat{H}_{el}$	Electronic Hamiltonian operator
a_o	Bohr radius
ψ	Spin orbital
ϕ	Orbital
α	Spin function, four-component matrix, Gaussian nuclei exponent, well-tempered basis set parameter
β	Spin function, four-component matrix, well-tempered basis set parameter
ω	Spin coordinate
ϵ	Vector of spin orbital energies
Φ	Basis function
ζ	Basis function exponent
λ	Orbital symmetry
μ	Orbital symmetry
c	Speed of light
m_0	Mass of object at rest
$\hbar$	Reduced Planck's constant
q	Particle charge
$\hat{A}, \hat{\varphi}$	Electromagnetic field potentials
$\boldsymbol{\psi}$	Large component spinor
$\boldsymbol{\phi}$	Large component spinor
ϵ_0	Permittivity of free space
δ	Well-tempered basis set parameter
γ	Well-tempered basis set parameter

Chapter 1

Introduction

1.1 Solving the Schrödinger Equation

The Schrödinger equation[1] in its time-independent, non-relativistic form is given in Equation 1.1

$$\hat{H}\Psi = E\Psi \tag{1.1}$$

where $\hat{H}$ is the Hamiltonian operator, Ψ is referred to as the wavefunction, and E is the energy of the system described by the wavefunction. For a molecular system, the Hamiltonian operator in atomic units (a.u.) is given by Equation 1.2

$$\hat{H} = -\frac{1}{2}\sum_i^n \nabla_i^2 - \sum_A^N \frac{1}{2M_A}\nabla_A^2 - \sum_i^n\sum_A^N \frac{Z_A}{r_{iA}} + \sum_i^n\sum_{j>i}^N \frac{1}{r_{ij}} + \sum_A^N\sum_{B>A}^N \frac{Z_A Z_B}{r_{AB}} \tag{1.2}$$

where i and j refer to electrons, A and B refer to nuclei, and n and N refer to the maximum number of electrons and nuclei, respectively. M_A and Z_A are the mass and charge of nucleus A. r_{xy} is the distance between particles x and y, and $\nabla_n^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ where x, y, and z are the Cartesian coordinates of particle n. Each term accounts for a different kind of energy: $-\frac{1}{2}\sum_i^n \nabla_i^2$ is the kinetic energy of the electrons, $\sum_A^N \frac{1}{2M_A}\nabla_A^2$ is the kinetic energy of the nuclei, $\sum_i^n\sum_A^N \frac{Z_A}{r_{iA}}$ is the potential energy of attraction between nuclei and electrons, $\sum_i^n\sum_{j>i}^N \frac{1}{r_{ij}}$ is the potential energy of repulsion between electrons, and $\sum_A^N\sum_{B>A}^N \frac{Z_A Z_B}{r_{AB}}$ is the potential energy of repulsion between nuclei. Together, this accounts for all of the energy of a molecular system. As this operator is quite complex, it would be helpful to simplify it if possible. Fortunately, Born and Oppenheimer[2] realized that an approximation could be made that would substantially reduce the amount of work we have to do.

Much like a satellite in orbit around the Earth, the proton and neutron vastly outweigh the electron. If we look at the second term in Equation 1.2, we notice that it is inversely

proportional to the mass of the nuclei. We can therefore assume, to a decent approximation, that the kinetic energy of the nuclei is very close to zero compared to the other terms. This allows us to treat them as stationary classical particles. As a bonus, this turns the potential energy of repulsion between nuclei into a constant. The killing of these two terms reduces the Hamiltonian operator to the *electronic* Hamiltonian given in Equation 1.3

$$\hat{H}_{el} = -\frac{1}{2} \sum_i^n \nabla_i^2 - \sum_i^n \sum_A^N \frac{Z_A}{r_{iA}} + \sum_i^n \sum_{j>i}^N \frac{1}{r_{ij}} + \sum_{B>A}^N \frac{Z_A Z_B}{r_{AB}} \tag{1.3}$$

$\hat{H}$ can be assumed to be $\hat{H}_{el}$ for the remainder of this Thesis, and with the Hamiltonian out of the way, we can now turn our attention to the wavefunction. While we were able to get away with turning nuclei into classical particles, the electrons unfortunately remain fully quantum. Therefore, we need to deal with all the complicated mathematical baggage this comes with.

Consider the hydrogen atom which, like all other one-electron atomic systems, is one of the few exactly solvable problems in quantum mechanics. The solution for its wavefunction in spherical coordinates is given in Equation 1.4[3]

$$\Psi_{nlm}(r, \theta, \varphi) = R_{nl}(r) Y_l^m(\theta, \varphi) \tag{1.4}$$

where R_{nl} is called the radial function and $Y_l^m(\theta, \varphi)$ is called the angular function. The radial equation has the form

$$R_{nl}(r) = \frac{1}{r} \left(\frac{r}{a_o n}\right)^{l+1} e^{-\frac{r}{a_o n}} L_{n-l-1}^{2l+1}\left(\frac{r}{a_o n}\right) \tag{1.5}$$

where a_o is the Bohr radius and $L_{n-l-1}^{2l+1}\left(\frac{r}{a_o n}\right)$ is polynomial of $\left(\frac{r}{a_o n}\right)$ and order $j_{\max} = n - l - 1$ of $\left(\frac{r}{a_o n}\right)$. Its coefficients are given by

$$c_{j+1} = \frac{2(j + l + 1 - n)}{(j + 1)(j + 2l + 2)} c_j \tag{1.6}$$

where c_0 is a constant determined by normalization. The polynomial itself given by

$$L_{q-p}^p(x) = (-1)^p \left(\frac{d}{dx}\right)^q L_q(x) \tag{1.7}$$

$$L_q(x) = e^x \left(\frac{d}{dx}\right)^q \left[x^q e^{-x}\right] \tag{1.8}$$

The angular function has the form

$$Y_l^m(\theta, \varphi) = \epsilon \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos\theta) e^{im\varphi} \tag{1.9}$$

where $i = \sqrt{-1}$ and $\epsilon = (-1)^m$ for $m \geq 0$ or 1 if $m < 0$. $P_l^m(\cos\theta)$ is a polynomial of $\cos\theta$ and has the form of

$$P_l^m(x) = (1-x^2)^{\frac{|m|}{2}} \frac{d^{|m|}}{dx^{|m|}} P_l(x) \tag{1.10}$$

$$P_l(x) = \frac{1}{2^l l!} \frac{d^j}{dx^j}(x^2-1)^l \tag{1.11}$$

Inspection of these equations reveals that n must be a positive integer greater than 0, l must be a positive integer between 0 and $n-1$, and $|m|$ can be no larger than l. These are called quantum numbers, specifically n is the principal quantum number, l is the azimuthal quantum number, and m is the magnetic quantum number. Notice that there is no limit to how large n can get, which suggests that there is not just one solution to this equation, but an infinite number of them. The solutions to the hydrogen-like atom wavefunctions are given the special name of orbitals. Also, orbitals with the same value of l are refered to be of the same symmetry. Table 1.1 shows the reduced form of Ψ for a few different orbitals. For atoms with more than one electron, if we make a further assumption that electrons can be modelled like ideal springs, it has been shown that an atomic system with two electrons can also be solved[4], but we can go no further than this. At least, not exactly. In the next section we will see how solutions for atoms, and indeed molecules, can begin to be approached.

Table 1.1: *The first few exact solutions for the hydrogen atom wavefunction.*

n	l		
	0	1	
	$m=0$	$m=0$	$m=\pm 1$
1	$\left(\dfrac{1}{\pi a_o^3}\right)^{\frac{1}{2}} e^{\left(-\frac{r}{a_o}\right)}$	—	—
2	$\dfrac{1}{\sqrt{8\pi a_o^3}}\left(1-\dfrac{r}{2a_o}\right)e^{\left(-\frac{r}{2a_o}\right)}$	$\dfrac{1}{\sqrt{32\pi a_o^3}}\dfrac{r}{a_o}e^{\left(-\frac{r}{2a_o}\right)}\cos\theta$	$\mp\dfrac{1}{\sqrt{64\pi a_o^3}}\dfrac{r}{a_o}e^{\left(-\frac{r}{2a_o}\right)}\sin\theta e^{\pm i\varphi}$

1.2 Hartree-Fock Method

While the wavefunction might appear complex, it is important to remember that it is still just a wave, and one of the fundamental properties of waves is that they can be added together to produce a separate, more complex wave. Likewise, any wave, no matter how complex, can be broken down as a summation of many simpler waves[5]. Thus, any linear combination of solutions to the wavefunction must therefore be a solution of a different (albeit perhaps non-physical) wavefunction. It is this property that will be exploited to solve our many electron problem. But first we must define some ground rules.

The first rule is that the overall wavefunction must describe a system of fermions. This means that the wavefunction must be antisymmetric with respect to the exchanging of electronic coordinates, i.e. $\Psi(1, 2, \ldots, n, \ldots, m) = -\Psi(1, 2, \ldots, m, \ldots, n)$ *and* the probability density of the wavefunction must be indistinguishable with respect to the exchanging of electronic coordinates, i.e. $|\Psi(1, 2, \ldots, n, \ldots, m)|^2 = |\Psi(1, 2, \ldots, m, \ldots, n)|^2$. The easiest way to meet this requirement is by using a Slater determinant[6], and by introducing the concept of spin. Spin is a fourth quantum number, which we have neglected to talk about so far. Spin is a confusing property that some quantum particles have. No one really knows what it is, other than to say it has something to do with the angular momentum of the particle, and oddly enough that it in no way refers to the direction that the particle is spinning. It has been shown to have binary values[7], which will be refered to as up spin ($\uparrow$) and down spin ($\downarrow$). A Slater determinant that obeys antisymmetry is shown in Equation 1.12

$$\Psi(1, 2, 3, \ldots, n) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \psi_1(1) & \psi_2(1) & \psi_3(1) & \ldots & \psi_n(1) \\ \psi_1(2) & \psi_2(2) & \psi_3(2) & \ldots & \psi_n(2) \\ \psi_1(3) & \psi_2(3) & \psi_3(3) & \ldots & \psi_n(3) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \psi_1(n) & \psi_2(n) & \psi_3(n) & \ldots & \psi_n(n) \end{vmatrix} \qquad (1.12)$$

Here, n refers to the number of electrons in the system, and the numbers in parentheses refer to specific electrons. ψ_i refers to the ith *spin* orbital which is equal to $\phi_{\frac{i+1}{2}}(r, \theta, \varphi)\alpha(\omega)$ if i is odd, or $\phi_{\frac{i}{2}}(r, \theta, \varphi)\beta(\omega)$ if i is even (ϕ is an orbital and ω is the spin coordinate). α and β are called spin functions. What they actually are doesn't matter, but they do have to be orthonormal to each other.

$$\begin{aligned} \langle\alpha(\omega_i)|\alpha(\omega_i)\rangle = \langle\beta(\omega_i)|\beta(\omega_i)\rangle = 1 \\ \langle\alpha(\omega_i)|\beta(\omega_i)\rangle = \langle\beta(\omega_i)|\alpha(\omega_i)\rangle = 0 \end{aligned} \qquad (1.13)$$

Using Slater determinants has some interesting consequences. If we had a system where two electrons of the same spin were in the same spin orbital, that would mean that two of the columns would be equal to one another. Such a determinant would be equal to zero,

which means that the wavefunction for such a system would not exist. This is the famous Pauli antisymmetry principle[8]. The math showing that is satisfies antisymmetry is shown below[9].

$$\Psi(1,2) = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_1(1)\alpha(\omega_1) & \phi_1(1)\beta(\omega_1) \\ \phi_1(2)\alpha(\omega_2) & \phi_1(2)\beta(\omega_2) \end{vmatrix} \tag{1.14}$$

$$\Psi(1,2) = \frac{1}{\sqrt{2}} [\phi_1(1)\alpha(\omega_1)\phi_1(2)\beta(\omega_2) - \phi_1(1)\beta(\omega_1)\phi_1(2)\alpha(\omega_2)] \tag{1.15}$$

$$-\Psi(1,2) = \frac{1}{\sqrt{2}} [\phi_1(1)\beta(\omega_1)\phi_1(2)\alpha(\omega_2) - \phi_1(1)\alpha(\omega_1)\phi_1(2)\beta(\omega_2)] \tag{1.16}$$

$$-\Psi(1,2) = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_1(2)\alpha(\omega_2) & \phi_1(2)\beta(\omega_2) \\ \phi_1(1)\alpha(\omega_1) & \phi_1(1)\beta(\omega_1) \end{vmatrix} = \Psi(2,1) \tag{1.17}$$

As can be seen, changing the indexing of the electrons is equivalent to swapping the rows for those electrons, which changes the sign of the determinant. It is important to note that because observable quantities depend not the wavefunction, but the *absolute square* of the wavefunction, altering the indexing of the electrons does not change the calculation of these observables. Consider the example of the electron density probability distribution shown below (note that ϕ, α, and β are real in this example, but even if they weren't the results would not change).

$$|\Psi(1,2)|^2 = \frac{1}{2} \left[\begin{vmatrix} \phi_1(1)\alpha(\omega_1) & \phi_1(1)\beta(\omega_1) \\ \phi_1(2)\alpha(\omega_2) & \phi_1(2)\beta(\omega_2) \end{vmatrix}^2 \right] \tag{1.18}$$

$$|\Psi(1,2)|^2 = \frac{1}{2} [(\phi_1(1)\alpha(\omega_1)\phi_1(2)\beta(\omega_2) - \phi_1(1)\beta(\omega_1)\phi_1(2)\alpha(\omega_2)) \\ \times (\phi_1(1)\alpha(\omega_1)\phi_1(2)\beta(\omega_2) - \phi_1(1)\beta(\omega_1)\phi_1(2)\alpha(\omega_2))] \tag{1.19}$$

$$|\Psi(1,2)|^2 = \frac{1}{2} \int \int \int \int [\phi_1(1)^2\alpha(\omega_1)^2\phi_1(2)^2\beta(\omega_2)^2 + \phi_1(1)^2\beta(\omega_1)^2\phi_1(2)^2\alpha(\omega_2)^2 \\ - 2\phi_1(1)^2\alpha(\omega_1)\beta(\omega_1)\phi_1(2)^2\beta(\omega_2)\alpha(\omega_2)]d1d2d\omega_1d\omega_2 \tag{1.20}$$

$$|\Psi(1,2)|^2 = \frac{1}{2} \left[|\phi_1(1)|^2|\phi_1(2)|^2 + |\phi_1(1)|^2|\phi_1(2)|^2 \right] \tag{1.21}$$

1.2.1 Two-Electron Integrals

As stated before, the main difficulty of performing calculations on many electron systems is figuring out how to deal with the interactions between the electrons. We can get around this

by making the assumption that these interactions can be approximated by figuring out how a single electron interacts with the *average* field generated by all the other electrons. This is called the self-consistent-field (SCF) method[10]. With this, we can replace the Hamiltonian operator with the one-electron Fock operator which is given below[9]

$$\hat{f}(i) = -\frac{1}{2}\nabla_i^2 - \sum_A \frac{Z_A}{r_{iA}} + v^{HF}(i) \tag{1.22}$$

The first two terms are combined into an operator called the H^{core} operator ($\hat{h}(i)$) and $v^{HF}(i)$ is the potential generated by all the electrons except the ith one. $v^{HF}(i)$ is composed of two different operators: the Coulomb operator and the exchange operator. The Coulomb operator is intuitive, it is simply the interaction electron 1 experiences when in a Coulomb potential generated by electron 2

$$\hat{J}_b(1)\psi_a(1) = \left[\int \psi_b^*(2)\frac{1}{r_{12}}\psi_b(2)d\mathbf{x}_2\right]\psi_a(1) \tag{1.23}$$

where $J_b(1)$ is the Coulomb operator for electron 1 when interacting with an electron in spin orbital b and $\mathbf{x}_i$ is the spatial coordinates of electron i. The exchange operator is slightly more complicated than the Coulomb operator. While the Coulomb operator can be defined without a function to operate on (simply remove $\psi_a(1)$ from both sides of Equation 1.23 and it is still a valid operator), the exchange operator only is defined when operating on a function. Its form is given below.

$$\hat{K}_b(1)\psi_a(1) = \left[\int \psi_b^*(2)\frac{1}{r_{12}}\psi_a(2)d\mathbf{x}_2\right]\psi_b(1) \tag{1.24}$$

where $K_b(1)$ is the exchange operator for electron 1 operating on spin orbital a. At first glance, it might appear to be almost the same as Equation 1.23, but notice that ψ_a has been swapped (or exchanged) with ψ_b *inside* the integral. The term itself does not really have a classical interpretation, but its need arises due to the use of an antisymmetric wavefunction.

Now it would be helpful to introduce some new notation. The expectation values for these operators are defined as follows

$$\left\langle \psi_a(1)|\hat{J}_b(1)|\psi_a(1)\right\rangle = \int\int \psi_a^*(1)\psi_a(1)\frac{1}{r_{12}}\psi_b^*(2)\psi_b(2)d\mathbf{x}_1 d\mathbf{x}_2 = (aa|bb) \tag{1.25}$$

$$\left\langle \psi_a(1)|\hat{K}_b(1)|\psi_a(1)\right\rangle = \int\int \psi_a^*(1)\psi_b(1)\frac{1}{r_{12}}\psi_b^*(2)\psi_a(2)d\mathbf{x}_1 d\mathbf{x}_2 = (ab|ba) \tag{1.26}$$

These are called the two-electron integrals. The Hartree-Fock equation can be written as

$$\left[\hat{h}(1) + \sum_{b \neq a}^{n} \left(J_b(1) - K_b(1) \right) \right] \psi_a(1) = \epsilon_a \psi_a(1) \tag{1.27}$$

$$\left\langle \psi_a(1) | \hat{h}(1) | \psi_a(1) \right\rangle + \sum_{b \neq a}^{n} [(aa|bb) - (ab|ba)] = \epsilon_a \tag{1.28}$$

where n is the number of spin orbitals in the system and ϵ_a is the energy of an electron in spin orbital a. If instead we are interested in the energy of an entire atom or molecule, the equation becomes

$$\sum_{a}^{n} \left\langle \psi_a(1) | \hat{h}(1) | \psi_a(1) \right\rangle + \frac{1}{2} \sum_{a}^{n} \sum_{b}^{n} [(aa|bb) - (ab|ba)] = \epsilon \tag{1.29}$$

where the factor of one half is multiplied to the two electron terms as a result of the interactions being double counted by the summations.

1.2.2 Basis Sets

Back in section 1.2, it was said that there were some ground rules for our wavefunction. The second of these ground rules is that our trial wavefunction (Φ) must meet the boundary conditions for the original problem. Up until now, we have used only spin orbitals (ψ) to describe the wavefunction, which as you'll recall is composed of a spin function, which we have already talked about at length, and a spatial orbital (ϕ) which we have almost completely ignored. I'll discuss the spatial orbitals next.

As mentioned, the overall wavefunction of a system (Ψ) cannot be known exactly unless we are looking only at hydrogen-like atoms. So instead we write the wavefunction as a Slater determinant composed of spin orbitals. What exactly is the mathematical form of these spin orbitals? As it turns out, unless we are talking only about atoms, we have no idea what these are either, even from a numerical stand point. However, we can make a fairly educated guess about what they *should* resemble. The general idea is this: by pre-calculating a set of functions with adjustable parameters (ζ), we can represent a wavefunction with a linear combination of $\Phi(\zeta)$ that adequately reproduces results of numerical calculations of ϕ. Mathematically, this is expressed as[9]

$$\phi_u \approx \sum_{i=1}^{K} C_{u,i} \Phi(\zeta_{u,i}) \tag{1.30}$$

where $\zeta_{u,i}$ is the ith adjustable parameters of the basis function Φ which is approximating the uth orbital, $C_{u,i}$ is the ith coefficient of ith basis function of the uth ϕ, and K is the total

number of basis functions that make up the basis set. $C_{u,i}^2$ can be thought of as the "weight" of ϕ that $\Phi(\zeta_{u,i})$ makes up.

When choosing the Φs, our ground rule comes into play. We must choose a function that meets the boundary conditions of the problem, which means that is must go to zero at infinity. Because an atom (or molecule) consists of electrons in the potential generated by one (or many) nuclei, it is reasonable to conclude that the wavefunction would be similar to that of the hydrogen atom. We might initially think to use the exact hydrogen atom solution, but it quickly requires lots of floating point operations to compute which makes it impractical to use for large scale calculations. A more reasonable function is the Slater-type orbital (STO)[11]. The radial part of its 1s orbital centered at R_A is shown below

$$\Phi_{1s}^{STO}(r - R_A) = \sqrt{\frac{\zeta^3}{\pi}} e^{-\zeta|r-R_A|} \tag{1.31}$$

This is more well-behaved than the exact hydrogen atom solution, but problems arise when we try to use them in the two-electron integrals. Because we may be trying to do a two-electron integral over functions which could have up to four different centers, we end up with something that is very hard to compute. Therefore, Gaussian-type orbitals (GTOs) have become the preferred functions for producing basis sets. The radial part of a 1s GTO centered at R_A is shown below[12]

$$\Phi_{1s}^{GTO}(r - R_A) = \left(\frac{2\zeta}{\pi}\right)^{\frac{3}{4}} e^{-\zeta|r-R_A|^2} \tag{1.32}$$

Because the product of two different Gaussians centered at two different locations is a third Gaussian centered at a third location, this makes the two-electron integrals much easier to deal with. The trade off is that GTOs do not as accurately match the hydrogen atom solution as STOs, but this can be fixed somewhat by just using more of them.

All of this reduces our wavefunction issue to a simple optimization problem, which will be discussed more in-depth in Chapter 3.

1.2.3   Hartree-Fock SCF

In the previous sections, we discussed the calculation of the two-electron integrals and introduced the concept of using basis sets to approximate a wavefunction. Now we will discuss how they are actually used.

SCF begins with the formation of the density matrices. If we are looking at a single atom, we can exploit the symmetry of the problem by having a smaller density matrix for each orbital symmetry (λ), instead of one large one for the whole system. There are also

two types of density matrices, one for contributions from closed shell spinors ($\mathbf{D_C}$) and one for contributions from open shelled spinors ($\mathbf{D_O}$).

Using a procedure developed independently by Roothaan and Hall[10, 13], we first need to find the matrix $\mathbf{C}$ which satisfies

$$\mathbf{FC} = \epsilon\mathbf{SC} \tag{1.33}$$

where $\mathbf{F}$ is the Fock matrix, ϵ is the vector of eigenvalues, and $\mathbf{S}$ is the overlap matrix. The matrix elements of $\mathbf{S}$ are

$$\mathbf{S}_{p_\lambda,q_\lambda} = \langle p_\lambda|q_\lambda\rangle \tag{1.34}$$

and the non-relativistic $\mathbf{F}$ for closed shells of symmetry λ is given by

$$\mathbf{F}_{p_\lambda,q_\lambda} = \langle p_\lambda|\hat{h}(1)|q_\lambda\rangle + \sum_{\mu=1}^{nsym}\sum_{r=1}^{K}\sum_{s=1}^{K}\left(2\sum_{a=1}^{occ}\mathbf{C}_{r_\mu,a}\mathbf{C}_{s_\mu,a}^*\right)\left[(p_\lambda q_\lambda|r_\mu s_\mu) - \frac{1}{2}(p_\lambda q_\lambda|s_\lambda r_\lambda)\right] \tag{1.35}$$

where p, q, r, and s are basis functions, K is the total number of basis functions for symmetry μ, and occ is the number of occupied closed shells in symmetry μ. Because p and q will always be of the same symmetry, and likewise with r and s, the λ and μ subscripts will be dropped from these indices and instead will be obtained from context. As we can see, $\mathbf{F}$ is itself a function of $\mathbf{C}$ and we are left with a rather disappointing outcome, that in order to find $\mathbf{C}$, we first must know what it is. And so it would seem that all is lost and all of this hard work was for nothing. And yet, there is a glimmer of hope. $\mathbf{F}$ is made up of two terms, and as only the term corresponding to contributions from the two-electron integrals depends on $\mathbf{C}$, the term from one-electron integrals can be used independently of $\mathbf{C}$. Thus, we begin by assuming that $\mathbf{F}$ depends solely on the one-electron term, effectively assuming that $\mathbf{C} = \mathbf{0}$. Solving Equation 1.33 with the H^{core} hamiltonian matrix only will give us the initial guess of $\mathbf{C}$. With this, we can then resolve $\mathbf{F}$ with the guess of $\mathbf{C}$ and diagonalize the new $\mathbf{F}$ matrix to get another set of values for $\mathbf{C}$. We can then repeat this process over and over until $\mathbf{C}$ is no longer changing outside an acceptable tolerance at which point we say the calculation has converged. We have now just completed a Hartree-Fock calculation.

1.3   Special Relativity

Another field of physics that was being developed at the same time as quantum mechanics was special relativity (general relativity which involves gravity's effect on spacetime is far, *far* outside the scope of this thesis. I will therefore be referring to special relativity as just

relativity from now on). Relativity traces its roots to Michelson and Morley[14] who made the startling observation that all observers, regardless of their frame of reference, will always agree on the speed of light. But it would not be until everyone's favourite Swiss patent clerk decided to examine this issue that the truly profound implications of this observation would be known.

This is perhaps a bit of an aside, but it is simply too interesting to pass up. Consider this thought experiment: you and I are in two different spaceships in space. We will define your frame of reference such that you are stationary at the origin, and that I am flying past you at a constant speed that is arbitrarily close to the speed of light, we will call this speed v. In my spaceship, there is a laser that I have pointed perfectly perpendicular to the direction I am traveling. From my frame of reference, where I am stationary, I turn the laser on and observe photons that move from the laser to the side of my spaceship, a distance equal to d_1, in time equal to t at the speed of light c. Back in your frame of reference, you are also watching my experiment, but you make some different observations. In the time it take for the photons to move across my spaceship, I move a small distance d_2. So the total distance you see the photons move is not d_1, but d_3 which is the hypotenuse of the right angle triangle formed by d_1, d_2, and d_3. *But*, we both agree that the photons were moving at speed c! So we can not agree on the time it took to do so because $d_3 > d_1$. So you will observe the time it takes for the photons to move to be a time equal to t'. We can derive the relationship between t and t' in the following way.

$$d_1 = ct \qquad\qquad d_2 = vt' \qquad\qquad d_3 = ct' \qquad (1.36)$$

$$d_3^2 = d_1^2 + d_2^2 \qquad (1.37)$$

$$(ct')^2 = (vt')^2 + (ct)^2 \qquad (1.38)$$

$$(ct)^2 = (ct')^2 - (vt')^2 \qquad (1.39)$$

$$t = t'\sqrt{1 - \frac{v^2}{c^2}} \qquad (1.40)$$

Not does this give the absolutely remarkable result that time is relative, Equation 1.40 also implies that an object can have a maximum speed. But how could this be possible, if I turned on the engine on my spaceship, what's to stop me from just getting faster and faster? The answer is that the mass of an object increases the closer to the speed of light it is. This can be described by[15]

$$m(v) = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \qquad (1.41)$$

where $m(v)$ is the mass of an object moving at speed v and m_0 is the rest mass of the

object. With this we can see how it would be impossible for an object with mass to move at the speed of light. Its mass would go to infinity and would therefore need an infinite amount of energy to get there.

If we expand Equation 1.41 as a Taylor series centered at zero, we obtain the following equation

$$m(v) = m_0 \left[1 + \frac{1}{2}\frac{v^2}{c^2} + \frac{3}{8}\frac{v^4}{c^4} + \frac{5}{16}\frac{v^6}{c^6} + \frac{35}{128}\frac{v^8}{c^8} + \ldots \right] \tag{1.42}$$

which after rearranging, and multiplying both sides by c^2 gives

$$(m(v) - m_0)c^2 = \frac{m_0 v^2}{2} + m_0 \left[\frac{3}{8}\frac{v^4}{c^2} + \frac{5}{16}\frac{v^6}{c^4} + \frac{35}{128}\frac{v^8}{c^6} + \ldots \right] \tag{1.43}$$

So if we take the mass of an object at speed v and subtract the mass of the object and then multiply the result by c^2, we get the kinetic energy of the object (plus some terms that very quickly go to zero). This is what led Einstein to believe that the total kinetic energy of an object was stored in its mass. He would use this to produce the world's most famous equation

$$E_{kin} = m(v)c^2 \tag{1.44}$$

1.3.1 Relativistic Quantum Mechanics

It would seem that relativity concerns the physics of the very large and fast, while quantum mechanics is concerned with the very small. To merge the two, let us start from the beginning with the time-dependent one-dimensional Schrödinger equation which we will rewrite here in a more general and explicit form[1]

$$i\hbar \frac{\partial \Psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2 \Psi(x,t)}{\partial x^2} + \hat{V}(x)\Psi(x,t) \tag{1.45}$$

where $i\hbar\frac{\partial}{\partial t}$ is the energy operator, $-\hbar^2\frac{\partial^2}{\partial x^2}$ is the momentum squared operator ($\hat{p} = -i\hbar\frac{\partial}{\partial x}$), $\hbar$ is the reduced Planck's constant ($\hbar = \frac{h}{2\pi}$) and $\hat{V}(x)$ is the potential energy operator that the particle described by $\Psi(x,t)$ experiences. The reason we can not use the Schrödinger equation directly in relativistic quantum mechanics is because it treats time and space differently (it depends on only the first derivative of time, but the second derivative of space). In order to fix this, we may express Equation 1.44 in terms of $\hat{p}$ and get[15]

$$E_{kin}^2 = \hat{p}^2 c^2 + m_0^2 c^4 \tag{1.46}$$

If we want to solve for a free particle ($\hat{V}(x) = 0$), we could put the energy and momentum operator from Equation 1.45 into Equation 1.46 and multiply both sides from the right to obtain

$$-\hbar^2 \frac{\partial^2 \Psi(x,t)}{\partial t^2} = c^2 \left(-\hbar^2 \frac{\partial^2 \Psi(x,t)}{\partial x^2} + m_0^2 c^2 \Psi(x,t) \right) \tag{1.47}$$

which is called the Klein-Gordon[16, 17] equation for a free particle. We could also solve for a particle of charge q in a electromagnetic field described by potentials $\hat{A}$ and $\hat{\varphi}$ as

$$\left(i\hbar \frac{\partial}{\partial t} - q\hat{\varphi} \right)^2 \Psi(x,t) = c^2 \left[\left(-i\hbar \frac{\partial}{\partial x} - \frac{q}{c}\hat{A} \right)^2 + m_0^2 c^2 \right] \Psi(x,t) \tag{1.48}$$

Here we are treating time and space exactly the same, but there is a problem. While this equation works perfectly well for spinless particles, we can not ad hoc add spin to this equation like we did with the original Schrödinger equation without violating its Lorentz invariance, which is also important for relativistic theories. Also, because this equation solves for the *square* of the energy, it implies that the energy could both be positive or negative. This is a puzzling result that we will return to later, but for now let us examine how Dirac fixed the issue of introducing spin.

1.3.2   Dirac Equation

Dirac[18] began with the Klein-Gordon equation which may be written in a three-dimensional form as

$$\boldsymbol{\pi}_0^2 - \left[\sum_{u=x,y,z} \boldsymbol{\pi}_u^2 + m_0^2 c^2 \right] = 0 \tag{1.49}$$

$$\boldsymbol{\pi}_0 = \frac{i\hbar \frac{\partial}{\partial t} - q\hat{\varphi}}{c}, \quad \boldsymbol{\pi}_u = -i\hbar \frac{\partial}{\partial u} - \frac{q}{c}\hat{A} \tag{1.50}$$

Dirac wanted to write this equation as the product of one equation and its conjugate, which can be done as follows

$$\left(\boldsymbol{\pi}_0 + \left[\sum_{u=x,y,z} \alpha_u \boldsymbol{\pi}_u + \alpha_0 m_0 c \right] \right) \left(\boldsymbol{\pi}_0 - \left[\sum_{u=x,y,z} \alpha_u \boldsymbol{\pi}_u - \alpha_0 m_0 c \right] \right) = 0 \tag{1.51}$$

so long as we restrict α as

$$\alpha_i^2 = 1, \quad \alpha_i \alpha_j + \alpha_j \alpha_i = 0 \quad \text{for} \quad i \neq j \tag{1.52}$$

It's clear that the αs must be matrices. The simplest of these that meet the requirements of Equation 1.52 are

$$\alpha_i = \begin{pmatrix} \mathbf{0} & \sigma_i \\ \sigma_i & \mathbf{0} \end{pmatrix} \quad \text{for} \quad i \neq 0 \tag{1.53}$$

$$\alpha_0 \equiv \beta = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{pmatrix} \tag{1.54}$$

where σ_i are the Pauli spin matrices for $i = x, y, z$, $\mathbf{0}$ is a 2×2 matrix of zeros, and $\mathbf{I}$ is a 2×2 identity matrix[15]. We now have two different operator equations

$$\left(\boldsymbol{\pi}_0 + \sum_{u=x,y,z} \alpha_u \boldsymbol{\pi}_u + \beta m_0 c \right) \Psi(x, y, z, t) = 0 \tag{1.55}$$

$$\left(\boldsymbol{\pi}_0 - \sum_{u=x,y,z} \alpha_u \boldsymbol{\pi}_u - \beta m_0 c \right) \Psi(x, y, z, t) = 0 \tag{1.56}$$

Equation 1.55 will give negative energy and Equation 1.56 will give positive energy. The positive energy solutions correspond to electrons. The negative energy solutions correspond to antimatter, which is remarkable because antimatter would not be discovered until years later[19]. The stationary states (time-independent) for the electron then become

$$\left(E - V - \beta m_0 c^2 - c \sum_{u=x,y,z} \alpha_u \boldsymbol{\pi}_u \right) \Psi(x, y, z) = 0 \tag{1.57}$$

where $V = q\hat{\varphi}$. Introducing α as a 4×4 matrix means that Ψ must also have four components. It is typically written as

$$\Psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\psi} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\phi} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{pmatrix} \tag{1.58}$$

where $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ are called spinors. $\boldsymbol{\psi}$ is called the large component and $\boldsymbol{\phi}$ is the small component.

1.3.3 Kinetic Balancing

Recall that Equation 1.57 is only half of the total equation. We still have all of the negative energy solutions to deal with. If we were to try an use Equation 1.57 as is, the numerical methods we use of solving it would find these negative energies, and as there is no lower bound on the negative solutions, our solutions to the equation would all drop to negative infinity. We can solve this problem if we assume that all the infinite number of negative energy states are filled with an infinite number of positrons. This means that the electron can not drop into the negative energies due to the Pauli exclusion principle. Mathematically,

we can reproduce this idea using a procedure called kinetic balancing, which is done in the following way[15].

Using the spinors, we can write Equation 1.57 as a matrix multiplication problem of the form

$$\begin{pmatrix} (V + \beta m_0 c^2) & c(\boldsymbol{\sigma}\boldsymbol{\pi}) \\ c(\boldsymbol{\sigma}\boldsymbol{\pi}) & (V + \beta m_0 c^2) \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{pmatrix} = \begin{pmatrix} E & 0 \\ 0 & E \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{pmatrix} \tag{1.59}$$

where $\boldsymbol{\sigma}\boldsymbol{\pi} = \sum_{u=x,y,z} \alpha_u \boldsymbol{\pi}_u$. This is equivalent to adding the following two equations together

$$(E - V - m_0 c^2)\boldsymbol{\psi} - c(\boldsymbol{\sigma}\boldsymbol{\pi})\boldsymbol{\phi} = 0 \tag{1.60}$$

$$(E - V + m_0 c^2)\boldsymbol{\phi} - c(\boldsymbol{\sigma}\boldsymbol{\pi})\boldsymbol{\psi} = 0 \tag{1.61}$$

We can set where the energy is zero to anywhere we want, so let us make zero the energy of a free electron at rest ($\epsilon = E - m_0 c^2$). Equations 1.60 and 1.61 now become

$$(\epsilon - V)\boldsymbol{\psi} - c(\boldsymbol{\sigma}\boldsymbol{\pi})\boldsymbol{\phi} = 0 \tag{1.62}$$

$$(\epsilon - V + 2m_0 c^2)\boldsymbol{\phi} - c(\boldsymbol{\sigma}\boldsymbol{\pi})\boldsymbol{\psi} = 0 \tag{1.63}$$

We can rewrite Equation 1.63 as

$$\boldsymbol{\phi} = \left(1 + \frac{(\epsilon - V)}{2m_0 c^2}\right)^{-1} \frac{1}{2m_0 c^2} \boldsymbol{\sigma}\boldsymbol{\pi}\boldsymbol{\psi} \tag{1.64}$$

To a good approximation, $\epsilon - V$ will be much smaller than $2m_0 c^2$, so we can assume the term in parentheses is equal to 1. Therefore, the small component can be written in terms of just $\boldsymbol{\psi}$ and becomes

$$\boldsymbol{\phi} = \frac{\boldsymbol{\sigma}\boldsymbol{\pi}}{2m_0 c^2} \boldsymbol{\psi} \tag{1.65}$$

Not only does this make solving the Dirac equation much simpler (we only need to solve for $\boldsymbol{\psi}$ instead of $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$) it also "hides" the negative energies from numerical methods. Note, though, that it relies on the assumption that $\epsilon - V$ is much smaller than $2m_0 c^2$. If this is not the case, such as in the electrons in the heaviest known elements, then this procedure will fail and we will get energies that are lower than they should be. When this occurs, we say that there is variational prolapse of the wavefunction.

1.3.4 The Dirac Hydrogen-Like Atom

Up until now, we have only looked at what Dirac's theory says about a free electron (well, a "free" electron in the presence of an infinite amount of positrons). In the case of hydrogen-like atom solutions, the quantum numbers are similar to the non-relativistic theory. They are: the principal quantum number ($n = 1, 2, 3, \ldots$), the azimuthal quantum number ($l = 0, 1, 2, \ldots, n-1$), the angular momentum quantum number ($j = |l \pm \frac{1}{2}|$), and the magnetic quantum number ($m = -j, -j+1, \ldots, j-1, j$). The energy in terms of the quantum numbers is given by[15]

$$E_{n,j} = -\frac{1}{2n^2} \left[1 + \frac{1}{nc^2} \left(\frac{1}{j+\frac{1}{2}} - \frac{3}{4n} \right) \right] \tag{1.66}$$

It can be seen that there is a splitting in the energy levels of spinors with the same azimuthal number, whereas in the non-relativistic theory, orbitals with the same azimuthal number would all be perfectly degenerate.

1.3.5 Relativistic Effects

The effects of relativity have a dramatic effect on chemistry. Two classic examples are gold's distinctive colour, and mercury being the only metal that is liquid at room temperature. In the case of gold, its $6s_{1/2}$ electrons are lowered in energy and its $5d_{5/2}$ electrons are raised. This shifts its absorption region into the blue end of the spectrum, making it appear yellow[20]. The $6s_{1/2}$ electrons of mercury are similarly stabilized and since its $5d_{3/2}$ and $5d_{5/2}$ orbitals are filled, it behaves quite like a noble gas, making bonding quite difficult at room temperature[21].

Relativistic effects are broken up into two categories: primary and secondary effects. Primary effects are due to the contraction of the s and p electrons. The non-relativistic Bohr radius for the 1s electron of hydrogen is[3]

$$a_0 = \frac{4\pi\epsilon_0 \hbar^2}{m_0 e^2} \tag{1.67}$$

where ϵ_0 is the permittivity of free space, $\hbar$ is the reduced planks constant, e is the charge of an electron, and m_0 is the mass of an election at rest. Recall from Equation 1.41 that the mass of an object increases with its speed. If we substitute $m(v)$ from Equation 1.41 for m_0 into Equation 1.67 we get

$$a_v = \frac{4\pi\epsilon_0 \hbar^2 \sqrt{1 - \frac{v^2}{c^2}}}{m_0 e^2} \tag{1.68}$$

Dividing Equation 1.68 by 1.67 gives

$$\frac{a_v}{a_0} = \sqrt{1 - \frac{v^2}{c^2}} \tag{1.69}$$

which is less than 1 and will get closer and closer to zero as v increases (v will increase in proportion to the nuclear charge). This contraction stabilizes the electrons, making them more inert.

Secondary effects are a direct result of the electron density contraction. The contraction of these electrons leads to greater shielding of the nucleus, reducing its effective nuclear charge. This causes the d and f electrons to feel a smaller attraction to their nucleus, which makes their orbitals enlarge. This destabilizes these electrons, making them more reactive[15].

1.3.6 Gaussian Nucleus

Up till now, we have assumed the nucleus can be represented as a point. For many applications this works just fine, but it can make it difficult to find convergence when relativity is considered. Therefore, it is helpful to use a model more grounded in reality in this case. One such model is the Gaussian nucleus. This changes the potential in Equation 1.59 to

$$V(\mathbf{r}) = Z \left(\frac{\alpha}{\pi}\right)^{\frac{3}{2}} \int \frac{e^{-\alpha(\mathbf{r}-\mathbf{R})^2}}{|\mathbf{r} - \mathbf{R}|} d\mathbf{R} \tag{1.70}$$

where $\mathbf{r}$ is the coordinates of an electron, $\mathbf{R}$ is the coordinates of the nuclei, Z is the nuclear charge, and α is an estimation of the "size" of the nucleus. This is usually approximated using the root-mean-square (RMS) radius of the nuclei[22], which can be related to the mass of the nucleus by[23]

$$\sqrt{\langle R^2 \rangle} = \left(0.836 A^{1/3} + 0.570\right) \tag{1.71}$$

where $\langle R^2 \rangle$ is the mean-square radius in femtometers, and A is the mass of the nucleus. Converting the RMS into atomic units and substituting its value into the following equation gives the value of α[22]

$$\alpha = \frac{3}{2 \langle R^2 \rangle} \tag{1.72}$$

Chapter 2

cudaDFRATOM

2.1 Introduction

In this Chapter, I present a program that can optimize four-component relativistic basis sets. It was adapted from Matsuoka's and Watanabe's DFRATOM program[24]. My program was written in the CUDA Fortran programming language[25], and so I have called it cudaDFRATOM. In the following sections I give an introduction to programing with CUDA Fortran, and then I give a detailed explanation of the algorithms cudaDFRATOM uses. Finally, I show how my new program is able to outperform the original.

2.2 GPU Architecture

While general purpose computing on graphics processing units (GPGPU) (GPUs) has been adopted by the high performance computing (HPC) community for quite some time, it can seem quite complex to the uninitiated. While most quantum chemists who decide to dip their toes into computational waters can get away with having little to no understanding of what is actually going on under the hood when programming for a central processing uint (CPU), the same can not at all be said for GPUs. Therefore, in this section I will explain how a GPU works in order to help understand the terms and techniques used in the following chapter. As cudaDFRATOM was optimized for the NVIDIA Tesla C2050, I will be referring to the Tesla C2050 specifications for examples when needed. I will begin will the most granularity possible, and zoom out so that by the end of this section, the reader should come away prepared for the rest of this chapter. The bulk of this section has been adapted from [26, 25] and is based on my own experience.

2.2.1 Threads, Blocks, and Grids

The most granular element of computation on a GPU is the thread. When a CUDA function or subroutine (hereafter called a kernel) is called, it is a thread that actually executes the code. What makes GPGPU so powerful is that while a thread in a kernel is always executing the same code, the data a thread works with can be completely different between threads. This method of parallelism is called single instruction, multiple data (SIMD).

32 threads are organized into a structure called a warp. Within a warp, all threads execute code in lock-step. If a condition arises where some threads in warp must execute some code, and other threads in the same warp execute some other code (for instance in an **if else** statement), this leads to serialization of code execution, a process called warp divergence. In a worst-case scenario, this could cause all 32 threads to execute serially which could lead to a massive hit to performance.

Warp divergence can also cause warps to fall behind other warps and has the potential to cause race conditions to appear. A race condition appears when different threads need to use the same memory location, but there is nothing in the code to enforce the order the threads will access the memory location. If one warp needs to read data written by another warp that is several steps behind, this can cause all kinds of confusing errors to appear. Therefore, it is generally advisable to avoid warp divergence whenever possible. But, for programs of any considerable usefulness, warp divergence will be inevitable. Thankfully, NVIDIA has included the **syncthreads** command which provides a barrier that all threads within a block must meet before any can continue.

The next highest structure is the thread block, or block for short. Simply put, a block is a collection of one or more threads. When a kernel is called, the block scheduler assigns each block to a streaming multiprocessor (SM). How many blocks a SM can run at once depends on the resources each block uses (more on this in section 2.2.2), but no more than 8 blocks per SM can ever be run at once on the C2050. Blocks also add an element of dimensionality to threads through the thread index ($threadIdx$) variable. This variable has three parts, $threadIdx.x$, $threadIdx.y$, and $threadIdx.z$[1]. These variables make dealing with matrixes and arrays much simpler and can help with parallelizing large nested loops. The size of dimensions of the grid can be accessed through the $blockDim$ variable. $blockDim$ also has x, y, and z parts. These variables are all integer type, and the $threadIdx$ variables range from 1 in Fortran (or 0 in C/++) to the relevant part of $blockDim$ in Fortran (or $blockDim$ -1 in C/++). The $threadIdx$ set of variables are all reset at the boundary of each block.

Finally there is the grid. The grid is the entire collection of blocks that are launched for

---

[1]Note that in Fortran, the x, (or y or z) part of the $threadIdx$ variable would be written as $threadIdx\%x$. I find the C style dot notation to be more visually appealing, so that is what is used in this thesis.

a kernel. The grid gives a dimensionality to blocks through the *blockIdx* variable which has similar parts to the *threadIdx* variable. The grid also has the *gridDim* variable which is analogous to *blockDim*.

2.2.2   Memory

To simplify the discussion of the kinds of memory available on a GPU we will limit ourselves to the three main types. They are the global memory, the shared memory, and the register memory. Because the execution time of a calculation is often not limited by the actual number of floating point operations to perform, but rather the reading and writing to memory, it is essential to be sure that the three main types of memory are managed correctly.

Global memory is both the most plentiful and the slowest available memory on a GPU. In a typical calculation, global memory will be allocated in code executed by the CPU (host). Data will be uploaded to the allocated memory and then used by the GPU (device). Global memory is distinct from the other kinds of device memory in that it can be read from and written to by all threads currently executing. This can allow threads of one block to pass messages to threads in an other block, but great care must be taken to avoid race conditions. There is an additional caveat with global memory in that if threads are reading from non-contiguous memory locations, then memory bandwidth plummets. There is also a problem when multiple threads try to read from the same location. Instead of being one transaction, the read is serialized which leads to a massive performance penalty. It is therefore to the programmers advantage to write their kernels with as little use of the global memory as possible.

Shared memory, as its name implies, is shared and private to threads in a single block. It can be accessed much faster than global memory, but this comes at the cost of a much reduced volume. The amount of it available per SM varies with hardware generation, but the maximum available on our card is 48KB. This amount is shared among all blocks that are running on the SM, so the amount available per block will typically be even less. Also, there is no guarantee that memory loaded by one thread will be available to all other threads right away, only a call to **syncthreads** ensures proper loading. But, when used properly, shared memory is indispensable for speeding up calculations. It does not suffer from threads reading from non-contiguous memory locations like global memory does, and if all threads in a warp read from the same memory location a "broadcast" occurs. This means that instead of the read being serialized, all threads are able to read the data in one transaction. Shared memory is also distinct from global memory in that instead of being declared in the host code, it is declared in the device code. This is done by giving the variable the "shared" attribute in its declaration. However, one must be careful not to overuse the shared memory

however. As mentioned, there is only 48KB available per SM. If a block needs more than is available, the function will fail to execute. It can also affect the occupancy of the kernel, a concept that will be explained later on.

Lastly there is the register memory. This is the fastest memory available and is also private to an individual thread. Figure 2.1 shows the hierarchy of the grid, blocks, and threads, as well as how each memory type is localized. On the Tesla C2050, there are a total of 32768 32-bit registers available per SM. An individual thread in a block can use no more than 63 registers, and if the number of threads in a block times the number of registers needed per thread for a kernel is larger than the total registers available per SM then the execution of the kernel will fail. If a thread needs more memory than the registers will allow, then "register-spilling" takes place. This means that instead of a variable being stored into a register, it "spills over" into a fourth memory type called local memory. Local memory is really just a section of the global memory and comes with all the problems that global memory has, but the compiler is able to arrange it such that it can be read efficiently without the programmer having to worry about this. There is no way for a programmer to specify which variables will be stored in register or local memory, the compiler handles all of this. There are compiler flags which can change the maximum amount of registers per thread allowed which will prompt the compiler to place more data into local memory. While this may sound like a bad idea, we will see in the next section how there are situations where this might be useful.

2.2.3   Occupancy and Performance

Having described the various resources, we can talk about how they can be used most effectively. The occupancy of a kernel is the number of warps that are actually able to run on a SM divided by the maximum possible warps an SM can run. Occupancy is important, as it determines the maximum number of blocks, and thus threads that are able to run simultaneously. The occupancy of a kernel is determined by three factors: the block size, the amount of shared memory per block, and the number of registers used per thread. While block size and shared memory are explicitly set by the programmer, it is next to impossible to guess how registers will be used before compile time. Thankfully, there are compiler flags that will display this information.

Armed with the resources our kernel will need, we can now determine its occupancy. The "CUDA Occupancy Calculator" is an Excel spreadsheet provided by NVIDIA and is an indispensable tool. We can input the needed resources and the hardware generation of our card and then receive the occupancy of the current resource configuration, where the bottleneck of the current configuration is, as well as information on how the occupancy will

Figure 2.1: *Architecture of a GPU. Organization of threads and blocks in the grid, and the location of different memory types within them. The grid is coloured dark blue, blocks are coloured light blue, and threads are coloured cyan. The global memory is coloured red, shared memory is coloured orange, and register memory is coloured yellow. The numbers in parenthesis indicate the block or thread's blockIdx or threadIdx variables. In this particular case, blockDim = (8, 4) and gridDim = (2, 2). A thread is able to access memory that is inside its registers, but not the registers of other threads, and memory that is inside the shared memory of its block, but not the shared memory of other blocks. All threads can access all of the global memory.*

change upon tweaking the configuration. In some sense, the best block size is determined by how the other resources are used. Therefore, in order to maximize the occupancy, we must first make sure that we are correctly using the memory provided.

Registers are the fastest memory available, but they are also the easiest to overuse. Therefore, they are best suited for small variables that need to be referenced frequently, such as accumulators. Shared memory is best used for storing large sections of non-contiguous data stored in the global memory. Finally data that is able to be stored such that it can be accessed effectively, or is infrequently used might best be left in the global memory.

As with everything, these rules are only guidelines. The best resource configuration will be individual not only to every kernel, but also to the GPU it is to be run on. The best way to maximize the efficiency of a calculation is to make minor alterations to where data is stored and play around with the code. Finally, it might not always be the best idea to maximize the occupancy of a kernel. If the bottleneck is not the actual calculation, but how fast the data can be read, then it might be advantageous to fill up the registers and shared memory as much as possible and have a very low occupation. The only way to know is to try!

2.2.4 Matrix Multiplication Example

To complete the introduction to CUDA programming, we will go over a real example of CUDA Fortran code. For our example, we will examine code that performs matrix multiplication on matrices of arbitrary dimensions. This makes use of all the concepts we've covered so far, so it will be a good exercise to examine. The code is given below.

```fortran
module mat_mul
  implicit none
  integer, parameter  ::  dp = kind(0.d0)
  integer, parameter  ::  block_dim = 16
  integer, constant ::  rowA, colA, rowB, colB
  contains
    attributes(global) subroutine mat_mul_kernel(inputA, inputB, output)
      real(dp), dimension(:,:)  ::  inputA, inputB, output
      real(dp)  ::  total
      real(dp), shared, dimension(block_dim, block_dim) ::  s_tileA, s_tileB
      integer ::  x, y, i, j

      x = threadIdx%x + blockDim%x * (blockIdx%x - 1)
      y = threadIdx%y + blockDim%y * (blockIdx%y - 1)
      total = 0.0_dp

      do i = 0, colA - 1, block_dim
        call syncthreads()
```

```fortran
            if ((x <= rowA) .and. (threadIdx%y + i <= colA)) then
               s_tileA(threadIdx%x, threadIdx%y) = inputA(x, threadIdx%y + i)
            else
               s_tileA(threadIdx%x, threadIdx%y) = 0.0_dp
            endif

            if ((threadIdx%x + i <= rowB) .and. (y <= colB)) then
               s_tileB(threadIdx%x, threadIdx%y) = inputB(threadIdx%x + i, y)
            else
               s_tileB(threadIdx%x, threadIdx%y) = 0.0_dp
            endif

            call syncthreads()

            do j = 1, block_dim
               total = total + s_tileA(threadIdx%x, j) * s_tileB(j, threadIdx%y)
            enddo

         enddo

         if((x <= rowA) .and. (y <= colB)) then
            output(x, y) = total
         endif
      end subroutine mat_mul_kernel
end module mat_mul

program main
   use cudafor
   use mat_mul
   implicit none
   ! Host variables
   real(dp), allocatable, dimension(:, :)  ::  inputA, inputB, output
   ! Device variables
   real(dp), device, allocatable, dimension(:, :)  ::  d_inputA, d_inputB, &
     d_output
   ! For generating random numbers
   integer, allocatable, dimension(:)  ::  seed
   integer ::  n
   ! block and grid dimensions
   type(dim3)  ::  blockDim, gridDim

   rowA = 1024
   colA = 432
   rowB = colA
   colB = 1783

   ! Set up random number generator
```

```
65    call random_seed(size = n)
66    allocate(seed(n))
67    ! Set seed to a constant so we always get the same matrices
68    seed = 150987
69    call random_seed(put = seed)
70    deallocate(seed)
71
72    ! Allocate host matrices and set up matrices
73    allocate(inputA(rowA, colA))
74    allocate(inputB(rowB, colB))
75    allocate(output(rowA, colB))
76
77    call random_number(inputA)
78    call random_number(inputB)
79
80    ! Allocate and upload device data
81    allocate(d_inputA(rowA, colA))
82    allocate(d_inputB(rowB, colB))
83    allocate(d_output(rowA, colB))
84    d_inputA = inputA
85    d_inputB = inputB
86
87    blockDim = dim3(block_dim, block_dim, 1)
88    gridDim = dim3((rowA - 1) / block_dim + 1, (colB - 1) / block_dim + 1, 1)
89
90    call mat_mul_kernel <<<gridDim, blockDim>>> (d_inputA, d_inputB, d_output)
91
92    ! Download device data
93    output = d_output
94
95    ! Deallocate variables
96    deallocate(d_inputA)
97    deallocate(d_inputB)
98    deallocate(d_output)
99    deallocate(inputA)
100   deallocate(inputB)
101   deallocate(output)
102 end program main
```

Listing 2.1: *Matrix multiplication kernel*

Lines 1 - 43 define a module which contains a matrix multiplication kernel. The kernel takes the matrices `inputA` and `inputB`, multiplies them, and stores the result in `output`. The kernel has the `attributes`(`global`) attribute, which means that it can be called from either device or host code. `rowA`, `colA`, `rowB`, and `colB` are the rows and columns of `inputA` and `inputB`. Because they are small and don't change over the course of the calculation, they get stored in a special memory type called constant memory. This is a small cache of read

24

only memory that is 64KB in size and allows for constant values to be read very quickly. There are two shared variables, `s_tileA` and `s_tileB`. Each of these are square matrices with dimensions of `block_dim` and these will be used to store sections or "tiles" of `inputA` and `inputB` respectively. We want each thread to do all the calculations needed to compute a simple element of `output`. Line 13 and 14 use the `threadIdx`, `blockDim`, and `blockIdx` variables so that each thread gets a unique combination of `x` and `y` values. `x` corresponds to a row of the `output` matrix and `y` corresponds to a column, so each thread is then uniquely mapped to an element of `output`. We will see in the host code how these index values are determined, but for now it is sufficient to know that there will be at least as many threads as elements of `output`.

Lines 17 - 37 are where the actual calculation takes place. For each iteration of the loop, a different tile of `inputA` and `inputB` is fetched from the global memory and stored into the shared memory. The reason for using this tiling approach can be seen when we consider how the threads must read `inputB`. If we did not use shared memory, the loop would just be `total = total + inputA(x, i) * inputB(i, y)`. In this case, access to `inputA` is coalesced, but access to `inputB` is not. Therefore, the kernel will be stunted due this unoptimized memory usage. Instead, we load tiles in a coalesced manner into the shared memory, synchronize the threads, and then read through the shared memory in an uncoalesced fashion which carries no penalty. At the end of each loop, the tiles are moved in the `y` direction for `inputA` and in the `x` direction for `inputB`, and the process repeats. Notice how it is possible that a tile might fall on the boundary of an `input` matrix. This is why there are checks to make sure the coordinates of the tile don't exceed the boundaries of a matrix so we don't start reading garbage.

In the host code on lines 45 - 102, we have two sets of variables, those for the host and those for the device. Notice how device variables have the `device` attribute. For clarity, they are given the same name as their host variables but with a `d_` prefix. We now assign constant variables from line 5 to arbitrary values. Next we set up the random number generator, allocate the host matrices memory, and randomly initialize the input matrices. Then the device variables are allocated memory and the host data is then uploaded to the GPU. The block and grid dimensions are set up on lines 87 and 88. `blockDim` and `gridDim` are variables of `type`(dim3) which is a tuple that is defined in the `cudafor` module. The block has two dimensions containing `block_dim`^{2} threads. The grid also has two dimensions, and as we need one thread per element of `output` we perform a calculation which returns the minimum number of blocks that will cover it. Finally, we launch the kernel on line 90 using the `<<< >>>` operators to provide the grid and block dimensions, download the result from the GPU, and deallocate the memory.

2.3 Program flow in cudaDFRATOM

In this section, the control flow of a typical calculation is given.

Step 1. The input file is read by the ***intin*** subroutine. If needed, the basis set and open-shell configurations are calculated by ***formbs*** and ***find_bin_configurations*** respectively. The options set in the input file are rewritten to stdout.

Step 2. The calculation of small arrays and other constants is performed by ***calc_parameters*** and ***bsnorm***. ***calc_parameters*** calls ***bcoef*** and ***setvc***.

Step 3. The mapping of threads to the unrolled one and two-electron integral matrixes are calculated by ***lmpqrsa*** on the GPU. All other data from step 1 and 2 is then uploaded to the GPU.

Step 4. The one and two electron integrals are calculated on the GPU by ***eint1gpu*** and ***eint2gpu*** respectively.

Step 5. The initial guess of the density matrix is calculated by ***guess*** on the GPU. This is done by the diagonalization of the one-electron Hamiltonian matrix.

Step 6. ***scfiter*** then performs the SCF iterations until convergence of the density matrix has been reached, or the maximum number of iterations has been reached. SCF is performed with cuSOLVER functions, as well as a few custom helper functions. The converged eigenvectors, values, and energies are downloaded from the GPU and then written to stdout.

Step 7 (optional). If jobtyp='bsopt', then an optional basis set optimization is carried out. This starts by assigning pointers to variables on the CPU and GPU through ***hookup_cpu*** and ***hookup_gpu***. Then, the four well-tempered basis set (WTBS) parameters are optimized by ***newuoa***. ***newuoa*** calls ***calc_energy*** which essentially reconstructs the basis set from new parameters from ***newuoa***, then repeats steps 2 - 6 and feeds the energy back into ***newuoa***. This repeats until optimal WTBS parameters have been found.

2.4 Alterations of DFRATOM for CUDA

While almost all of the code from the original DFRATOM[24] was modified in some way, the most extreme changes were the integral evaluation and the formation of the P and Q matrixes. Therefore, we will discuss these changes in detail in this section.

2.4.1 Two-electron Integrals

The original DFRATOM calculates the two electron integrals with a long series of nested **for** loops. Working from the outside in, the indices of the loops are L, P, Q, M, R, and S where L and M are spinor symmetries, P and Q are basis functions of symmetry L, and R and S are basis functions of symmetry M. The pseudocode for these is shown in Algorithm

1 where $nsym$ is the total number of symmetries, and $nbs(i)$ is the number of basis functions for symmetry species i. Thus, each set of J and K integrals is uniquely defined by its L, M, P, Q, R, and S values. In order to have an effective CUDA implementation of this algorithm, we need to find a way to map these six numbers to CUDA threads. The simplest approach would be to map the values of L, P, and Q onto the $threadIdx.x$, $threadIdx.y$, and $threadIdx.z$ variables for each thread, launch the needed number of blocks, and then have each thread loop over the remaining indices. The pseudocode for this can be seen in Algorithm 2. A similar method is employed by many other programs, and for larger systems it works perfectly well. But because this program is for only single atoms, problems begin to appear.

Algorithm 1 The original

    for $L = 1$ to $nsym$ **do**
      for $P = 1$ to $nbs(L)$ **do**
        for $Q = 1$ to P **do**
          for $M = 1$ to L **do**
            if $L = M$ **then**
              $maxr = P$
            else
              $maxr = nbs(M)$
            end if
            for $R = 1$ to $maxr$ **do**
              if $(L = M)$ **and** $(P = R)$ **then**
                $maxs = Q$
              else
                $maxs = R$
              end if
              for $S = 1$ to $maxs$ **do**
                compute the J and K integrals of L, M, P, Q, R, and S
              end for
            end for
          end for
        end for
      end for
    end for

The first problem arises due to warp divergence: as the maximum values of M, R, and S depend on L, P, and Q, different threads will have a different number of loops to complete than others. Because a streaming multiprocessor (SM) must finish the block it is currently working on before it can grab another, there is the possibility that most of the threads in a block are idling while waiting for others in the same block to finish. The second problem with this method is that there will always be several blocks which have threads that remain

Algorithm 2 Easy code

$L = threadIdx.x + (blockIdx.x - 1) * blockDim.x$
$P = threadIdx.y + (blockIdx.y - 1) * blockDim.y$
$Q = threadIdx.z + (blockIdx.z - 1) * blockDim.z$

if $(L \leq nsym)$ **and** $(P \leq nbs(L))$ **and** $(Q \leq P)$ **then**
  for $M = 1$ to L **do**
    if $L = M$ **then**
      $maxr = P$
    else
      $maxr = nbs(M)$
    end if
    for $R = 1$ to $maxr$ **do**
      if $(L = M)$ **and** $(P = R)$ **then**
        $maxs = Q$
      else
        $maxs = R$
      end if
      for $S = 1$ to $maxs$ **do**
        compute the J and K integrals of L, M, P, Q, R, and S
      end for
    end for
  end for
end if

---

idle throughout the block's runtime, no matter what. This can be seen more clearly in Figure 2.2. The third problem with this is that with the maximum values for L, P, and Q available for a single atom, there might not even be enough combinations to completely fill the GPU. While all of these issues begin to disappear once the number of integrals to evaluate becomes large enough, we are still very much in the range where they are in play. Therefore a smarter algorithm had to be used.

The second problem can be solved by restricting the dimensionality of the solution instead of the using the three dimensional one of Algorithm 2. We might consider using a two dimensional solution, but then we still will have blocks with idle threads which can be seen in Figure 2.3. By restricting ourselves to only using $threadIdx.x$, we ensure that only the last block to run will have the possibility of having threads remaining idle throughout the block's runtime as shown in Figure 2.4.

The first and third problems can be solved by having each thread calculate one and only one set of J and K integrals. If we start with a valid combination of L, M, P, Q, R, and S values, we can very easily figure out which thread will calculate that set of integrals by using the following equations:

$$n'(j) = \frac{j^2 + j}{2} \tag{2.1}$$

$$y = n'(nbs(L-1)) + n'(P-1) + Q \tag{2.2}$$

$$x = n'(nbs(M-1)) + n'(R-1) + S \tag{2.3}$$

$$i = n'(x) + y \tag{2.4}$$

$$i_{max} = n'(\sum_{j=1}^{nsym} n'(nbs(j))) \tag{2.5}$$

where $nbs(0) = 0$ and $i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$. Starting with a value of i and working our way back is a much more challenging task. It becomes easier if we reframe it in the following way.

Figure 2.2: *Two-electron integral layout: naive implementation. Mapping of integrals to threads using Algorithm 2. The basis set used had nsym = 7 with nbs(i) = (30, 25, 25, 20, 20, 15, 15). The blockDim = (1, 32, 32) and gridDim = (7, 1, 1). Each histogram is one block of threads, where the horizontal axes are the threadIdx.y and threadIdx.z variables, and the vertical axis is the number of Two-Electron Integrals it must compute. Notice that in all blocks there are threads that calculate no integrals and therefore are idle throughout the lifetime of the block. Also note that the number of integrals each thread must calculate grows both from block to block, and from thread to thread within a block. Both of these factors make this algorithm an undesirable choice.*
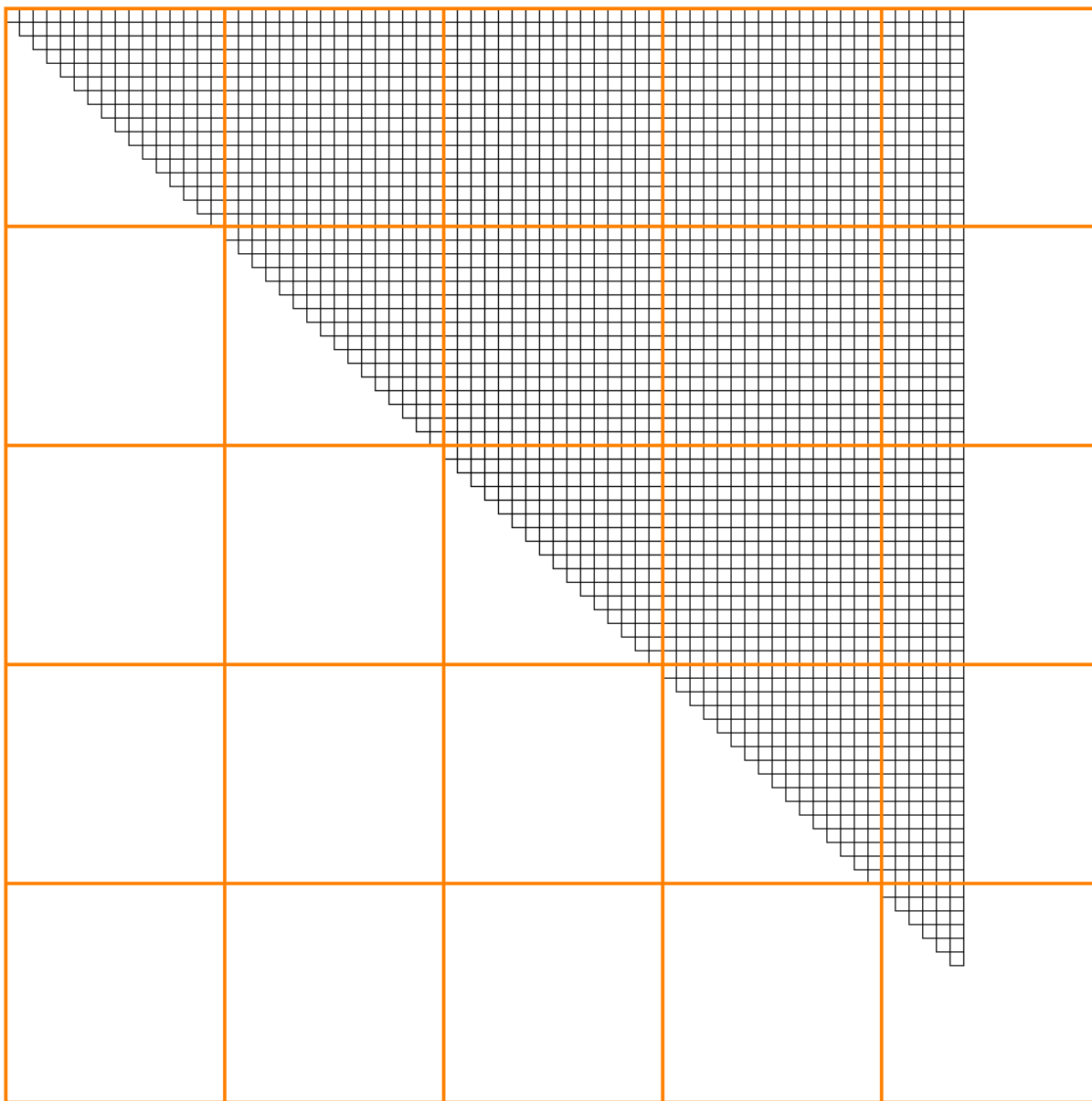
Figure 2.3: *Two-electron integral layout: 2D blocks and grid. Orange lines indicate the border of blocks. Notice that many blocks have idle threads.*
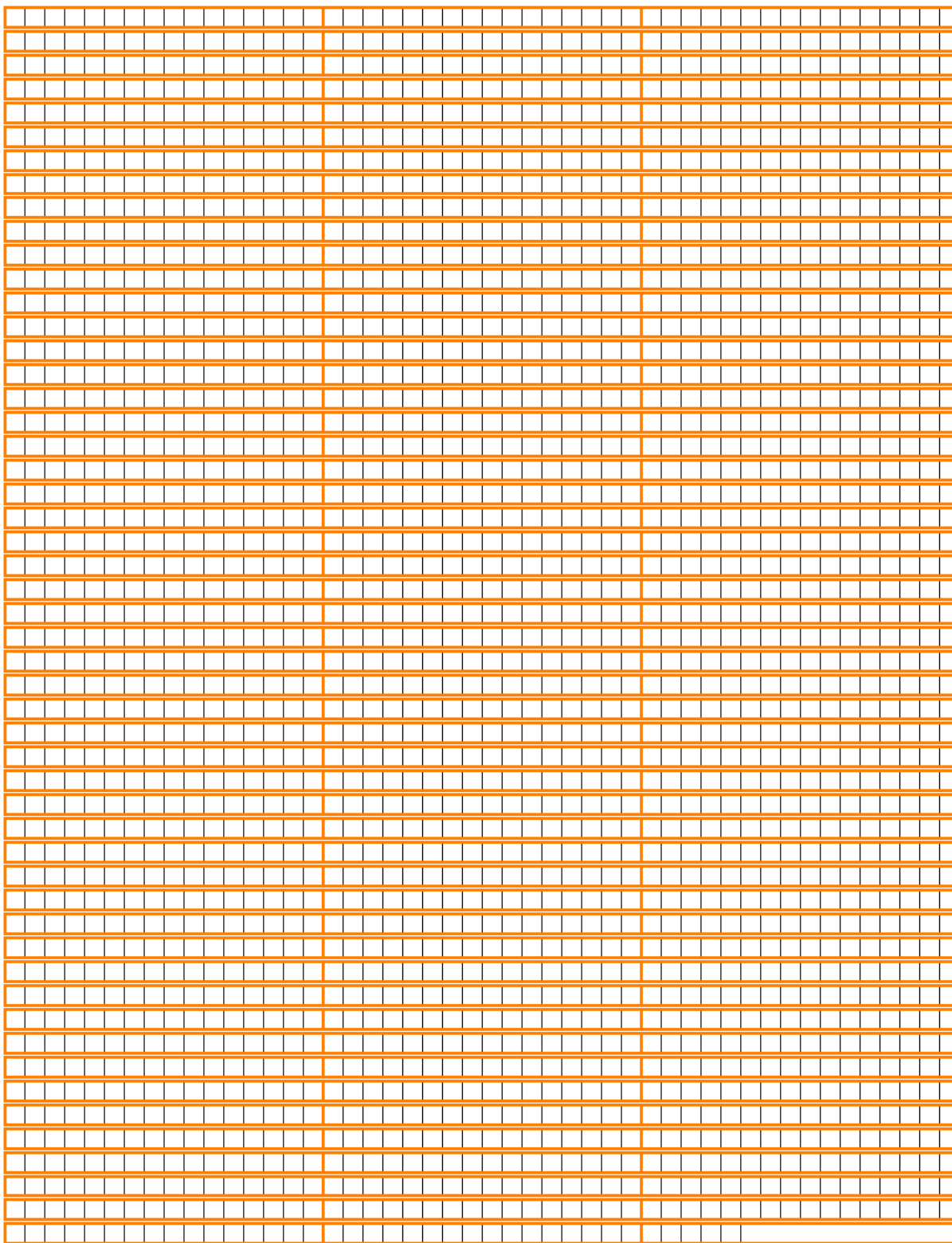
Figure 2.4: *Two-electron integral layout: 1D blocks and grid. Orange lines indicate the border of blocks. Notice that only the last block contains idle threads.*

Consider Figures 2.5 and 2.6. Figure 2.5 shows the top half of the symmetric two-electron integral matrix for a problem with $nsym = 2$ and 3 basis function for symmetry one, and 2 for symmetry two. Figure 2.6 shows the same but mapped onto a vector representation. In each element of both figures there is a set of seven numbers. The top two are L and M, then P and Q, then R and S, and the last number is the value of i for the thread calculating that integral. With this, it can be seen from the matrix representation that each element in the same row have the same M, R, and S values and the elements in the same column have the same L, P, and Q values. Therefore, finding out which column the element belongs to gives us the value for y, and finding the row give us the value of x. This can be done with the binary search algorithm shown in Algorithm 3.

Algorithm 3 Binary search for x and y

---

  if $theadIdx.x \leq nsym$ **then**
    $s\_nsym = nsym$
    $s\_nbs(threadIdx.x) = nbs(threadIdx.x)$
    $s\_nprime(threadIdx.x) = n'(s\_nbs(threadIdx.x))$
  end if
  call **syncthreads**
  $i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$
  if $i \leq i_{max}$ **then**
    $low = 1$
    $high = \textbf{sum}(s\_nprime(1 : s\_nsym))$
    while $low \leq high$ **do**
      $mid = \frac{(low+high)}{2}$
      $lownum = \frac{(mid-1)(mid-2)}{2} + mid$
      $highnum = lownum - 1 + mid$
      if $(i \leq highnum)$ **and** $(i \geq lownum)$ **then**
        $y = mid$
        exit
      else if $i > highnum$ **then**
        $low = mid + 1$
      else if $i < lownum$ **then**
        $high = mid - 1$
      end if
    end while
    $x = i - lownum + 1$
  end if

In this algorithm, variables with the $s\_$ prefix refer to those in the shared memory, and *lownum* and *highnum* refer to the minimum and values that i could have for the current guess (mid) of y. From here, L can be found with Algorithm 4, and P and Q can be found with Algorithm 5. The same set of algorithms can then be used to get M, R, and S

Figure 2.5: *Two-electron integral matrix as a matrix: The top triangle of the two-electron symmetric matrix in a matrix representation. In this instance, the total number of symmetries is 2, with 3 basis functions for the first symmetry and 2 basis functions for the second. The key for translating the numbers in each box into the L, M, P, Q, R, and S values is given in the bottom left. Notice how all rows have the same M, R, and S values, while all columns have the same L, P, and Q values. For clarity, the boxes have been coloured, based on the spinor symmetries that they contain.*

by substituting the relevant variables. If there is sufficient global memory available, these values can be stored and referred to later as needed. Otherwise, they could be calculated on the fly as needed. Because binary search scales as $\mathcal{O}(n \log n)$, this should ensure that this

Figure 2.6 grid (each box key: L M / P Q / R S, with box index I):

I	L	M	P	Q	R	S
1	1	1	1	1	1	1
2	1	1	2	1	1	1
3	1	1	2	1	2	1
4	1	1	2	2	1	1
5	1	1	2	2	2	1
6	1	1	2	2	2	2
7	1	1	3	1	1	1
8	1	1	3	1	2	1
9	1	1	3	1	2	2
10	1	1	3	1	3	1
11	1	1	3	2	1	1
12	1	1	3	2	2	1
13	1	1	3	2	2	2
14	1	1	3	2	3	1
15	1	1	3	2	3	2
16	1	1	3	3	1	1
17	1	1	3	3	2	1
18	1	1	3	3	2	2
19	1	1	3	3	3	1
20	1	1	3	3	3	2
21	1	1	3	3	3	3
22	2	1	1	1	1	1
23	2	1	1	1	2	1
24	2	1	1	1	2	2
25	2	1	1	1	3	1
26	2	1	1	1	3	2
27	2	1	1	1	3	3
28	2	2	1	1	1	1
29	2	1	2	1	1	1
30	2	1	2	1	2	1
31	2	1	2	1	2	2
32	2	1	2	1	3	1
33	2	1	2	1	3	2
34	2	1	2	1	3	3
35	2	2	2	1	1	1
36	2	2	2	1	2	1
37	2	1	2	2	1	1
38	2	1	2	2	2	1
39	2	1	2	2	2	2
40	2	1	2	2	3	1
41	2	1	2	2	3	2
42	2	1	2	2	3	3
43	2	2	2	2	1	1
44	2	2	2	2	2	1
45	2	2	2	2	2	2

Key:

L M
P Q
R S
I

Figure 2.6: *Two-electron integral matrix as a vector: The top triangle of the two-electron symmetric matrix in a vector representation. In this instance, the total number of symmetries is 2, with 3 basis functions for the first symmetry and 2 basis functions for the second. The key for translating the numbers in each box into the L, M, P, Q, R, and S values is given in the bottom center. Notice how the L, M, P, Q, R, and S values are much harder to predict than from the matrix representation. For clarity, the boxes have been coloured based on the spinor symmetries that they contain. Gaps between rows are used to reinforce that the elements are not connected like a matrix.*

remains a fast method of mapping threads to integrals for large problems as well. With some alterations, this method could also apply to molecular symmetries and not only to a single atom. For instance in the C_1 symmetry, all possible combinations of four basis functions must be used (ignoring those that appear on the bottom triangle of the two electron integral matrix of course). We could simply remove the search for L and M, have the initial value of *high* in Algorithm 5 be the total number of basis functions, and remove the **sum**($s\_nprime(1 : L-1)$) term from *lownum*.

From here, the new code for evaluating the integrals remains largely the same as the

---
Algorithm 4 Binary search for L

$i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$
if $i \leq i_{max}$ **then**
   $low = 1$
   $high = s\_nsym$
   while $low \leq high$ **do**
      $mid = \frac{(low+high)}{2}$
      $lownum = 1 + \mathbf{sum}(s\_nprime(1:mid-1)$
      $highnum = lownum - 1 + s\_nprime(mid)$
      if $(i \leq highnum)$ **and** $(i \geq lownum)$ **then**
         $L = mid$
         exit
      else if $y > highnum$ **then**
         $low = mid + 1$
      else if $y < lownum$ **then**
         $high = mid - 1$
      end if
   end while
end if

Algorithm 5 Binary search for P and Q

$i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$
if $i \leq i_{max}$ **then**
   $low = 1$
   $high = s\_nbs(L)$
   while $low \leq high$ **do**
      $mid = \frac{(low+high)}{2}$
      $lownum = \frac{(mid-1)(mid-2)}{2} + mid + \mathbf{sum}(s\_nprime(1:L-1))$
      $highnum = lownum + mid - 1$
      if $(i \leq highnum)$ **and** $(i \geq lownum)$ **then**
         $P = mid$
         exit
      else if $y > highnum$ **then**
         $low = mid + 1$
      else if $y < lownum$ **then**
         $high = mid - 1$
      end if
   end while
   $Q = y - lownum + 1$
end if

---

original, except for some minor changes to allow for more efficient global or shared memory access. We also use a process referred to as "grid-stride looping" where all these binary search algorithms have their **if** $i \leq i_{max}$ **then** removed, and then are placed within the following loop: **for** $i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$ to i_{max}, $i \mathrel{+}= blockDim.x * gridDim.x$ **do**. If we know the occupancy of the algorithm on the GPU beforehand, we can launch exactly the number of blocks that will fill the GPU. This reduces the overhead of block swapping and lets us further eke out some performance.

2.4.2 Density Matrix Formation

The term in parenthesis in Equation 1.35 corresponds to the non-relativistic density matrix. But as we are interested in relativistic calculations, it will need to be altered. Firstly, because a relativistic wavefunction has both large and small components, so too does its density matrix. Therefore, it will have two dimensions that span from 1 to $2K$ where dimensions 1 to K are the large components and the rest are for small components. Secondly, the factor of 2 is replaced by the number of electrons that can occupy spinors of symmetry μ. The density matrix's matrix and vector representations are shown in Figures 2.7 and 2.8 respectively. The mathematical form of the density matrix is given in Equation 2.6[24]

$$\mathbf{D}_{\mathbf{C}r,s} = N_\mu \sum_{a=1}^{occ} \mathbf{C}_{r,a}\mathbf{C}_{s,a}^* \tag{2.6}$$

where N_i is the number of electrons that occupy a closed shell of symmetry i. $\mathbf{D_O}$ is similar to this and is shown below.

$$\mathbf{D}_{\mathbf{O}r,s} = N_i \mathbf{C}_{r,(occ+1)}\mathbf{C}_{s,(occ+1)}^* \tag{2.7}$$

where N_i is equal to the number of electrons in the open shell. Note that in this program, only the most energetic shell of any symmetry can be open.

The total density matrix is $\mathbf{D_T} = \mathbf{D_C} + \mathbf{D_O}$

2.4.2.1 P and Q Supermatrices

With the density matrices in hand we can begin to combine the two-electron integrals from the previous section. These integrals have the following form.[24]

Figure 2.7: *Density matrix as a matrix. The top triangle of the symmetric density matrix in a matrix representation. The key for translating the numbers in each box into the R and S values is given in the bottom right. All active symmetries would have a similar, separate matrix. For clarity, the elements have been colorized based on the combination of large and small components they contain.*

Figure 2.8: *Density matrix as a vector. The top triangle of the symmetric density matrix in a vector representation. The key for translating the numbers in each box into the R and S values is given in the bottom left. All active symmetries would have a similar, similar vector that continue one after the other. For clarity, the elements have been colorized based on the combination of large and small components they contain.*

$$\mathbf{X}^1_{I(pqrs)} = (p_L q_L | r_L s_L) - \frac{1}{2} \left[(p_L r_L | q_L s_L) + (p_L s_L | q_L r_L) \right]$$

$$\mathbf{X}^2_{I(pqrs)} = (p_L q_L | r_S s_S)$$

$$\mathbf{X}^3_{I(pqrs)} = (p_S q_S | r_L s_L)$$

$$\mathbf{X}^4_{I(pqrs)} = - (p_L r_S | q_S s_L)$$

$$\mathbf{X}^5_{I(pqrs)} = - (p_L s_S | q_S r_L)$$ \hfill (2.8)

$$\mathbf{X}^6_{I(pqrs)} = - (p_S r_L | q_L s_S)$$

$$\mathbf{X}^7_{I(pqrs)} = - (p_S s_L | q_L r_S)$$

$$\mathbf{X}^8_{I(pqrs)} = (p_S q_S | r_S s_S) - \frac{1}{2} \left[(p_S r_S | q_S s_S) + (p_S s_S | q_S r_S) \right]$$

where the subscripts L and S refer to the large and small components respectively. I is a function that returns the values of x and y from Equations 2.2 and 2.3 where $L = \lambda$ and $M = \mu$. From here, all subscripts of $\mathbf{X}$ will be implied to go through this I function.

The $\mathbf{P}$ and $\mathbf{Q}$ super matrices have a similar layout to the density matrices. The elements of $\mathbf{P}$ have the following form[24]

$$\mathbf{P}^{LL}_{pq} = \sum_{\mu=1}^{nsym} \sum_{r=1} \sum_{s=1} \left(\mathbf{D}^{LL}_{\mathbf{T}rs} \mathbf{X}^1_{pqrs} + \mathbf{D}^{SS}_{\mathbf{T}rs} \mathbf{X}^{J(pqrs)}_{pqrs} \right)$$

$$\mathbf{P}^{SS}_{pq} = \sum_{\mu=1}^{nsym} \sum_{r=1} \sum_{s=1} \left(\mathbf{D}^{SS}_{\mathbf{T}rs} \mathbf{X}^8_{pqrs} + \mathbf{D}^{LL}_{\mathbf{T}rs} \mathbf{X}^{J(pqrs)}_{pqrs} \right)$$ \hfill (2.9)

$$\mathbf{P}^{LS}_{pq} = \sum_{\mu=1}^{nsym} \sum_{r=1} \sum_{s=1} \left(\mathbf{D}^{LS}_{\mathbf{T}rs} \mathbf{X}^5_{pqrs} + \mathbf{D}^{SL}_{\mathbf{T}rs} \mathbf{X}^{J(pqrs)}_{pqrs} \right)$$

$$\mathbf{P}^{SL}_{pq} = \sum_{\mu=1}^{nsym} \sum_{r=1} \sum_{s=1} \left(\mathbf{D}^{SL}_{\mathbf{T}rs} \mathbf{X}^7_{pqrs} + \mathbf{D}^{LS}_{\mathbf{T}rs} \mathbf{X}^{J(pqrs)}_{pqrs} \right)$$

where $J(pqrs)$ is an integer function that selects specific definitions of the two-electron integrals from Equation 2.8. How exactly this works is shown in the following sections.

The form of the $\mathbf{Q}$ super matrix is almost the same a $\mathbf{P}$, but there will be and additional multiplication of the vector coupling coefficient between the open shells for each summation.

2.4.2.2 FORMPQ Algorithm

Forming the P and Q matrices proved to be the most difficult part of this program to parallelize. Its difficulty was due to the second term in the summations where each $\mathbf{X}$ matrix used depends on how far the summation has progressed. Further, each element of

P will have the **X** matrix change at different times. These two factors result in many **if** statements which GPUs have difficulty handling in parallel. There is also the issue where each element in the LL and SS sections need to loop over every element of the same sections of the density matrix, as do the LS and SL sections. This means that there will be lots of global memory reads unless handled properly. All together, this makes for a very ugly algorithm to parallelize.

Two different attempts were made to try and accelerate this algorithm. Each tried to exploit a different property of GPUs. One attempted to combat the amount of floating point operations to perform by distributing the amount of work to as many threads as possible. This will be referred to as the Multiple Threads Single Element (MTSE) algorithm. The second one tried to deal with the amount of global memory reads by making the most of each read. This will be referred to as Single Thread Single Element (STSE) algorithm. Each will be described in the sections below.

2.4.2.3   MTSE Algorithm

As previously stated, the purpose of the MTSE algorithm is to distribute the amount of work needed to be done to as many threads as possible. This was achieved by having multiple threads each computing two of the multiplications needed for a single element of two matrices in Equation 2.9, and then combining the products of these multiplications in parallel. These two tasks were performed by two separate kernels. Pseudocode showing how each calculates the **P** matrix is shown in Algorithms 6 and 7 respectively.

inttype refers to one or more if the **P** or **Q** matrices. In Algorithm 6, *inttype* can have a value of 0 or 1, where 0 refers to the LL and SS **P** or **Q** matrices, and 1 is the SL and LS matrices. In Algorithm 7, *inttype* can have a values ranging from 0 to 7 where 0 to 3 are the **P** LL, SS, SL, LS matrices, and the rest are the **Q** matrices in the same order. d is **D** in its vector representation and *pos* are the locations of the p and q elements within it. *locmx* is an array with 7 elements, and each element is equal to the number of unique LL, SS, SL, LS elements of symmetry $l-1$. *fctr* accounts for when an element is single or double counted in the summation. $x(\chi,:)$ is the vector representation of $\mathbf{X}^\chi$, *loc* is the location of needed *pqrs* element in the vector, *total* is the length of one of the dimensions of $\mathbf{X}^\chi$, and $k1$ and $k2$ are the outputs of the J function from Equation 2.9. Finally, *pXY* variables are matrices dimension $total \times total$ and store the sum of the two multiplications of the d and x vectors, *pmx* if the vector representation of the **P** matrices, and *s_tile* is a vector in the shared memory for summing up the elements of each row of the *pXY* matrices.

The grid will need to have $\frac{total-1}{blockDim.x} \times \frac{total-1}{blockDim.y} \times 2$ dimensions for Algorithm 6 and $\frac{total-1}{blockDim.x} \times \frac{total-1}{blockDim.y} \times 8$ for Algorithm 7. Algorithm 6 is fairly straightforward, but Algorithm 7 could use a little more explanation. It is a "parallel reduction" algorithm. At the start, each

Algorithm 6 Kernel 1 for MTSE

$i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$
$j = blockIdx.y$
$inttype = blockIdx.z - 1$
if $i \leq total$ **then**
  $l = lpq\_array(i, 1)$
  $p = lpq\_array(i, 2)$
  $q = lpq\_array(i, 3)$
  $fctr = 1.0$
  if $(p \neq q)$ **and** $(inttype == 0)$ **then**
    $fctr = 2.0$
  else if $(p == q)$ **and** $(inttype == 1)$ **then**
    $fctr = 0.5$
  end if
  $loc = \mathbf{max}(i, j) + ((\mathbf{max}(i, j) - 1) * (\mathbf{max}(i, j) - 2)/2) + \mathbf{min}(i, j) - 1$
  if $(inttype == 0)$ **then**
    if $i > j$ **then**
      $k1 = 3$
      $k2 = 2$
    else
      $k1 = 2$
      $k2 = 3$
    end if
    $pos1 = locmx(l) + ((p - 1)^2 + p - 1)/2 + q$
    $pos2 = locmx(l) + ((p - 1 + nbs(l))^2 + p + nbs(l) - 1)/2 + q + nbs(l)$
    $pLL(i, j) = d(pos1) * fctr * x(1, loc) + d(pos2) * fctr * x(k1, loc)$
    $pSS(i, j) = d(pos1) * fctr * x(k2, loc) + d(pos2) * fctr * x(k8, loc)$
  else
    if $i > j$ **then**
      $k1 = 4$
      $k2 = 6$
    else
      $k1 = 6$
      $k2 = 4$
    end if
    $pos1 = locmx(l) + ((p - 1 + nbs(l))^2 + p - 1 + nbs(l))/2 + q$
    $pos2 = locmx(l) + ((q - 1 + nbs(l))^2 + q - 1 + nbs(l))/2 + p$
    $pSL(i, j) = d(pos1) * fctr * x(7, loc) + d(pos2) * fctr * x(k1, loc)$
    $pLS(i, j) = d(pos1) * fctr * x(k2, loc) + d(pos2) * fctr * x(5, loc)$
  end if
end if

Algorithm 7 Kernel 2 for MTSE

$i = threadIdx.x + (blockIdx.x - 1) * blockDim.x * 2$
$inttype = blockIdx.z - 1$
if $inttype == 0$ **then**
  if $i \leq total$ **then**
    $s\_tile(threadIdx.x) = pLL(i, blockIdx.y)$
  else
    $s\_tile(threadIdx.x) = 0.0$
  end if
  if $i + blockDim.x \leq total$ **then**
    $s\_tile(threadIdx.x + blockDim.x) = pLL(i + blockDim.x, blockIdx.y)$
  else
    $s\_tile(threadIdx.x + blockDim.x) = 0.0$
  end if
else
  Perform a similar instruction on the other pXY or qXY matrices depending on $inttype$
end if
for $j = \log_2(blockDim.x), j \geq 0, j = j - 1$ **do**
  call **syncthreads**
  if $threadIdx.x \leq 2^j$ **then**
    $s\_tile(threadIdx.x) = s\_tile(threadIdx.x) + s\_tile(threadIdx.x + 2^j)$
  end if
end for
if $threadIdx.x == 1$ **then**
  $l = lpq\_array(blockIdx.y, 1)$
  $p = lpq\_array(blockIdx.y, 2)$
  $q = lpq\_array(blockIdx.y, 3)$
  if $(inttype == 0)$ **then**
    $pos = locmx(l) + ((p - 1)^2 + p - 1)/2 + q$
    $istat = $ **atomicadd**$(pmx(pos), s\_tile(threadIdx.x))$
  else
    Perform a similar instruction on the other pXY or qXY matrices depending on $inttype$
  end if
end if

---

thread loads an element of the relevant *pXY* array into the shared variable *s_tile* and also loads the element that is *blockDim.x* elements ahead of the original element. Then, in a loop starting from $j = \log_2(blockDim.x)$ and descending to $j = 0$ threads with a *threadIdx.x* value of less then 2^j add the value of *s_tile(threadIdx.x + 2^j)* to *s_tile(threadIdx.x)*. The key feature is that the amount of elements that need to be added is reduced by half every loop. Because it is possible that the total number of elements of *pXY* needed to sum up for the relevant *pmx* element could be larger than one block can handle, atomic operations are used to force the final addition to occur serially.

2.4.2.4 STSE Algorithm

The STSE algorithm tries a different approach to this problem than the MTSE algorithm. Instead of trying to spread out the amount of work to do as much as possible, it tries to reduce the amount of global memory reads by making each thread do as much work as it can. This results in fewer blocks, but each block needs to run longer. Giving only the calculations for *pmx* and when *inttype* = 0, Algorithm 8 shows how this is done.

All variables are the same as in Algorithms 6 and 7, but *inttype* can only be 0 or 1 (the meanings are the same as in Algorithm 6). As for the new variables, *s_d* is a chunk of *d* that has been loaded into the shared memory. *mink* and *maxk* are used to calculated the sections of *x* that current chunk of *d* in shared memory applies to and *loop* is used to adjust the value of *loc* if the needed two-electron integral is on the bottom triangle of a **X** matrix. *pm1* and *pm2* are accumulator variables that track the sum of an element of *pmx* as *d* is looped through.

Algorithm 8 has two nested loops. The outer one loops over elements of *d*, loading chunks of it into the shared memory as it goes. The inner one loops over elements of *x* and performs the multiplications of *d* and *x* elements, keeping a running tally of the sum of these multiplications. Something that is not shown in the pseudocode is checking if *l* and *m* have open shells. If both of them do, then *qm1* and *qm2* will be calculated as well. If only one or neither of them do, then this calculation is skipped. Because most of the shells will be closed, this allows for many calculations to be skipped. Once the inner loop has finished, the next section of *d* is loaded into the shared memory and then the multiplication of *x* and *s_d* continues. This process goes on until all elements of a row of **X** have been used.

2.5 Discussion

In this section, the results of GPU resource usage and profiling the various algorithms discussed previously will be presented. For profiling, the radon atom was chosen. Radon was selected because it contains many spinor orbitals and should be a good test of the ***formpq***

Algorithm 8 STSE algorithm

$i = threadIdx.x + (blockIdx.x - 1) * blockDim.x$

$inttype = blockIdx.z - 1$

if $i \leq total$ **then**

  $l = lpq\_array(i, 1)$

  $p = lpq\_array(i, 2)$

  $q = lpq\_array(i, 3)$

  $fctr = 1.0$

  $loop = 0$

  for $j = 1, j \leq (total - 1)/blockDim.x + 1, j = j + 1$ **do**

    $mink = 1 + (j - 1) * blockDim.x$

    $maxk = \mathbf{min}(j * blockDim.x, total)$

    if $threadIdx.x + (j - 1) * blockDim.x \leq total$ **then**

      $m = lpq\_array(threadIdx.x + (j - 1) * blockDim.x, 1)$

      $r = lpq\_array(threadIdx.x + (j - 1) * blockDim.x, 2)$

      $s = lpq\_array(threadIdx.x + (j - 1) * blockDim.x, 3)$

      if $(r \neq s)$ **and** $(inttype == 0)$ **then**

        $fctr = 2.0$

      end if

      $s\_d(threadIdx.x, 1) = fctr * d(locmx(m) + ((r - 1)^2 + r - 1)/2 + s - 1)$

      $s\_d(threadIdx.x, 2) = fctr * d(locmx(m) + ((r - 1 + nbs(m))^2 + r + nbs(m) - 1)/2 + s + nbs(m))$

    end if

    $fctr = 1.0$

    call **syncthreads**

    for $k = mink, k \leq maxk, k = k + 1$ **do**

      if $(i \geq k)$ **and** $(inttype == 0)$ **then**

        $loc = ((i - 1)^2 + i - 1)/2 + k$

        $k1 = 3$

        $k2 = 2$

      else if $(i < k)$ **and** $(inttype == 0)$ **then**

        $loop = loop + k - 1$

        $loc = (i^2 + i)/2 + loop$

        $k1 = 2$

        $k2 = 3$

      end if

      $pm1 = pm1 + s\_d(1 + k - mink, 1) * x(1, loc) + s\_d(1 + k - mink, 2) * x(k1, loc)$

      $pm2 = pm2 + s\_d(1 + k - mink, 1) * x(k2, loc) + s\_d(1 + k - mink, 2) * x(k8, loc)$

    end for

    call **syncthreads**

  end for

  $pmx(locmx(l) + ((p - 1)^2 + p - 1)/2 + q) = pm1$

  $pmx(locmx(l) + ((p - 1 + nbs(l))^2 + p + nbs(l) - 1)/2 + q + nbs(l)) = pm2$

end if

45

and *eint2* kernels on large atomic systems. Here, we only care about how long a calculation will take, not how accurate it is. Therefore, basis sets use the arbitrarily chosen wtbs parameters of $\alpha = 6.87 \times 10^{-2}$, $\beta = 1.83$, $\delta = 3.97$, and $\gamma = 0.78$. Additionally, due to the spherical symmetry of an atom exploited by cudaDFRATOM, the growth of a calculation differs depending on how the basis functions are distributed among the spinor symmetries. Therefore, results are reported in terms of the number of two-electron integrals instead of number of basis functions as is typically done.

2.5.1 Resource Usage

In the case of the work in this Thesis, the resource bottleneck for my kernels was almost always the number of registers used per thread. 63 32-bit registers provide only 252 bytes to work with, which when using double precision 8-byte numbers is hardly anything at all. A summary of the block sizes and resource usage of all kernels is given in Table 2.1.

Table 2.1: *Kernel resource usage*

Kernel	Block Size	Registers	Shared Memory (bytes)
eint1	(512, 1, 1)	63	45056
eint2	(512, 1, 1)	63	45088
vec2matrix	(512, 1, 1)	17	0
formd	(512, 1, 1)	33	28672
binary_search	(512, 1, 1)	30	60
convd	(512, 1, 1)	25	0
MTSE	(512, 1, 1)	46	32768
STSE	(512, 1, 1)	56	39388
forme	(512, 1, 1)	39	4152
formf	(512, 1, 1)	18	0
xtrpf	(512, 1, 1)	12	0
formg	(32, 16, 1)	55	12288
swapcol	(32, 16, 1)	10	0

Of these kernels, *eint2* and the *formpq_alg* kernels take up the bulk of the time needed in a typical calculation. This is in part due to the exponential growth in the number of two-electron integrals, but it is also caused by the amount of register spilling that each kernel will have. As stated in section 2.2.2 when the amount of registers a thread needs exceeds the amount that is available, it will begin to store the data in the global memory. As global memory is the slowest available memory type, this is very undesirable. To counteract this, many variables that would otherwise be better stored in register memory were moved into the shared memory. But this is also undesirable because it means that there will be less shared

memory available for storing data that needs to be read frequently from global memory. This in turn leads to more frequent reads from the global memory. Solving this balancing act required many iterations of testing, moving variables around, compiling, and retesting. Eventually, I arrived at the configurations shown above. Note how each kernel uses a block containing 512 threads. This number was chosen because it is the largest possible block size that can run when the required number of register per thread exceeds 32. 512 is also a good block size because in instances where a kernel is not limited by register or shared memory usage, it allows for the maximum number of threads per SM (1536) to be run.

2.5.2 Two-Electron Integral performance

Table 2.2 and Figure 2.9 show the results of profiling the **eint2** kernel against its CPU counterpart.

Table 2.2: *eint2 kernel performance. Times are in ms.*

Two-Electron Integrals[a]	CPU Time	*eint2*	*binary_search*	Total GPU time	Speedup[b]	Speedup[c]
74305	59	4.68	0.18	4.86	12.81	12.32
353220	320	18.05	0.71	18.77	17.72	17.04
1081185	980	52.51	2.21	54.72	18.66	17.90
2588950	2360	118.20	5.36	123.56	19.96	19.09
5299140	4910	237.96	11.07	249.03	20.63	19.71
9726255	8880	437.88	20.81	458.69	20.27	19.35
16476670	15140	730.24	35.96	766.20	20.73	19.75
26248635	24180	1159.98	57.99	1217.97	20.84	19.85

[a] For only the upper triangle of one of the eight **X** matrices.
[b] Speedup comparing just the time of **eint2** to the CPU time.
[c] Speedup comparing the time of **eint2** and **binary_search** to the CPU time.

As can be seen, the GPU calculations outperform the CPU calculations with an average speedup factor of over 19. For calculations with fewer number of integrals to compute the speedup is not as dramatic. This is most likely due to the overhead involved in just starting a GPU kernel. Table 2.2 shows two different speedups: one for just calculation of the integrals, and one that shows the speedup if the time needed for **binary_search** is included. The time needed to evaluate indices does not have much of an impact on the total performance on the total calculation time which proves this was a good way to generate the *lmpqrs* indices. Additionally, because cudaDFRATOM is a basis set optimizer, **binary_search** will need to
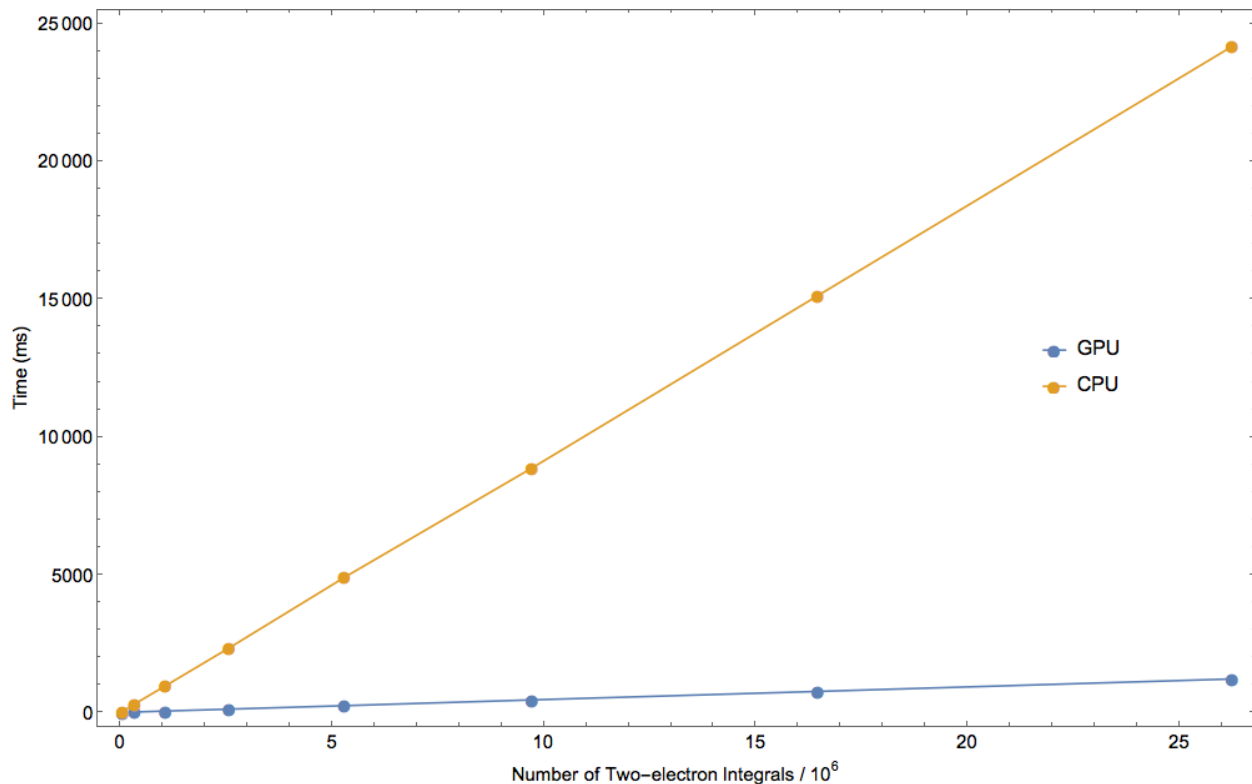
Figure 2.9: *Profiling of two-electron Integral calculations. The results of profiling the calculation of growing amounts of two-electron integrals. The blue line shows the time needed for GPU calculations and the yellow line shows time needed for CPU calculations. The time for the GPU calculations is the combined time of the **eint2** and **binary_search** kernels.*

be called only once, whereas ***eint2*** may need to be called many hundreds of times. Therefore, for an optimization calculation (which the original program is incapable of doing) the total speedup will limit to the just the integral calculations.

2.5.3   P and Q Matrix Formation Performance

As explained in section 2.4.2.1, there were two different attempts to speedup the formation of **P** and **Q**. The results of these attempts are shown in Tables 2.3 and 2.4 and also in Figure 2.10.

As can be seen, the attempt at ***MTSE*** was a complete failure and it isn't even faster than the CPU calculations. ***STSE*** was much better, but only just. It achieves little more than a two-fold speedup at best.

There are several reasons this could be the case. The first due to the poor global memory access that reading x has. x is stored as a vector. This helps greatly with memory cost and integral calculation, but makes it difficult to retrieve it as if it were a matrix.There is the cost of calculating where the necessary integral is, and then the cost of the very uncoalesced

Table 2.3: *MTSE kernel profiling results. Times are in ms.*

Two-Electron Integrals	Time per Iteration CPU	Time per Iteration GPU	Speedup
74305	1.90	6.06	0.31
353220	5.78	14.25	0.40
1081185	18.42	27.81	0.66
2588950	39.47	53.95	0.73
5299140	81.05	96.13	0.84
9726255	152.50	173.64	0.87
16476670	266.53	291.79	0.91
26248635	426.25	459.38	0.92

Table 2.4: *STSE kernel profiling results. Times are in ms.*

Two-Electron Integrals	Time per Iteration CPU	Time per Iteration GPU	Speedup
74305	1.90	2.94	0.64
353220	5.78	7.56	0.76
1081185	18.42	15.34	1.20
2588950	39.47	26.32	1.49
5299140	81.05	39.45	2.05
9726255	152.50	88.78	1.71
16476670	266.53	135.64	1.96
26248635	426.25	193.99	2.19

reads. Calculating the integrals as a vector, and then copying them into a matrix afterwards could help with this, but the large amount of integrals needed for large calculations makes this impractical on current hardware. How much of an effect inefficient access of global memory has on a calculation does have a limit however which is most likely why the speedup sees an improvement as the calculations get larger. A second reason that the speedup is limited is caused by the occupancy that the kernels can achieve on the GPU. A block size of 512 means that there is a maximum number of 3 blocks running per SM at a time. But these kernels are both limited by the register usage to just running one block per SM at a time. This is also most likely why **MTSE** performs so poorly compared **STSE**. **MTSE** was designed to spread out the amount of work to as many threads as possible. This results in a lot of blocks to be run. But because the amount of blocks that can run at a time is so limited there is no chance to see an improvement. But in the case of **STSE**, there are a fewer number of blocks needing be be run, although each one needs to run for longer than those of **MTSE**. Overall **MTSE** comes out on top.
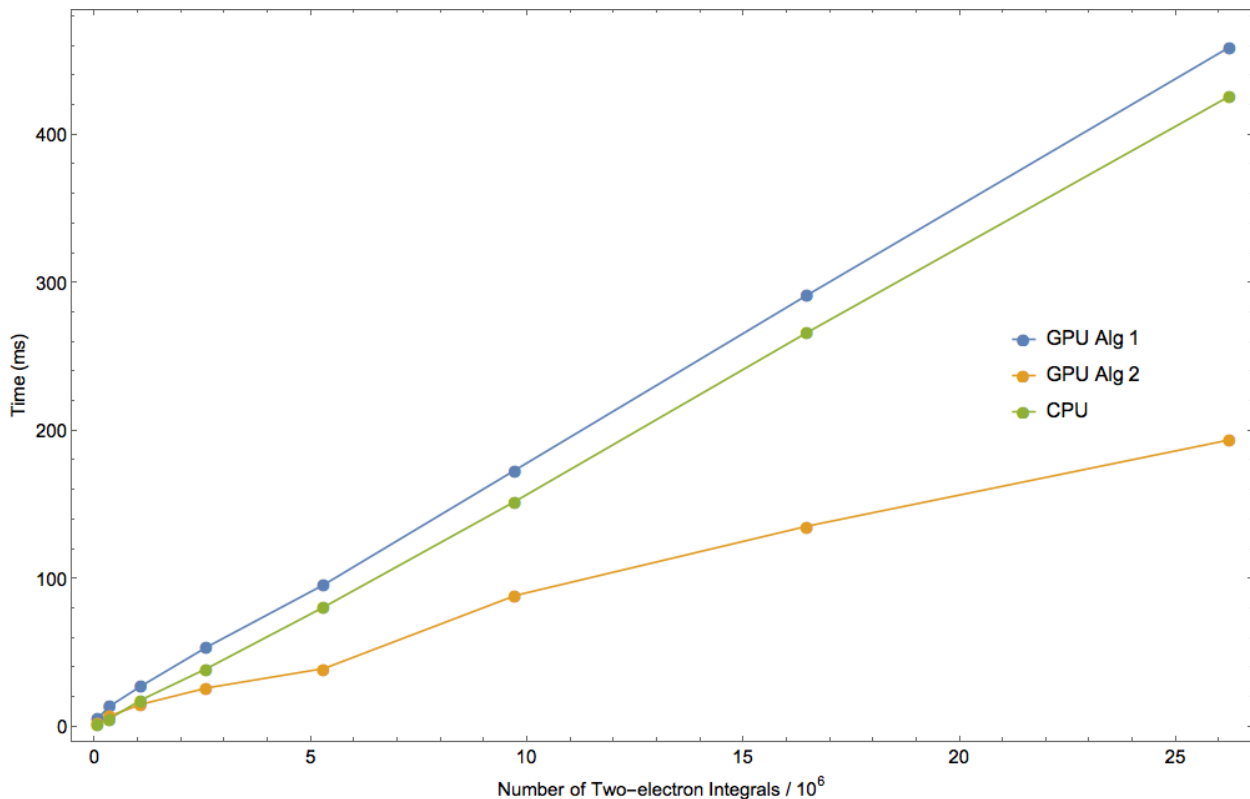
Figure 2.10: *Profiling of the **formpq** kernels. The results of profiling the formation of the **P** and **Q** matrices. The blue line shows the time needed for GPU calculations using the **MTSE** algorithm, the yellow line shows time needed for GPU calculations using **STSE**, and the green line shows the time needed for the CPU. Because the GPU calculations converge in fewer SCF iterations than the CPU calculations, only the average time for one iteration is shown.*

At present, the limitations of these algorithms are caused by the hardware they run on, not the efficiency of the algorithms themselves. If we imagine removing these limitations, we could consider which one will be better. I propose that **MTSE** will eventually surpass **STSE** for very large systems for the following reasons: the time it takes for a single block of **MTSE** to run is mostly governed by how well it can read the x vector. The effect of spreading out the elements has on memory bandwidth has a limit, once the elements are spread out enough, spreading them out further has no impact. Therefore there is a maximum amount of time it takes for a single block to run. The time for a single block of **STSE** on the other hand is controlled by both the efficiency of reading global memory *and* the amount of two-electron integrals it must read though. So there is no limit to how long a block can take. Therefore, I predict that **MTSE** should be faster as the number of integrals becomes very large.
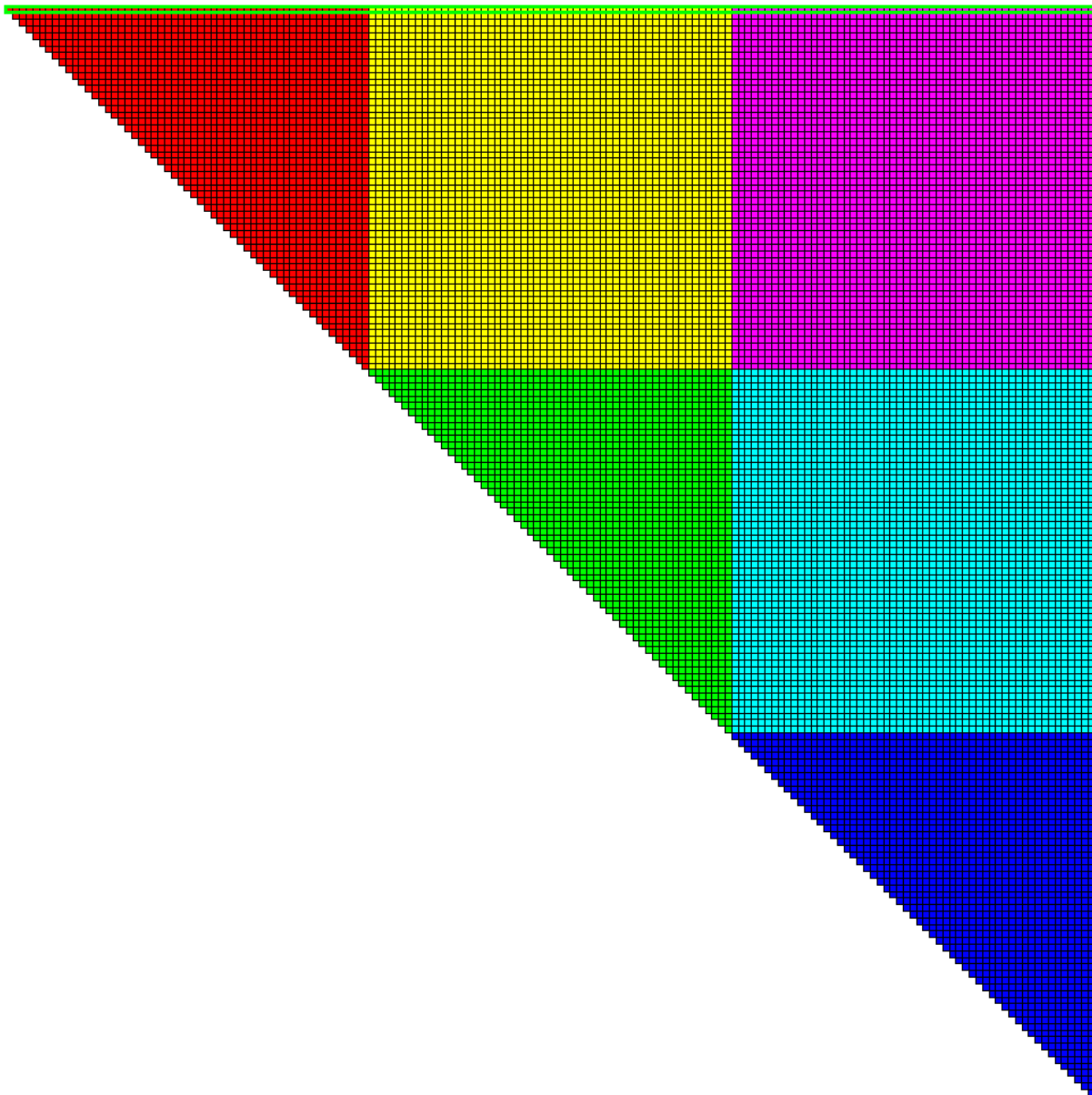
Figure 2.11: *Integrals needed for the first element of $\boldsymbol{P}$: The top triangle of the symmetric two-electron integral matrix in a matrix representation. The required integrals have been highlighted with a green border. The basis set used had $nsym = 3$ with $nbs(i) = (10, 10, 10)$. For clarity, the boxes have been coloured based on the spinor symmetries that they contain.*
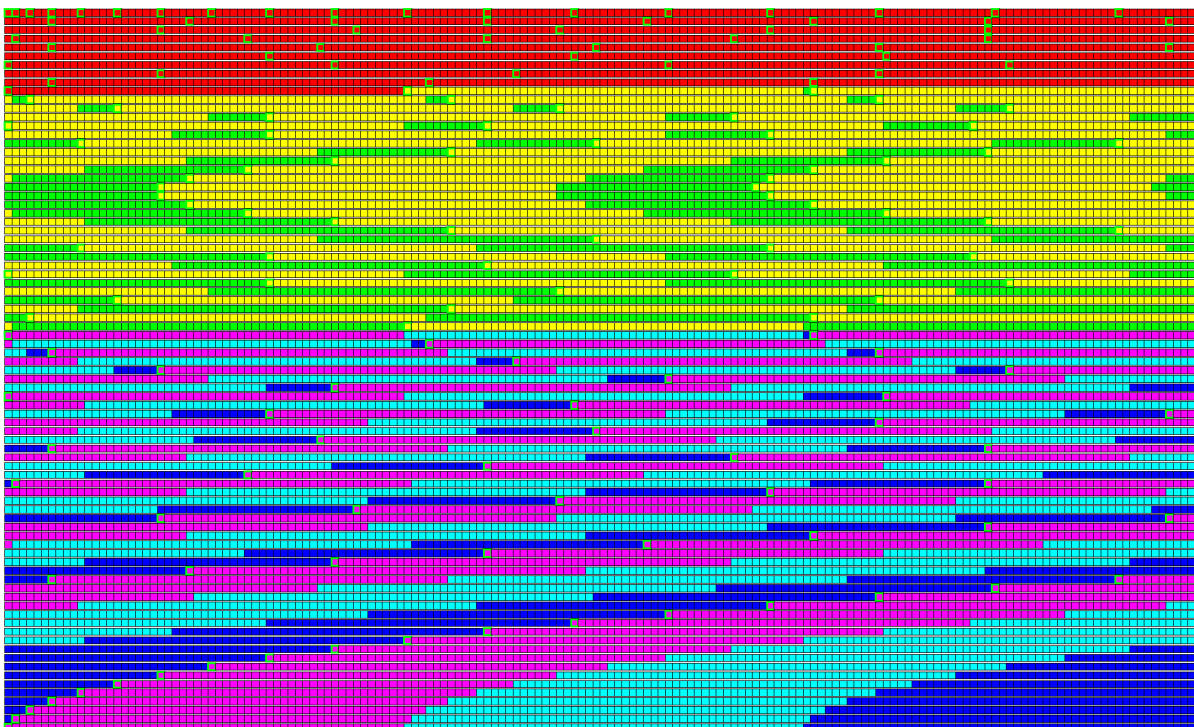
Figure 2.12: *Integrals needed for the first element of $\boldsymbol{P}$: The top triangle of the symmetric two-electron integral matrix in a vector representation. The required integrals have been highlighted with a green border. The spreading out of the elements means that reads to them can not be done effectively. The basis set used had nsym = 3 with nbs(i) = (10, 10, 10). For clarity, the boxes have been coloured in the same way as Figure 2.12.*

2.5.4 Total Speedup

There was some difficulty in determining the total speedup of a calculation. The reason why can be seen in Tables 2.5 and 2.6.

Table 2.5: *Total speedup of cudaDFRATOM using* **MTSE**. *The results for a total calculation are shown. All times are in s.*

Two-Electron Integrals	SCF Iterations CPU	Total CPU Time	SCF Iteractions GPU	Total GPU Time	Speedup
74305	21	0.12	21	0.84	0.14
353220	19	0.49	19	1.19	0.41
1081185	19	1.44	19	1.75	0.82
2588950	19	3.38	19	2.58	1.30
5299140	19	6.78	19	3.80	1.78
9726255	24	13.29	19	5.77	2.30
16476670	26	23.23	19	8.60	2.69
26248635	24	35.63	19	12.5	2.82

Table 2.6: *Total speedup of cudaDFRATOM using* **STSE**. *The results for a total calculation are shown. All times are in s.*

Two-Electron Integrals	SCF Iterations CPU	Total CPU Time	SCF Iteractions GPU	Total GPU Time	Speedup
74305	21	0.12	21	0.78	0.15
353220	19	0.49	19	1.07	0.46
1081185	19	1.44	19	1.51	0.95
2588950	19	3.38	19	2.08	1.62
5299140	19	6.78	19	2.72	2.49
9726255	24	13.29	19	4.14	3.20
16476670	26	23.23	19	5.57	4.16
26248635	24	35.63	19	7.47	4.76

As can be seen, the number of SCF iterations differs when comparing the GPU calculations to the CPU even though the convergence criteria was exactly the same. This is somewhat puzzling but I offer the following explanation. First, when writing the SCF routines, it became apparent that rewriting the linear algebra into CUDA would be rather pointless when I could just use the cuSOLVER and cuBLAS libraries and achieve the same result. As such, the SCF is completed using entirely different algorithms. Second, it is important to realize that a real number can only be *approximated* using double precision variables.
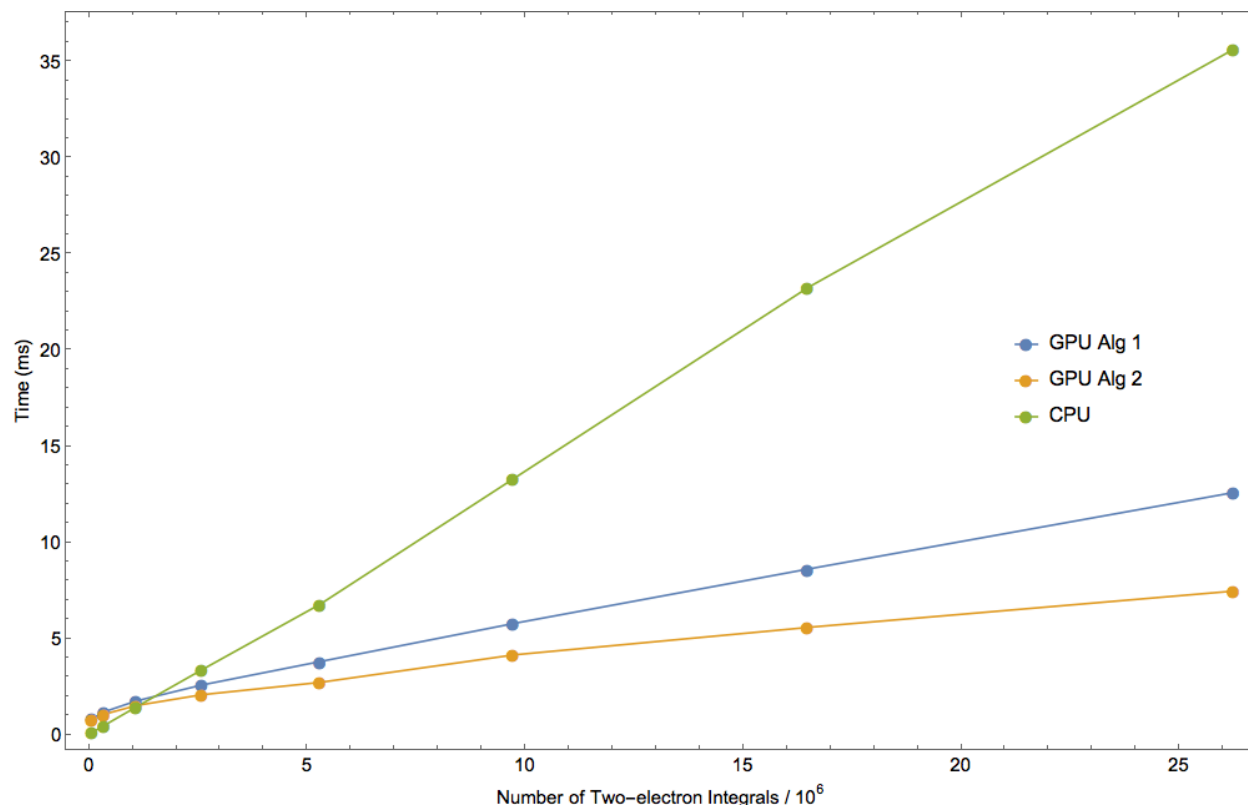
Figure 2.13: *Total time need to complete a calculation.*

This introduces numerical instability. For instance, assume that we have a computer that could only hold the first four digits of a number (leading zeros don't count). Then we take the number 1.000 and we want to add 0.0001000 to it 10000 times. The result of the first addition should be 1.0001, but because our computer can only hold four digits, the number is truncated to 1.000. The final result of this computation would not be 2.000 as expected but 1.000. During an SCF calculation, there are millions of numbers being multiplied together and added up, many of which may be close to, but not quite 0. This allows for lots of error to creep into a calculation just because of the limitations of the hardware. The fact that two different algorithms are being used on the GPU and CPU, and that they might not be handing numerical stability in the same way is what is most likely causing this to occur. It should also be pointed out that the final energy was not exactly the same from GPU to CPU either. Typically, the last 4 of 16 significant digits would be different. Because the rest are the same however, we still have chemical accuracy and I argue that cudaDFRATOMs results are therefore valid.

2.6 Input Description

cudaDFRATOM can be executed on Unix-like systems in the following way

```
$ path_to_executable input_file > output_file
```

The input file must end in ".inp" or an error will be given. Redirection of stdout to an output file is optional, but is recommended to save the results of a calculation. The input file is read using the namelist functionality of Fortran. A description of what must appear on each line of the input file is given below. Sample input files are also given in Appendix A.

1. A title of no more than 200 characters.

2. $contrl

jobtype			The type of calculation to be performed.
	=	'energy'	Will do a single point energy calculation.
	=	'bsopt'	Will optimize the basis set. Can only be used if bastype equals 'wtbs'.
c			The speed of light. If not given, the default is set to 137.03599976 au.

3. $nuc

znuc			The charge of the nucleus.
nucmdl			The nuclear model to use.
	=	1	Point nucleus (default).
	=	2	Finite sphere (not yet supported).
	=	3	Gaussian.
rnuc			The radius of the finite sphere nucleus.
alpha			The exponent for the gaussian nucleus. Defaults are given in litdata.f90.

4. $bas

nsym			The number of symmetries to be used.
bastype			The type of basis set given.
	=	'wtbs'	Use a WTBS basis set.
	=	'rdin'	Read in the basis set from the input file.
ngroup			The number of different groups to use in the WTBS scheme (default 1).

The next line will depend on the value of bastype. If bastype equals 'rdin' then the following lines must be the number of functions for the S+ symmetry, followed by the

exponents to use, each on a new line. The pattern repeats for each new symmetry. See the sample input files for further clarification. Otherwise, if bastype equals 'WTBS' the $wtbs group is read next.

5. $wtbs has to be given if bastype='wtbs'.

wtbspara	The α, β, δ, and γ WTBS parameters. If there is more than one group, the order would be α_1, β_1, δ_1, γ_1, α_2, β_2, δ_2, γ_2 and so on.
nbs	The number of functions used in each symmetry.
start	Where in the ζ pool each symmetry starts taking exponents from (default=1,1,1,1,1,1,1).
groups	What group each symmetry belongs to (default=1,1,1,1,1,1,1).

The next line depends on what the jobtype was set to. If jobtype equals 'energy' $newuoa is skipped and $econfig will be read next. If jobtype equals 'bsopt', then the $newuoa group will be needed.

6. $newuoa has to be given if jobtype='bsopt'. Refer to the newuoa documentation [27] for detailed information if needed.

rhobeg			The initial value of the trust region used by newuoa (default=0.1).
rhoend			The final value of the trust region used by newuoa. Must be smaller than rhobeg (default=1.0×10^{-4}).
iprint			The print level for newuoa.
	=	0	No printing from newuoa (default).
	=	1	Print only when newuoa has finished.
	=	2	Print only when the trust region has decreased by an order of magnitude.
	=	3	Print every iteration of newuoa.
maxfun			The maximum number of calls to calfun newuoa will make before terminating (default=500).

7. $econfig

nclose		The number of closed spinors for each symmetry.
nopen		The number of open spinors for each symmetry. There is a limit to one open orbital per symmetry.
freeel		The number of electrons available in the open spinors.
autogen		Automatically generates all possible combinations of spinor occupancies (default=.false.).
nconf		The number of configurations to be read in (needed if autogen is false).

If autogen is false, then the next nconf lines will be the spinor occupancies. They will be given as real numbers with one configuration per line.

8. $scf

maxitr			The maximum number of SCF iterations (default=50).
ixtrp			The method of extrapolation.
	=	0	No extrapolation (default).
	=	1	Extrapolate the Fock matrix.
dfctr			Damping factor for Fock maxtrix (default=0.3).
thdll			Convergence limit for the large-large components of the density matrix (default=1.0×10^{-5})
thdsl			Convergence limit for the small-large components of the density matrix (default=1.0×10^{-7})
thdss			Convergence limit for the small-small components of the density matrix (default=1.0×10^{-9})

Chapter 3

Basis Sets

3.1 Introduction

As stated in section 1.2.2, the use of Gaussian-type functions is the favored way of constructing basis sets for the use in molecular calculations. The use of basis sets introduces a new source of error, as we are now only approximating a wavefunction instead of directly numerically calculating it. One of the ways this error can manifest itself is in basis set superposition error (BSSE). BSSE appears when comparing the differences in energy between two different systems, as one might do to compare the difference in energy between reactants and product in a chemical reaction. Because we can not use a basis set with an infinite amount of basis functions, the basis set for the products and reactants in this example will be different (each will potentially have a different number of basis function centers, locations, etc.). This means that we are not comparing calculations with exactly the same level of accuracy, and the results may be invalidated because of this.

There is also the possibility of approximate linear dependencies in the basis set, which many show up during the SCF process. This happens when the overlap between any two different functions is close to unity (at least as far as a double precision number is concerned) and will mean that there will be eigenvalues that are equal to zero when diagonalizing the overlap matrix. We cannot find the inverse of such a matrix, so therefore the functions leading to these eigenvalues of zero must get transformed out of the matrix. This is kind of bandaid solution though, as the resulting Fock matrix after this transformation could produce errors in the energy differences on the order of chemical accuracy. It can also lead to a very aggravating process during a geometry optimization.

For example, say we have a molecular system in geometry state A. In this state, there are no approximate linear dependencies, and the geometry optimizer determines that the system would prefer to be in geometry state B. It updates the geometry, but in state B, some atoms might be too close together and now there are eigenfunctions that need to be rejected. The

energy calculated for geometry state B is inaccurate, which causes the geometry optimizer to think that it should place the atoms back into state A. But now, everything is exactly the same as it was, and the geometry optimizer, with no memory at all of what just happened, thinks that the perfect geometry is state B again. There is no way out of this loop, and we will not be able to find a stationary state, and all because of a poor choice in basis set. It is therefore important for a computational chemist to be aware of these issues so they can identify them when they occur, but even more so for someone who wants to make a new basis set, so that these issues can be avoided in the first place.

Designing a basis set that reduces the effects of BSSE is relatively straightforward, we can simply brute force the problem by adding more and more functions until they begin to cancel each other out. Designing one that avoids the issue of approximate linear dependencies however is more tricky. It is simply not possible to test every possible combination of atoms in every possible geometry to determine if a basis set has these problems. But there are some strategies we can use to avoid this problem.

One strategy that has shown to be particularly effective is using an exponent generating function to ensure that the exponents of the basis set are sufficiently different. There are many of these generating functions. One of these give what are called even-tempered basis sets (ETBS)[28]. The generating function for an ETBS is given below

$$\zeta_{l,k} = \alpha_l \beta_l^{k-1}, \quad \alpha_l, \beta_l > 0, \quad \beta_l \neq 1, \quad k = 1, 2, \ldots \tag{3.1}$$

where $\zeta_{l,k}$ is the kth exponent of basis function for orbital symmetry l, α_l is the exponent for the first, most diffuse, basis function, and β_l is a growth parameter that determines the gap between $\zeta_{l,k}$ and $\zeta_{l,k+1}$. This greatly reduces the difficulty of the problem, because instead of needing to optimize all the exponents of our basis set, we only need to optimize the α and β parameters of each symmetry. We can further reduce the number of parameters to optimize by using the same α and β parameters *for each symmetry*. This is what I did for the work in this chapter, so I will drop the l subscript. However, there is a problem with ETBS: the growth in the value of the exponents is not always sufficient. So we get diminishing returns by increasing the maximum value of k, and we have to add many more functions to improve the accuracy of the basis set by even a small amount. This is improved in the well-tempered basis set (WTBS)[29] generating function, which is shown below

$$\zeta_k = \zeta_{k-1}\beta \left[1 + \gamma \left(\frac{k}{n}\right)^\delta\right], \quad k = 2, 3, \ldots, n, \quad \zeta_1 = \alpha \tag{3.2}$$

where n is the maximum number of basis functions for the set, and α and β have the same restrictions on them as they did before. The parameters δ and γ have the effect of increasing the gap between ζs caused by β at high k. This means that the growth of the exponents
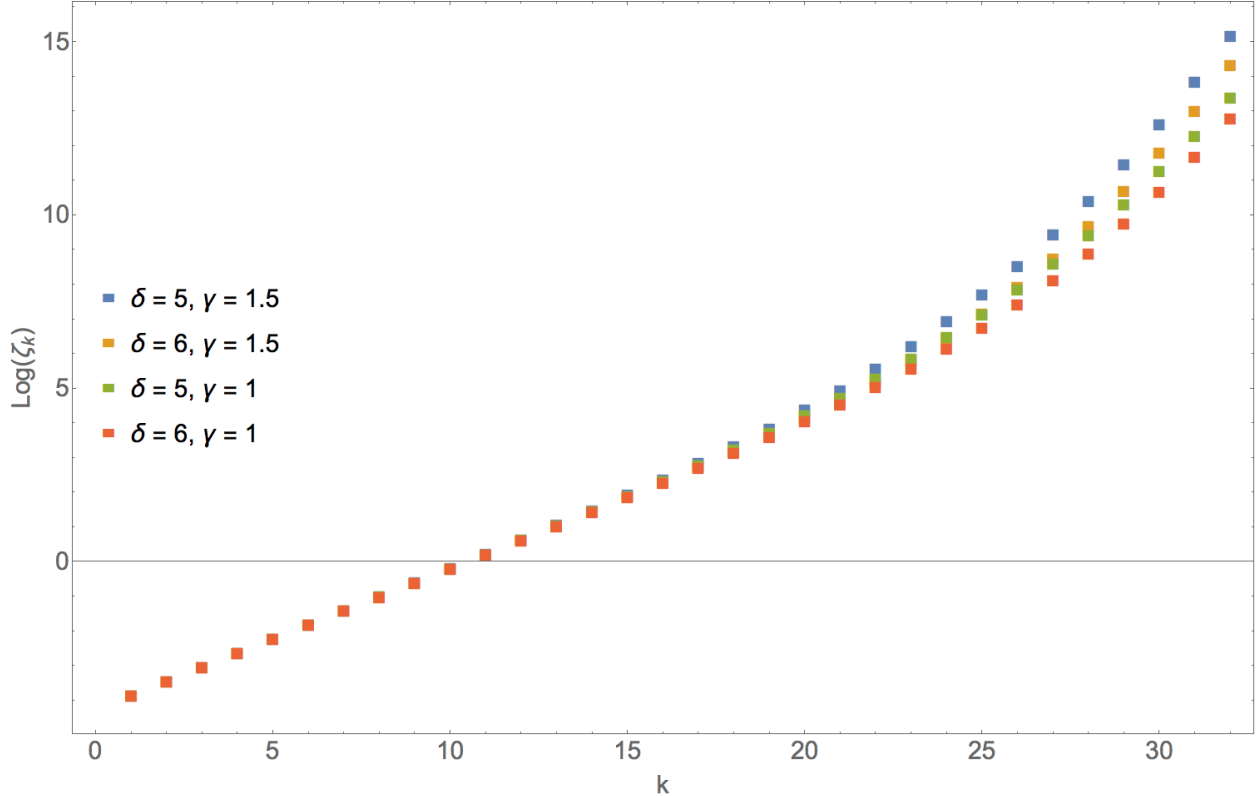
Figure 3.1: *Changes in ζ as the values of δ and γ are changed. The value of n was 36, α was 0.02, and β was 1.5*

is increased as they get larger, which allows them to make a more meaningful contribution to the total wavefunction. Thus, we get a more accurate basis set with far fewer functions. The effect that altering γ and δ has on the exponents can be seen in Figure 3.1.

It should be mentioned that each symmetry does not need to use the entire set of ζs that we optimize. Indeed, the most diffuse orbitals in the valence shells often don't use the most contracted functions at all, and conversely the most contracted orbitals in the core don't make much use of the most diffuse functions. Therefore, it is better to think of the set of ζs as pool, and that each symmetry is drawing a range of functions from that pool.

3.2 Methods

3.2.1 Basis set size optimization

We decided to use a somewhat novel method of generating our basis sets. We begin by choosing an arbitrarily large basis set size, such as s(1 : 40)p(1 : 40)d(1 : 40)f(1 : 40) (here $i(\# : \#)$ refers to the starting and ending index of ζ's used for symmetry i). Next, we then find the optimal α, β, δ, and γ WTBS parameters for this basis set. Finally, we use the

following steps to find the optimal basis set size.

Step 1. Begin by finding the fewest number of f functions necessary. This can be done by generating input files that range in size from s(1 : 40)p(1 : 40)d(1 : 40)f(1 : 1) to s(1 : 40)p(1 : 40)d(1 : 40)f(1 : 40).

Step 2. Optimize the basis sets and select the smallest basis set that is still below some minimum accuracy threshold. We chose a relative error to numerical calculations of no greater than 5.0×10^{-9} for non-relativistic sets and 5.0×10^{-8} for relativistic sets where relative error is defined as $\frac{|(\text{optimized energy}) - (\text{numerical energy})|}{\text{numerical enlet usergy}}$. The size of this basis set is s(1 : 40)p(1 : 40)d(1 : 40)f(1 : x_f).

Step 3. Replace the WTBS parameters with those from the newly optimized set and generate a list of input files that range in size from s(1 : 40)p(1 : 40)d(1 : x_f)f(1 : x_f) to s(1 : 40)p(1 : 40)d(1 : 40)f(1 : x_f).

Step 4. Optimize these new sets and select the smallest one that brings about an error that is still below the accuracy threshold. The size of this basis set is s(1 : 40)p(1 : 40)d(1 : x_d)f(1 : x_f).

Step 5. Repeat steps 3 and 4 for the remaining symmetries. The size of the basis set at the end of this step will be of size s(1 : x_s)p(1 : x_p)d(1 : x_d)f(1 : x_f).

Step 6. Replace the WTBS parameters with those from the newly optimized set and generate a list of input files that range in size from s(1 : x_s)p(1 : x_p)d(1 : x_d)f(1 : x_f) to s(1 : x_s)p(1 : x_p)d(1 : x_d)f(x_f : x_f).

Step 7. Optimize and select from the basis sets. The new set will be of size s(1 : x_s)p(1 : x_p)d(1 : x_d)f(y_f : x_f).

Step 8. Repeat steps 6 and 7 for the other symmetries except s. The final basis set will be of size s(1 : x_s)p(y_p : x_p)d(y_d : x_d)f(y_f : x_f).

This procedure is perhaps a bit clunky, so let us work through an example using Ne. Ne has both s and p type orbitals, and so will our basis set. We might initially guess that no more than 26 functions per symmetry will enough to get the accuracy we want, so we start with a basis set of size s(1:26)p(1:26). We then optimize the α, β, δ, and γ parameters for a basis set of this size. We make sure that the energy using a basis set of this size produces results that are accurate enough, and then we begin to reduce the size of the basis set for the p space. We carry over the WTBS parameters optimized in the last step, and generate input files corresponding to basis sets of size s(1:26)p(1:15) to s(1:26)p(1:26) (12 calculations in total) to determine the minimum number of tight functions we will need to accurately describe the p orbitals. After reoptimizing the WTBS parameters, we find that while s(1:26)p(1:15) is too small to be accurate enough, s(1:26)p(1:16) is still inside the threshold we want. So, we take the WTBS parameters from this calculation, and use them to generate new basis sets ranging in size from s(1:16)p(1:16) to s(1:26)p(1:16) (so we are

trying to eliminate contracted functions from the s orbitals). Here we find that s(1:23)p(1:16) is the smallest basis set we can use. Now, we want to eliminate diffuse functions. Carrying over the WTBS parameter from the last acceptable basis set, we generate basis sets of size s(1:23)p(1:16) to s(1:23)p(5:16). We find that trying to remove any more functions makes the basis set unacceptable so we keep the s(1:23)p(1:16) basis set. We could try to remove diffuse functions from the s symmetry, but experience teaches that the s space needs all the functions to be accurate, so we don't bother.

This process has a major drawback in that it requires a lot of computer power to run efficiently. But this is not really a problem if access to large computer clusters is available. Additionally, considering that calculations for the chain 1-8 can be run in parallel for different atoms, this optimization process is trivially parallel. The advantages of this process are twofold: it finds a very small basis set that is still accurate and it is completely automatable. If there are no problems with individual calculations not converging, the output files never even need to be manually (and intellectually) examined! We generated non-relativistic basis sets for elements 2-86 using the rWTBS program[29–34] and relativistic basis sets for elements 55-86 using the cudaDFRATOM program which is outlined in detail in Chapter 2.

3.2.2 Automating Optimization

Automating the optimization process was quite simple. I did this by writing two scripts: BS_sub_auto_rwtbs.sh and BS_find_best_rwtbs.sh. The code for BS_sub_auto.sh is given here

```bash
#!/bin/bash

sym=''
alljobs=false
bottom=false
path_to_exe='/some/path' # must be set!
wtime=01:00:00

if [[ -z $path_to_exe ]]; then
  echo "You must set the path to rwtbs.exe"
  exit
fi

while [ $# -gt 0 ]
do
  val=$1
  shift
  case $val in
    "-a")
      alljobs=true
      ;;
```

```bash
        "-b")
            bottom=true
        ;;
        "-S")
            sym='S'
            low=$1
            high=$2
            shift
            shift
         ;;
        "-P")
            sym='P'
            low=$1
            high=$2
            shift
            shift
         ;;
         "-D")
            sym='D'
            low=$1
            high=$2
            shift
            shift
         ;;
         "-F")
            sym='F'
            low=$1
            high=$2
            shift
            shift
         ;;
         "-t")
            wtime=$1
            shift
         ;;
         *)
            job=$val
         ;;
    esac
done


boilerplate(){
    cat << EOF
#!/bin/bash
#SBATCH --account=def-mariusz # change depending
#SBATCH --mem=2GB
```

```bash
#SBATCH --time=$1
#SBATCH --job-name=$2

echo ----------------------------------------
echo ----------------------------------------
echo began running on \'date\'
echo ----------------------------------------
echo ----------------------------------------
$3/rwtbs $2 >& $4
echo ----------------------------------------
echo ----------------------------------------
echo stopped running on \'date\'
echo ----------------------------------------
echo ----------------------------------------
EOF
}


if [ "$alljobs" == true ]; then
   job=`ls *inp`
elif [ -z "$job" ]; then
   echo no job given
   exit
fi

for file in $job; do
   dir=$(echo $file | sed -e "s/-.*//")
   if [[ -d "$dir" ]]; then
      echo "$dir already here"
   else
      mkdir $dir
   fi
   cd $dir

   # keep old out files and delete others
   rename "out" "out-old" *.out
   rm *inp
   rm *GUS
   rm *MCS

   cp ../$file $file"-orig"
   file_base=$(echo $file | sed -e 's/_.*//')

   # read the number of basis functions
   read -r -a old_bs <<< $(sed -n -e '/nbfsym/{s/.*nbfsym(1)=//; s/\$end//; p}'
     $file"-orig")
   # get the most contracted basis function
```

```
115    IFS='+' read -r -a old_top <<< $(sed -n -e '/ +/p' $file"-orig")

116

117    # keep track of how many files are made
118    total=0
119    for i in $(seq $low $high); do
120      for j in $(seq 0 3); do
121        new_bs[j]=${old_bs[j]}
122        new_top[j]=${old_top[j]}
123      done
124

125      # make alterations based on symmetry
126      case $sym in
127        "F")
128          ((new_bs[3]-=$i))
129          # make sure that not too many functions are cut
130          if [[ "${new_bs[3]}" == "0" ]]; then
131            break
132          fi
133          ((new_top[3]+=$i))
134          ;;
135        "D")
136          ((new_bs[2]-=$i))
137          if [[ "${new_bs[2]}" == "0" ]]; then
138            break
139          fi
140          ((new_top[2]+=$i))
141          ;;
142        "P")
143          ((new_bs[1]-=$i))
144          if [[ "${new_bs[1]}" == "0" ]]; then
145            break
146          fi
147          ((new_top[1]+=$i))
148          ;;
149        "S")
150          ((new_bs[0]-=$i))
151          if [[ "${new_bs[0]}" == "0" ]]; then
152            break
153          fi
154          ((new_top[0]+=$i))
155          ;;
156      esac
157

158      # make new calculation
159
       new_file=$file_base"_"${new_bs[0]}"s"${new_bs[1]}"p"${new_bs[2]}"d"${new_bs[3]}"f.inp"
160      cp $file"-orig" $new_file
```

```
161    sed -i "s/nalp=${old_bs[0]}/nalp=${new_bs[0]}/" $new_file
162    sed -i "s/${old_bs[0]} ${old_bs[1]} ${old_bs[2]} ${old_bs[3]}/${new_bs[0]}
       ${new_bs[1]} ${new_bs[2]} ${new_bs[3]}/" $new_file
163    if [ "$bottom" == false ]; then
164      sed -i -e "s/${old_top[0]}+/${new_top[0]} +/" -e
       "s/${old_top[1]}+/${new_top[1]} +/" -e "s/${old_top[2]}+/${new_top[2]} +/" -e
       "s/${old_top[3]} +/${new_top[3]} +/" $new_file
165    fi
166
167    # write submission script
168    boilerplate $wtime $new_file $path_to_exe ${new_file%???}out >
       ${new_file%???}.sh
169    #sbatch ${new_file%???}.sh # uncomment to automatically submit job
170    ((total++))
171  done
172  echo "$total files made"
173  cd ../
174 done
```

Listing 3.1: *BS_sub_auto_rwtbs.sh*

The first few lines handle the arguments given to the script. The arguments are flags which tell the script what to do. The flags can be given in any order, so long as each flag is followed by the appropriate options. The default flag is a file ending in ".inp" for the calculation that you would like to run. Optionally, this flag can be omitted, which will tell the script to run over all ".inp" fils in the current directory. I will refer to the input file to be used as a seed. The flags "-S", "-P", "-D", and "-F" tell the script which set of orbitals we would like to remove functions from ("-S" means s orbitals, "-P" means p orbitals, and so on). These flags _must_ be followed by by two integers, with the first being smaller than the second. The first number is the minimum number for functions to cut, and the second is the maximum number to cut. By default, cutting begins with the most contracted function. For example, using "-S 0 3" would take the seed file with a basis set of size s(1:20) and produce otherwise identical files with basis sets of size s(1:20), s(1:19), s(1:18), and s(1:17). If the number are negative, this can be used to add basis functions ("-S -1 0" on the same seed would produce basis sets of size s(1:21) and s(1:20)). Using the "-B" (for bottom) flag changes the default behavior of cutting contracted functions, to cutting diffuse functions. So for example, using the flags "-S 0 3 -B" on the seed would produce s(1:20), s(2:20), s(3:20), and s(4:20).

Once the flags have been parsed, the script makes a directory with the same name as the title of the seed, moves inside this directory, and generates the needed input files. It also creates a job submission script using the boilerplate function for each of new input files and submits them. Note that this part of the script will need to be changed depending on the

66

cluster it is being used on. If more than one seed is used, then the script will do the same thing for each seed. After the jobs are completed, the user runs BS_find_best_rwtbs.sh in the directory that contains all of the newly created directories. The code for BS_find_best_rwtbs.sh is given below

```bash
#!/bin/bash

tol=$1

if [ -z $tol ]; then
  tol='5*10^-9'
fi

echo The tolerance in relative error is $tol

mkdir best
for dir in `ls -d */`; do
  if [ "$dir" != 'old/' ]; then
    if [ "$dir" != 'best/' ]; then
      if [ "$dir" != 'tests/' ]; then
      cd $dir
      for file in *.out; do
        error=`grep -A 1 '[E(WT)-E(rf)]/E(rf)' $file | tail -1 | awk '{print $4}'`
        comperror=`echo $error | sed -e 's/D/\*10\^/'`
        testthress=$(bc -l <<< "scale=20; ($tol - $comperror)*10^10 >= 1")
        if [ $testthress -ge 1 ]; then
          cp `echo $file | sed -e 's/out/inp/'` ../best
          cp $file ../best
          echo Found best for $file
          break
        fi
      done
      fi
    fi
  fi
  cd ..
done
```

Listing 3.2: *BS_find_best_rwtbs.sh*

This script takes in one argument, the relative error we are looking for in a calculation. It enters each of the subdirectories made by BS_sub_auto.sh and scans the output files for the relative error compared to numerical calculations. It finds the file with the smallest basis set that is still under the error threshold, and copies the input and output files for that calculation into a new directory called "best". The user can then repeat this process for as many orbitals as they wish. There are also similar scripts for cudaDFRATOM which are

given in Appendix B.

3.3 Discussion

3.3.1 Optimized Non-Relativistic Basis Sets

The optimized non-relativistic WTBS parameters and basis set sizes for elements 2-86 are shown in Table 3.1. Relativistic WTBS parameters for elements 55-86 are shown in Table 3.2. Table 3.3 gives the energies calculated with these basis sets compared to their literature values.

Table 3.1: *Basis sets optimized using rWTBS. We did not have literature data for some elements. These elements are marked with a *. The size of the basis set for these elements was determined by using the size of the nearest elements.*

Element	Term Symbol	α	β	δ	γ	s	p	d	f
02 He	^1S	8.140×10^{-2}	1.953	4.504	1.515	(1:18)			
03 Li	^2S	1.596×10^{-2}	1.933	5.701	1.573	(1:22)			
04 Be	^1S	2.647×10^{-2}	1.938	5.841	1.594	(1:22)			
05 B	^2P	3.238×10^{-2}	1.948	5.573	1.523	(1:22)	(1:15)		
06 C	^3P	4.613×10^{-2}	1.941	4.717	1.317	(1:23)	(1:15)		
07 N	^4S	5.976×10^{-2}	1.923	5.183	1.442	(1:23)	(1:16)		
08 O	^3P	6.967×10^{-2}	1.936	5.170	1.426	(1:23)	(1:16)		
09 F	^2P	8.321×10^{-2}	1.942	5.084	1.408	(1:23)	(1:16)		
10 Ne	^1S	9.943×10^{-2}	1.945	4.988	1.392	(1:23)	(1:16)		
11 Na	^2S	1.696×10^{-2}	1.950	6.056	1.469	(1:26)	(3:19)		
12 Mg	^1S	2.375×10^{-2}	1.932	5.686	1.430	(1:26)	(3:19)		
13 Al	^2P	2.239×10^{-2}	1.898	5.555	1.413	(1:27)	(1:20)		
14 Si	^3P	3.352×10^{-2}	1.921	5.419	1.410	(1:26)	(1:19)		
15 P	^4S	4.410×10^{-2}	1.907	5.196	1.390	(1:26)	(1:19)		
16 S	^3P	5.004×10^{-2}	1.896	4.962	1.362	(1:26)	(1:19)		
17 Cl	^2P	5.831×10^{-2}	1.886	4.784	1.344	(1:26)	(1:19)		
18 Ar	^1S	6.834×10^{-2}	1.878	4.654	1.331	(1:26)	(1:19)		
19 K	^2S	1.392×10^{-2}	1.889	6.354	1.523	(1:28)	(3:22)		
20 Ca	^1S	1.837×10^{-2}	1.874	6.109	1.504	(1:28)	(3:22)		
21 Sc	^2D	2.040×10^{-2}	1.878	6.160	1.505	(1:28)	(3:22)	(2:16)	
22 Ti	^3F	2.218×10^{-2}	1.886	6.247	1.508	(1:28)	(3:22)	(2:16)	
23 V	^4F	2.361×10^{-2}	1.887	6.573	1.507	(1:29)	(3:23)	(2:16)	
24 Cr	^7S	2.345×10^{-2}	1.865	5.348	1.407	(1:29)	(3:22)	(2:17)	
25 Mn	^6S	2.572×10^{-2}	1.865	5.336	1.407	(1:29)	(3:22)	(2:17)	
26 Fe	^5D	2.621×10^{-2}	1.869	5.301	1.395	(1:29)	(3:22)	(3:17)	
27 Co	^4F	2.643×10^{-2}	1.871	5.905	1.413	(1:29)	(4:23)	(3:17)	
28 Ni	^3F	2.882×10^{-2}	1.875	5.935	1.413	(1:29)	(3:23)	(3:17)	

Element	Term Symbol	α	β	δ	γ	s	p	d	f
29 Cu	^2S	2.283×10^{-2}	1.857	5.485	1.413	(1:30)	(4:23)	(3:18)	
30 Zn	^1S	3.297×10^{-2}	1.881	5.817	1.339	(1:30)	(3:24)	(2:17)	
31 Ga	^2P	2.742×10^{-2}	1.863	5.703	1.462	(1:30)	(1:23)	(3:18)	
32 Ge	^3P	3.143×10^{-2}	1.848	5.290	1.386	(1:30)	(1:23)	(4:18)	
33 As	^4S	4.728×10^{-2}	1.869	5.325	1.340	(1:30)	(1:23)	(3:17)	
34 Se	^3P	5.087×10^{-2}	1.847	4.949	1.533	(1:30)	(1:22)	(3:18)	
35 Br	^2P	5.927×10^{-2}	1.861	5.192	1.333	(1:30)	(1:23)	(3:17)	
36 Kr	^1S	6.804×10^{-2}	1.859	5.510	1.370	(1:29)	(1:23)	(3:17)	
37 Rb	^2S	1.421×10^{-2}	1.856	7.181	1.626	(1:30)	(3:25)	(6:21)	
38 Sr	^1S	1.808×10^{-2}	1.846	6.674	1.553	(1:30)	(3:25)	(6:20)	
39 Y	^2D	1.997×10^{-2}	1.841	6.569	1.543	(1:30)	(3:25)	(2:20)	
40 Zr	^5F	2.184×10^{-2}	1.837	6.485	1.535	(1:30)	(3:25)	(2:20)	
41 Nb	^6D	2.340×10^{-2}	1.835	6.434	1.530	(1:30)	(3:25)	(2:20)	
42 Mo	^7S	2.537×10^{-2}	1.832	6.372	1.524	(1:30)	(3:25)	(2:20)	
43 Tc	^6S	2.597×10^{-2}	1.836	6.453	1.534	(1:30)	(3:25)	(2:20)	
44 Ru	^5F	2.682×10^{-2}	1.833	6.788	1.606	(1:30)	(3:25)	(2:21)	
45 Rh	^4F	2.751×10^{-2}	1.835	6.883	1.620	(1:30)	(3:25)	(2:21)	
46 Pd	^1S	5.944×10^{-2}	1.827	5.980	1.504	(1:29)	(2:24)	(1:19)	
47 Ag	^2S	2.438×10^{-2}	1.800	5.521	1.423	(1:31)	(4:25)	(3:21)	
48 Cd	^1S	2.911×10^{-2}	1.786	5.334	1.408	(1:31)	(4:25)	(3:21)	
49 In	^2P	2.714×10^{-2}	1.799	5.540	1.431	(1:31)	(1:25)	(3:21)	
50 Sn	^3P	2.884×10^{-2}	1.794	5.422	1.415	(1:31)	(1:25)	(4:21)	
51 Sb	^4S	4.481×10^{-2}	1.815	6.567	1.605	(1:30)	(1:25)	(3:21)	
52 Te	^3P	4.846×10^{-2}	1.817	6.247	1.528	(1:30)	(1:26)	(3:20)	
53 I	^2P	5.369×10^{-2}	1.810	6.016	1.505	(1:30)	(1:25)	(3:20)	
54 Xe	^1S	5.981×10^{-2}	1.802	5.862	1.490	(1:30)	(1:25)	(3:20)	
55 Cs	^2S	1.177×10^{-2}	1.792	6.485	1.510	(1:33)	(4:28)	(5:23)	
56 Ba	^1S	1.560×10^{-2}	1.770	6.152	1.518	(1:32)	(3:28)	(6:23)	
57 La	^2D	1.716×10^{-2}	1.764	6.067	1.513	(1:32)	(3:28)	(2:23)	(5:14)
58 Ce*	^3H	1.652×10^{-2}	1.774	6.254	1.529	(1:32)	(3:28)	(2:23)	(4:19)
59 Pr	^4I	1.696×10^{-2}	1.776	6.297	1.533	(1:32)	(3:28)	(6:23)	(4:19)
60 Nd	^5I	1.737×10^{-2}	1.778	6.563	1.579	(1:32)	(3:28)	(6:24)	(4:19)
61 Pm	^6H	1.784×10^{-2}	1.780	6.615	1.585	(1:32)	(3:28)	(6:24)	(4:19)
62 Sm	^7F	1.857×10^{-2}	1.778	5.842	1.464	(1:33)	(3:27)	(6:23)	(4:19)
63 Eu	^8S	1.876×10^{-2}	1.784	6.718	1.596	(1:32)	(3:28)	(6:24)	(4:19)
64 Gd*	^7F	1.926×10^{-2}	1.782	6.844	1.599	(1:33)	(3:28)	(6:24)	(4:19)
65 Tb*	^6H	1.971×10^{-2}	1.784	6.898	1.604	(1:33)	(3:28)	(6:24)	(4:19)
66 Dy	^5I	2.020×10^{-2}	1.788	6.543	1.540	(1:33)	(3:28)	(6:23)	(4:19)
67 Ho*	^4I	2.064×10^{-2}	1.785	7.090	1.640	(1:33)	(3:28)	(6:23)	(4:20)
68 Er	^3H	2.106×10^{-2}	1.792	7.029	1.638	(1:32)	(3:28)	(6:24)	(4:20)

Element	Term Symbol	α	β	δ	γ	s	p	d	f
69 Tm	^2F	2.152×10^{-2}	1.794	7.075	1.643	(1:32)	(3:28)	(6:24)	(4:20)
70 Yb*	^1S	2.200×10^{-2}	1.795	7.141	1.652	(1:32)	(3:28)	(2:24)	(4:20)
71 Lu	^2D	2.373×10^{-2}	1.790	7.018	1.640	(1:32)	(3:28)	(2:24)	(4:20)
72 Hf	^3F	2.511×10^{-2}	1.788	6.963	1.636	(1:32)	(3:28)	(2:24)	(5:20)
73 Ta	^4F	2.684×10^{-2}	1.784	6.907	1.633	(1:32)	(2:28)	(2:24)	(5:20)
74 W	^5D	2.814×10^{-2}	1.783	6.889	1.633	(1:32)	(3:28)	(2:24)	(5:20)
75 Re	^6S	2.931×10^{-2}	1.782	6.886	1.634	(1:32)	(3:28)	(2:24)	(5:20)
76 Os	^5D	3.075×10^{-2}	1.781	6.896	1.637	(1:32)	(2:29)	(2:24)	(4:20)
77 Ir	^4F	3.195×10^{-2}	1.780	6.867	1.635	(1:32)	(3:28)	(2:24)	(5:20)
78 Pt	^3D	3.173×10^{-2}	1.785	7.020	1.654	(1:32)	(3:28)	(2:24)	(5:20)
79 Au	^2S	3.190×10^{-2}	1.791	6.702	1.569	(1:33)	(4:28)	(2:23)	(5:19)
80 Hg	^1S	3.546×10^{-2}	1.781	6.691	1.604	(1:32)	(3:28)	(2:23)	(5:20)
81 Tl	^2P	3.241×10^{-2}	1.794	6.826	1.561	(1:34)	(1:29)	(3:23)	(6:20)
82 Pb	^3P	3.867×10^{-2}	1.778	6.948	1.656	(1:32)	(1:28)	(3:25)	(6:21)
83 Bi	^4S	4.514×10^{-2}	1.766	6.158	1.545	(1:32)	(1:27)	(2:24)	(6:19)
84 Po	^3P	4.783×10^{-2}	1.763	5.977	1.516	(1:32)	(1:27)	(3:23)	(6:19)
85 At	^2P	5.210×10^{-2}	1.757	5.846	1.502	(1:32)	(1:27)	(3:23)	(6:19)
86 Rn	^1S	5.716×10^{-2}	1.749	5.695	1.486	(1:32)	(1:27)	(3:23)	(6:19)

Table 3.2: *Basis sets optimized using cudaDFRATOM*

Element	Electron Configuration	α	β	δ	γ	s	p	d	f
55 Cs	[Xe] 6s^1	1.824×10^{-2}	1.921	6.085	1.062	(1:29)	(3:28)	(5:20)	
56 Ba	[Xe] 6s^2	2.033×10^{-2}	1.904	5.473	1.161	(1:30)	(3:28)	(6:20)	
57 La	[Ba] 4f^1	1.540×10^{-2}	1.871	5.166	2.036	(1:33)	(4:28)	(5:22)	(4:16)
58 Ce	[Ba] 4f^2	1.838×10^{-2}	1.857	4.969	1.450	(1:31)	(3:28)	(5:22)	(4:16)
59 Pr	[Ba] 4f^3	1.945×10^{-2}	1.857	4.947	1.438	(1:31)	(3:28)	(5:22)	(4:16)
60 Nd	[Ba] 4f^4	2.043×10^{-2}	1.859	4.928	1.422	(1:31)	(3:28)	(5:22)	(4:16)
61 Pm	[Ba] 4f^5	2.142×10^{-2}	1.860	4.906	1.405	(1:31)	(2:28)	(4:22)	(4:16)
62 Sm	[Ba] 4f^6	2.414×10^{-2}	1.856	4.854	1.379	(1:31)	(3:28)	(5:22)	(3:16)
63 Eu	[Ba] 4f^7	1.993×10^{-2}	1.877	5.099	1.020	(1:31)	(4:29)	(5:22)	(4:16)
64 Gd	[Ba] 4f^8	2.063×10^{-2}	1.880	5.064	0.996	(1:31)	(4:29)	(5:22)	(4:16)
65 Td	[Ba] 4f^9	2.396×10^{-2}	1.869	4.840	0.967	(1:31)	(3:29)	(5:22)	(4:16)
66 Dy	[Ba] 4f^{10}	2.177×10^{-2}	1.888	5.353	0.839	(1:30)	(4:29)	(5:22)	(4:16)
67 Ho	[Ba] 4f^{11}	2.517×10^{-2}	1.878	5.126	0.816	(1:30)	(3:29)	(5:22)	(4:16)
68 Er	[Ba] 4f^{12}	2.811×10^{-2}	1.875	5.145	1.045	(1:30)	(3:29)	(5:22)	(3:16)
69 Tm	[Ba] 4f^{13}	2.628×10^{-2}	1.887	5.480	0.886	(1:31)	(3:30)	(5:22)	(4:16)
70 Yb	[Ba] 4f^{14}	2.918×10^{-2}	1.883	5.490	0.873	(1:31)	(3:30)	(5:22)	(3:16)

Table 3.2: *(continued)*

Element	Electron Configuration	α	β	δ	γ	s	p	d	f
71 Lu	[Yb] $5d^1$	3.424×10^{-2}	1.873	5.124	1.353	(1:32)	(2:29)	(1:23)	(3:16)
72 Hf	[Yb] $5d^2$	3.982×10^{-2}	1.900	6.117	1.473	(1:32)	(2:29)	(1:22)	(4:16)
73 Ta	[Yb] $5d^3$	4.271×10^{-2}	1.899	6.126	1.450	(1:32)	(3:29)	(1:22)	(4:16)
74 W	[Yb] $5d^4$	4.564×10^{-2}	1.901	5.936	1.351	(1:32)	(3:29)	(1:21)	(4:16)
75 Re	[Yb] $5d^5$	4.827×10^{-2}	1.900	5.898	1.323	(1:32)	(3:29)	(1:21)	(4:16)
76 Os	[Yb] $5d^6$	5.081×10^{-2}	1.899	5.873	1.296	(1:32)	(3:29)	(1:21)	(4:16)
77 Ir	[Yb] $5d^7$	5.333×10^{-2}	1.898	5.846	1.274	(1:32)	(3:29)	(1:21)	(4:16)
78 Pt	[Yb] $5d^8$	5.391×10^{-2}	1.892	6.058	1.365	(1:32)	(3:29)	(1:23)	(5:16)
79 Au	[Yb] $5d^9$	5.684×10^{-2}	1.893	6.111	1.354	(1:32)	(3:29)	(1:23)	(5:16)
80 Hg	[Yb] $5d^{10}$	5.965×10^{-2}	1.893	6.156	1.342	(1:32)	(3:29)	(1:23)	(5:16)
81 Tl	[Hg] $6p^1$	4.071×10^{-2}	1.875	5.161	0.944	(1:30)	(1:29)	(1:22)	(5:17)
82 Pb	[Hg] $6p^2$	4.364×10^{-2}	1.869	5.015	0.925	(1:30)	(1:29)	(3:22)	(6:17)
83 Bi	[Hg] $6p^3$	5.109×10^{-2}	1.854	3.953	0.792	(1:33)	(1:30)	(2:21)	(5:17)
84 Po	[Hg] $6p^4$	5.439×10^{-2}	1.849	4.666	0.890	(1:30)	(1:29)	(3:22)	(6:17)
85 At	[Hg] $6p^5$	6.177×10^{-2}	1.892	10.223	1.500	(1:30)	(1:31)	(3:22)	(6:16)
86 Rn	[Hg] $6p^6$	6.671×10^{-2}	1.831	3.906	0.867	(1:31)	(1:29)	(3:21)	(6:17)

Table 3.3: *Energies produced from basis sets in Tables 3.1 and 3.2. Relativistic basis sets are labeled with ^{r}. All energies are in Hartree. Non-relativisic data comes from [35]. Relativistic data comes from [22].*

Element	Calculated Energy	Literature Energy	Absolute Error	Relative Error
02 He	-2.861	-2.861	$> 1 \times 10^{-6}$	3.6×10^{-9}
03 Li	-7.432	-7.432	$> 1 \times 10^{-6}$	2.9×10^{-9}
04 Be	-14.573	-14.573	$> 1 \times 10^{-6}$	3.2×10^{-9}
05 B	-24.529	-24.529	$> 1 \times 10^{-6}$	4.8×10^{-9}
06 C	-37.688	-37.688	$> 1 \times 10^{-6}$	4.1×10^{-9}
07 N	-54.400	-54.400	$> 1 \times 10^{-6}$	2.8×10^{-9}
08 O	-74.809	-74.809	$> 1 \times 10^{-6}$	3.4×10^{-9}
09 F	-99.409	-99.409	$> 1 \times 10^{-6}$	4.0×10^{-9}
10 Ne	-128.547	-128.547	1×10^{-6}	4.4×10^{-9}
11 Na	-161.858	-161.858	1×10^{-6}	4.4×10^{-9}
12 Mg	-199.614	-199.614	1×10^{-6}	3.9×10^{-9}
13 Al	-241.876	-241.876	1×10^{-6}	2.8×10^{-9}
14 Si	-288.854	-288.854	1×10^{-6}	4.6×10^{-9}
15 P	-340.718	-340.718	1×10^{-6}	4.3×10^{-9}
16 S	-397.504	-397.504	2×10^{-6}	4.5×10^{-9}
17 Cl	-459.482	-459.482	2×10^{-6}	4.6×10^{-9}
18 Ar	-526.817	-526.817	2×10^{-6}	4.5×10^{-9}

Table 3.3: *(continued)*

Element	Calculated Energy	Literature Energy	Absolute Error	Relative Error
19 K	-599.164	-599.164	3×10^{-6}	4.6×10^{-9}
20 Ca	-676.758	-676.758	3×10^{-6}	3.8×10^{-9}
21 Sc	-759.735	-759.735	3×10^{-6}	4.3×10^{-9}
22 Ti	-848.405	-848.405	4×10^{-6}	4.9×10^{-9}
23 V	-942.884	-942.884	4×10^{-6}	4.4×10^{-9}
24 Cr	-1043.356	-1043.356	4×10^{-6}	4.1×10^{-9}
25 Mn	-1149.866	-1149.866	6×10^{-6}	4.8×10^{-9}
26 Fe	-1262.443	-1262.443	6×10^{-6}	4.5×10^{-9}
27 Co	-1381.414	-1381.414	6×10^{-6}	4.6×10^{-9}
28 Ni	-1506.870	-1506.870	7×10^{-6}	4.8×10^{-9}
29 Cu	-1638.963	-1638.963	6×10^{-6}	3.4×10^{-9}
30 Zn	-1777.848	-1777.848	9×10^{-6}	4.8×10^{-9}
31 Ga	-1923.261	-1923.261	9×10^{-6}	4.5×10^{-9}
32 Ge	-2075.359	-2075.359	7×10^{-6}	3.3×10^{-9}
33 As	-2234.238	-2234.238	1×10^{-5}	4.8×10^{-9}
34 Se	-2399.867	-2399.867	1×10^{-5}	4.4×10^{-9}
35 Br	-2572.441	-2572.441	1×10^{-5}	4.8×10^{-9}
36 Kr	-2752.054	-2752.054	1×10^{-5}	4.7×10^{-9}
37 Rb	-2938.357	-2938.357	1×10^{-5}	4.7×10^{-9}
38 Sr	-3131.545	-3131.545	1×10^{-5}	4.4×10^{-9}
39 Y	-3331.684	-3331.684	1×10^{-5}	4.4×10^{-9}
40 Zr	-3538.995	-3538.995	1×10^{-5}	4.4×10^{-9}
41 Nb	-3753.597	-3753.597	1×10^{-5}	4.3×10^{-9}
42 Mo	-3975.549	-3975.549	1×10^{-5}	4.5×10^{-9}
43 Tc	-4204.788	-4204.788	2×10^{-5}	4.7×10^{-9}
44 Ru	-4441.539	-4441.539	2×10^{-5}	4.5×10^{-9}
45 Rh	-4685.881	-4685.881	2×10^{-5}	4.8×10^{-9}
46 Pd	-4937.921	-4937.921	2×10^{-5}	4.8×10^{-9}
47 Ag	-5197.698	-5197.698	2×10^{-5}	4.3×10^{-9}
48 Cd	-5465.133	-5465.133	2×10^{-5}	4.2×10^{-9}
49 In	-5740.169	-5740.169	2×10^{-5}	4.6×10^{-9}
50 Sn	-6022.931	-6022.931	2×10^{-5}	3.9×10^{-9}
51 Sb	-6313.485	-6313.485	2×10^{-5}	4.4×10^{-9}
52 Te	-6611.784	-6611.784	3×10^{-5}	4.8×10^{-9}
53 I	-6917.980	-6917.980	3×10^{-5}	4.6×10^{-9}
54 Xe	-7232.138	-7232.138	3×10^{-5}	4.3×10^{-9}
55 Cs	-7553.933	-7553.933	2×10^{-5}	3.6×10^{-9}
56 Ba	-7883.543	-7883.543	3×10^{-5}	4.9×10^{-9}
57 La	-8221.066	-8221.066	4×10^{-5}	4.8×10^{-9}
58 Ce	-8566.919	-	-	-
59 Pr	-8921.180	-8921.181	4×10^{-5}	4.9×10^{-9}

Table 3.3: *(continued)*

Element	Calculated Energy	Literature Energy	Absolute Error	Relative Error
60 Nd	-9283.882	-9283.882	4×10^{-5}	4.8×10^{-9}
61 Pm	-9655.098	-9655.098	4×10^{-5}	4.7×10^{-9}
62 Sm	-10034.952	-10034.952	4×10^{-5}	4.4×10^{-9}
63 Eu	-10423.543	-10423.543	4×10^{-5}	4.7×10^{-9}
64 Gd	-10820.617	-	-	-
65 Tb	-11226.568	-	-	-
66 Dy	-11641.452	-11641.452	5×10^{-5}	4.4×10^{-9}
67 Ho	-12065.289	-	-	-
68 Er	-12498.152	-12498.152	6×10^{-5}	4.8×10^{-9}
69 Tm	-12940.174	-12940.174	6×10^{-5}	4.8×10^{-9}
70 Yb	-13391.456	-	-	-
71 Lu	-13851.807	-13851.808	6×10^{-5}	4.7×10^{-9}
72 Hf	-14321.249	-14321.249	6×10^{-5}	4.8×10^{-9}
73 Ta	-14799.812	-14799.812	7×10^{-5}	4.9×10^{-9}
74 W	-15287.546	-15287.546	7×10^{-5}	4.8×10^{-9}
75 Re	-15784.533	-15784.533	7×10^{-5}	4.9×10^{-9}
76 Os	-16290.648	-16290.648	8×10^{-5}	5.0×10^{-9}
77 Ir	-16806.113	-16806.113	8×10^{-5}	5.0×10^{-9}
78 Pt	-17331.069	-17331.069	8×10^{-5}	5.0×10^{-9}
79 Au	-17865.399	-17865.400	8×10^{-5}	4.9×10^{-9}
80 Hg	-18408.991	-18408.991	9×10^{-5}	5.0×10^{-9}
81 Tl	-18961.824	-18961.824	8×10^{-5}	4.6×10^{-9}
82 Pb	-19524.008	-19524.008	9×10^{-5}	4.9×10^{-9}
83 Bi	-20095.586	-20095.586	1×10^{-4}	5.0×10^{-9}
84 Po	-20676.500	-20676.500	9×10^{-5}	4.6×10^{-9}
85 At	-21266.881	-21266.881	9×10^{-5}	4.5×10^{-9}
86 Rn	-21866.772	-21866.772	9×10^{-5}	4.3×10^{-9}
55 Cs^r	-7786.771	-7786.771	3×10^{-4}	4.6×10^{-8}
56 Ba^r	-8135.644	-8135.645	3×10^{-4}	4.7×10^{-8}
57 La^r	-8493.543	-8493.543	4×10^{-4}	4.8×10^{-8}
58 Ce^r	-8860.997	-8860.997	4×10^{-4}	4.5×10^{-8}
59 Pr^r	-9238.148	-9238.148	4×10^{-4}	4.6×10^{-8}
60 Nd^r	-9625.131	-9625.131	4×10^{-4}	4.8×10^{-8}
61 Pm^r	-10022.094	-10022.095	5×10^{-4}	4.9×10^{-8}
62 Sm^r	-10429.162	-10429.163	5×10^{-4}	4.8×10^{-8}
63 Eu^r	-10846.504	-10846.505	4×10^{-4}	4.4×10^{-8}
64 Gd^r	-11274.242	-11274.242	5×10^{-4}	4.7×10^{-8}
65 Td^r	-11712.544	-11712.545	5×10^{-4}	4.3×10^{-8}
66 Dy^r	-12161.545	-12161.545	6×10^{-4}	4.9×10^{-8}
67 Ho^r	-12621.412	-12621.413	5×10^{-4}	4.6×10^{-8}
68 Er^r	-13092.269	-13092.270	6×10^{-4}	4.9×10^{-8}

Table 3.3: *(continued)*

Element	Calculated Energy	Literature Energy	Absolute Error	Relative Error
69 Tm^r	-13574.316	-13574.317	6×10^{-4}	4.8×10^{-8}
70 Yb^r	-14067.676	-14067.677	6×10^{-4}	4.6×10^{-8}
71 Lu^r	-14572.532	-14572.533	7×10^{-4}	4.9×10^{-8}
72 Hf^r	-15088.785	-15088.786	7×10^{-4}	4.9×10^{-8}
73 Ta^r	-15616.630	-15616.630	7×10^{-4}	4.7×10^{-8}
74 W^r	-16156.184	-16156.185	8×10^{-4}	4.9×10^{-8}
75 Re^r	-16707.619	-16707.620	8×10^{-4}	4.9×10^{-8}
76 Os^r	-17271.081	-17271.082	8×10^{-4}	4.9×10^{-8}
77 Ir^r	-17846.787	-17846.788	8×10^{-4}	4.9×10^{-8}
78 Pt^r	-18434.873	-18434.874	9×10^{-4}	4.9×10^{-8}
79 Au^r	-19035.525	-19035.526	9×10^{-4}	4.9×10^{-8}
80 Hg^r	-19648.895	-19648.896	9×10^{-4}	4.9×10^{-8}
81 Tl^r	-20274.849	-20274.850	9×10^{-4}	4.8×10^{-8}
82 Pb^r	-20913.713	-20913.714	9×10^{-4}	4.7×10^{-8}
83 Bi^r	-21565.705	-21565.706	1×10^{-3}	4.9×10^{-8}
84 Po^r	-22231.012	-22231.013	9×10^{-4}	4.1×10^{-8}
85 At^r	-22909.806	-22909.807	8×10^{-4}	3.7×10^{-8}
86 Rn^r	-23602.103	-23602.104	1×10^{-3}	4.8×10^{-8}

Figures 3.2, 3.3, and 3.4 show the trends of various non-relativistic parameters with respect to nuclear charge. In Figure 3.2, we can see that there is an interesting pattern where the exponent of the most diffuse function becomes larger as we move right across a row. It is also interesting to note that the rate at which the exponent increases changes when going from one block of that row to the next. The trend roughly follows what we expect from periodic trends of atomic size: atoms get larger by going left on the table. The trend of atoms getting larger when moving down the periodic table is mostly obeyed, however there are exceptions. For instance, the 4d elements have smaller values of α than the 5d elements.

In Figure 3.3 shows that β generally decreases as nuclear charge increases. As β is involved in controlling the growth of the exponents, this indicates that the basis sets are getting more diffuse as the nuclear charge increases.

Finally 3.4 shows the values of δ and γ with respect to nuclear charge. There are no clear trends in these parameters, but they are shown for completeness.

As for the relativistic basis sets, we can see from Figure 3.5 that the trend in α from the non-relativistic basis sets mostly holds, however there is a slight change in the trend starting at thallium. There is a substantial drop in the value of α when changing from the d to the p block in the relativistic basis sets which is not seen in the non-relativistic sets. We can also see that the change in α for the relativistic basis sets is much larger going
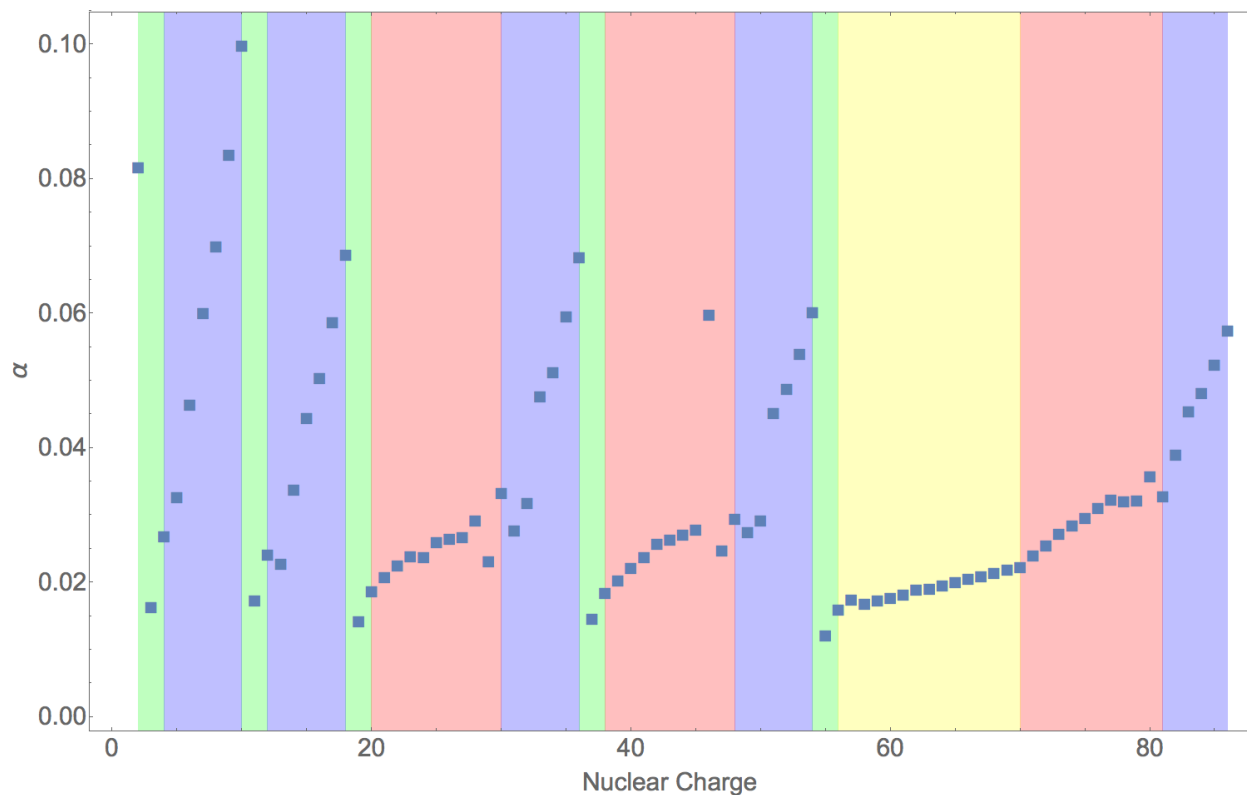
Figure 3.2: *Scatter plot of the α parameter vs. nuclear charge. Regions of the plot are shaded based on periodic table block regions. The s block is green, the p block is blue, the d block is red and the f block is yellow. There is a distinct pattern where the value of α increases along a row until a noble gas is reached. Also note how the rate of the increase changes at periodic table block boundaries.*
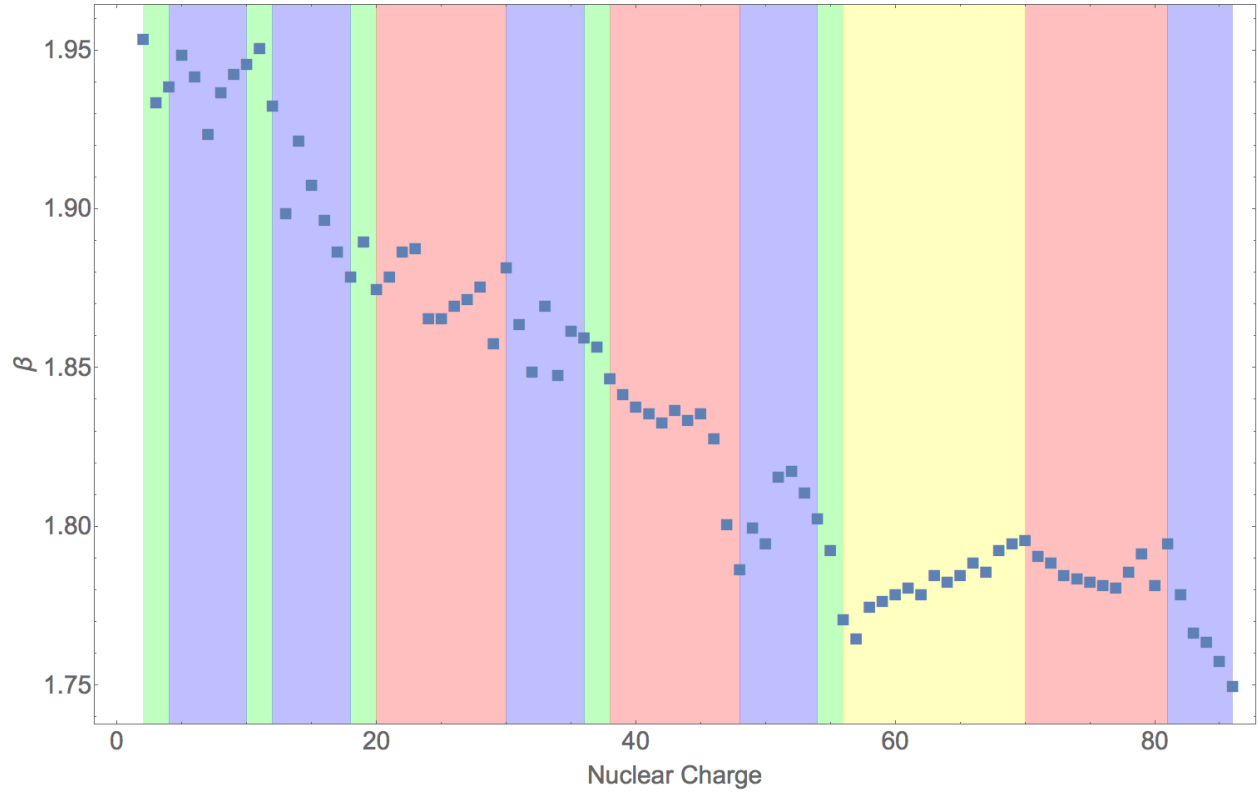
Figure 3.3: *Scatter plot of the β parameter vs. nuclear charge. Regions of the plot are shaded based on periodic table block regions. The s block is green, the p block is blue, the d block is red and the f block is yellow. There is a general trend of the value of β to decrease as the nuclear charge grows larger.*

(a) δ

(b) γ
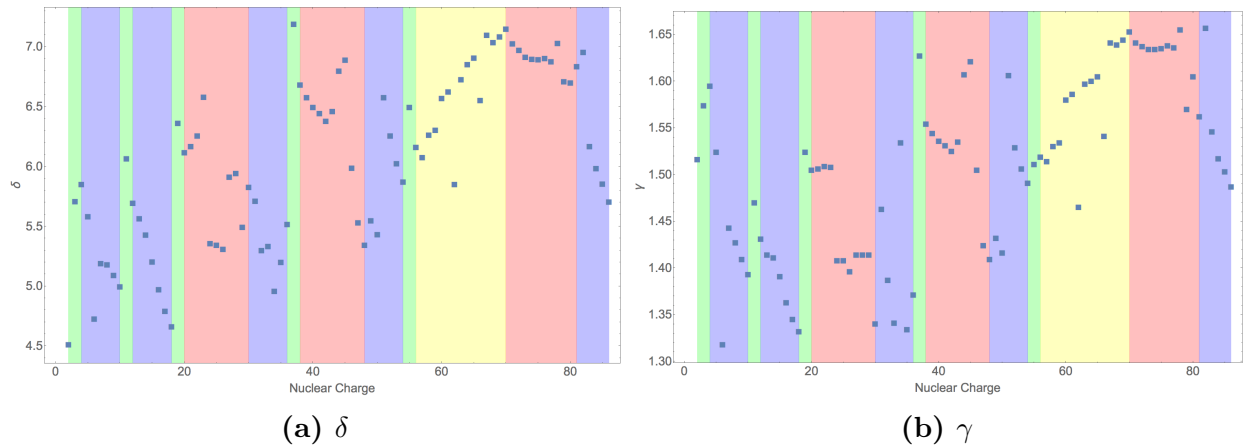
Figure 3.4: *Scatter plots of the δ and γ parameters. Regions of the plot are shaded based on periodic table block regions. The s block is green, the p block is blue, the d block is red and the f block is yellow. There are no clear trends with respect to the nuclear charge.*
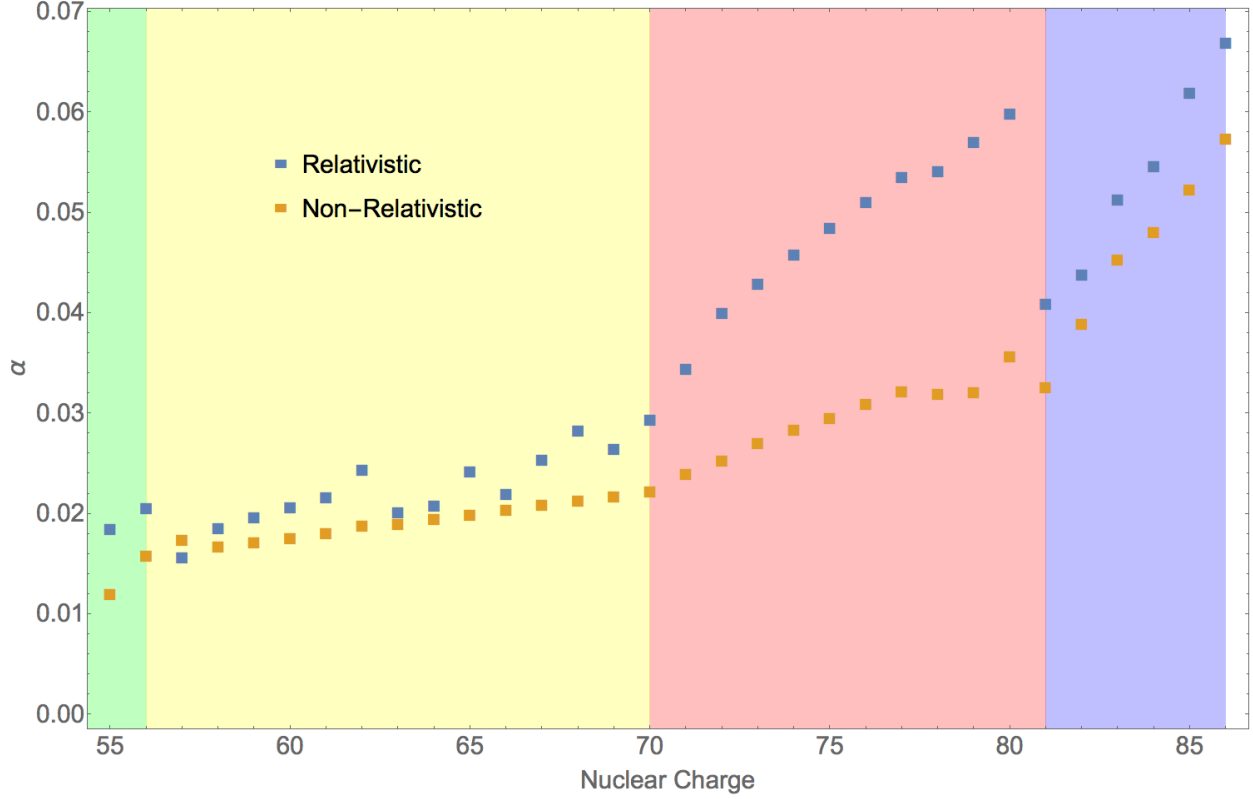
Figure 3.5: *Scatter plot of the relativistic and non-relativistic α parameters vs. nuclear charge. Regions of the plot are shaded based on periodic table block regions. The s block is green, the p block is blue, the d block is red and the f block is yellow.*

through the d block than the change for the non-relativistic basis sets. As for the other WTBS parameters, the trends in the relativistic basis sets are mostly consistent with their non-relativistic counterparts. Figures for the remaining parameters are given in Figures 3.6 and 3.7.
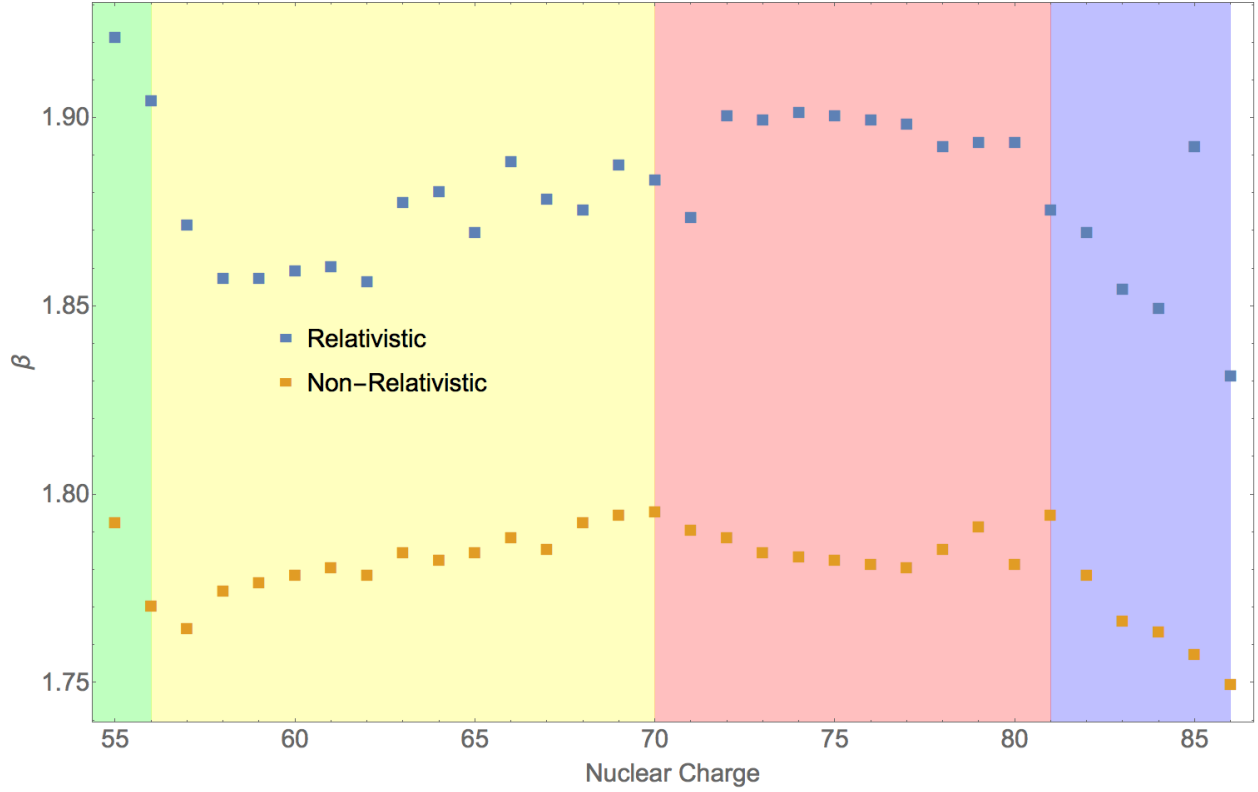
Figure 3.6: *Scatter plot of the relativistic and non-relativistic β parameters vs. nuclear charge. Regions of the plot are shaded based on periodic table block regions. The s block is green, the p block is blue, the d block is red and the f block is yellow.*

(a) δ **(b)** γ
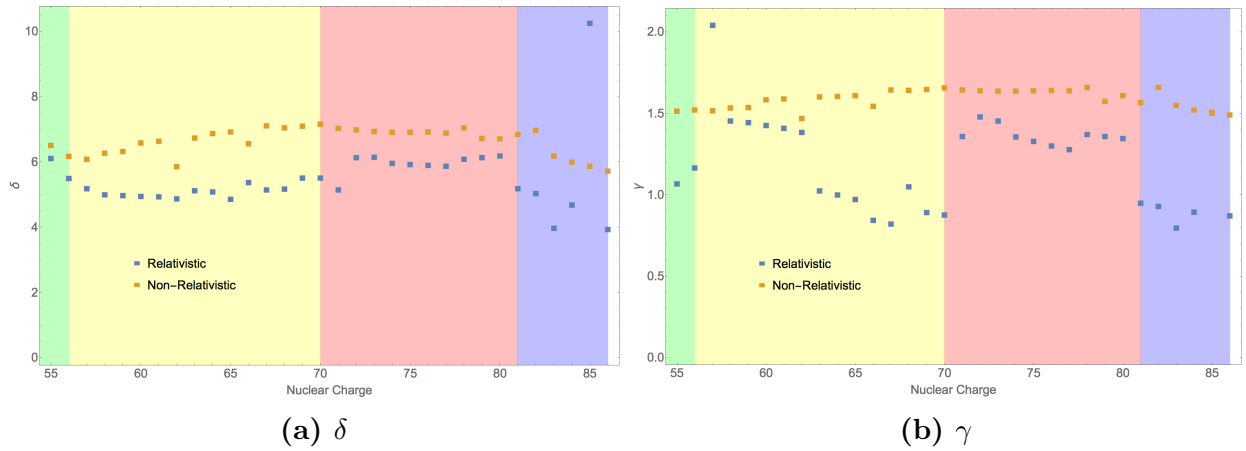
Figure 3.7: *Scatter plot of the relativistic and non-relativistic δ and γ parameters. Regions of the plot are shaded based on periodic table block regions. The s block is green, the p block is blue, the d block is red and the f block is yellow.*

Chapter 4

Conclusions and Future Work

In this thesis I have illustrated how even small problems in computational quantum chemistry can benefit from parallel computation. I have done this both in the internal sense, where I rewrote the DFRATOM program to calculate the electronic structure of atoms on GPUs, and in the external sense, where I designed an algorithm that very effectively optimizes basis sets. This chapter will be broken up into two sections: the first which covers the rewriting of DFRATOM into cudaDFRATOM, and the second one which summarizes the results of basis set optimization using cudaDFRATOM and rwtbs.

4.1   cudaDFRATOM

By harnessing the power of GPUs, I was able get the cudaDFRATOM program to accelerate the biggest bottleneck of these calculations, the evaluation of the two-electron integrals, by almost a factor of 20. This changes the bottleneck to be the matrix operations required in SCF. As a result, the total speedup is only about 2-4 times faster than the original DFRATOM program. While this is not as much of an increase as we originally hoped, it is still a very big improvement on the original program.

It is also worth pointing out that as the problems get larger, the speedup gets larger as well. This indicates that the algorithms written for this program could potentially work even better on even larger problems such as molecules. Modifications required to make the algorithms work on molecules are as follows.

One could implement direct SCF. As the amount of two-electron integrals grows, it becomes impractical to try and store them all in memory. Instead, in the direct SCF algorithm, the integrals are calculated and consumed as needed. The current program has the following flow: the threads are mapped to all the integrals, all the integrals are calculated and stored in memory, and then all the integrals are used to form the $\mathbf{P}$ and $\mathbf{Q}$ matrices. If we wish to reuse as much of the code I've written as possible, the direct SCF implementation would

work as follows. First we would map the threads to a chunk of the integrals, then we could calculate only that chunk, and then we would build up only the elements of the **P** and **Q** that use those integrals. Now, we no longer need these integrals, so we can throw them away and start the calculation of a new chunk. We would then loop over these steps until the the **P** and **Q** matrices have been completely calculated. The mapping takes very little time to calculate, so the fact that we are calling it so many times would have relatively little impact on performance.

One could also add some prescreening of the integrals to determine a rough estimate of their order of magnitude. If a particular integral is very close to zero, the effect it has on the calculation is relatively negligible. Therefore we could skip the calculation of this integral, and there would be almost no impact on the accuracy of the calculation. Due to how GPUs perform the calculations in warps, we would need to presort the integrals so that the ones that are likely to be close to zero are grouped together in order to maximize the returns of this procedure.

One would also need to alter the mapping algorithm so that it can handle calculations of different point groups. Atoms have spherical symmetry, meaning we can eliminate the need to calculate a lot of the integrals because many of them will be symmetrically the same. But this would not be the case in systems that have no symmetry, so we would need to change the mapping accordingly. This would actually be very simple to change, as we would just need to write nearly identical code for each symmetry that we would want to support.

One final comment regarding cudaDFRATOM is the algorithm to form the **P** and **Q** matrices. I am still not completely satisfied with the current state of this part of the code. It is quite complicated and inelegant. It would seem to me that there should be a better way of performing this calculation, but so far I have been unable to improve upon the version that appears in this thesis. One possible approach would be to break up the calculation of the matrices into two different subroutines: one that calculates **P** and one that calculates **Q**. The calculation of **P** can be reformed into a simple matrix multiplication problem, but **Q** requires many conditional statements as a result of having to deal with the math of open shells. I did not try this as I believed the main bottleneck was reading the two-electron integrals. Therefore, I only wanted to have to read them once to reduce the amount of global memory reads as possible. But it might be the case that specialized code for each matrix could outperform generalized code for both.

4.2 Basis Set Optimization

This thesis also covered the optimization of new basis sets. I used a novel algorithm that almost entirely eliminates the need for human involvement in the optimization process. It

produces useful accurate basis sets, that are practically guarantied to be as small as possible. I have optimized non-relativistic basis sets for the elements helium through radon. I have also optimized relativistic basis sets for the elements caesium through radon. Future work in the area would be the optimization of relativistic basis sets for the rest of the periodic table. Care must be taken when optimizing sets for the seventh row elements, because the assumption made for kinetic balancing begins to break down there. As a result, early attempts to optimize basis sets for these elements have proven to be prone to prolapsing. As a result, the method of automating the optimization process should be used with caution in this range, as the results it produces may erroneously appear to be successful.

Further work that could be done includes modifications to the cudaDFRATOM program to support basis sets other than WTBS, and also to include other optimization algorithms than ***newuoa***, which might be less prone to generating prolapsing basis sets. cudaDFRATOM also includes the option to optimize a different set of ζs for different groups of spinor symmetries. Therefore, it might be best to try and optimize basis sets that have grouped the s and p type spinors into one group and the d and f type spinors in another. The logic of this is that relativity has different effects on these two groups, so it might be best to try and separate the two so that each gets its own set of specialized basis functions.

Bibliography

[1] E. Schrödinger. An Undulatory Theory of the Mechanics of Atoms and Molecules. *Phys. Rev.*, 28:1049–1070, 1926.

[2] M. Born and R. Oppenheimer. Zur Quantentheorie der Molekeln. *Ann. Physik*, 389:457–484, 1927.

[3] D. J. Griffiths. *Introduction to Quantum Mechanics.* Pearson Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2005.

[4] S. Kais, D. R. Herschbach, N. C. Handy, C. W. Murray, and G. J. Laming. Density functionals and dimensional renormalization for an exactly solvable model. *J. Chem. Phys.*, 99(1):417–425, 1993.

[5] J. Fourier. *Théorie analytique de la chaleur.* Chez Firmin Didot, père et fils, 1822.

[6] J. C. Slater. The Theory of Complex Spectra. *Phys. Rev.*, 34:1293–1322, 1929.

[7] W. Gerlach and O. Stern. Der experimentelle Nachweis der Richtungsquantelung im Magnetfeld. *Z. Physik*, 9(1):349–352, 1922.

[8] W. Pauli. Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren. *Z. Physik*, 31(1):765–783, 1925.

[9] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry : Introduction to Advanced Electronic Structure Theory.* Dover Publications, Mineola, N.Y, 1996.

[10] C. C. J. Roothaan. New Developments in Molecular Orbital Theory. *Rev. Mod. Phys.*, 23:69–89, 1951.

[11] J. C. Slater. Atomic Shielding Constants. *Phys. Rev.*, 36:57–64, 1930.

[12] S. F. Boys. Electronic wave functions - I. A general method of calculation for the stationary states of any molecular system. *Proc. R. Soc. Lond. A*, 200(1063):542–554, 1950.

[13] G. G. Hall. The molecular orbital theory of chemical valency VIII. A method of calculating ionization potentials. *Proc. R. Soc. Lond. A*, 205(1083):541–552, 1951.

[14] A. A. Michelson and E. W. Morley. On the Relative Motion of the Earth and of the Luminiferous Ether. *Am. J. Sci*, 34:333, 1887.

[15] L. Piela. *Ideas of Quantum Chemistry.* Elsevier, Amsterdam, 2007.

[16] W. Gordon. Der Comptoneffekt nach der Schrödingerschen Theorie. *Z. Physik*, 40(1):117–133, 1926.

[17] O. Klein. Elektrodynamik und Wellenmechanik vom Standpunkt des Korrespondenzprinzips. *Z. Physik A*, 41(6):407–442, 1927.

[18] P. A. M. Dirac. The quantum theory of the electron. *Proc. R. Soc. Lond. A*, 117(778):610–624, 1928.

[19] C. D. Anderson. The Positive Electron. *Phys. Rev.*, 43:491–494, 1933.

[20] N. Bartlett. Relativistic effects and the chemistry of gold. *Gold Bull*, 31(1):22–25, 1998.

[21] L. J. Norrby. Why is mercury liquid? Or, why do relativistic effects not get into chemistry textbooks? *J. Chem. Educ.*, 68(2):110, 1991.

[22] L. Visscher and K. Dyall. Dirac-Fock atomic electronic structure calculations using different nuclear charge distributions. *Atom. Data Nucl. Data Tabl.*, 67(2):207 – 224, 1997.

[23] W. R. Johnson and G. Soff. The Lamb Shift in Hydrogen-like Atoms, $1 \leq Z \leq 110$. *Atom. Data Nucl. Data Tabl.*, 33:405, 1985.

[24] O. Matsuoka and Y. Watanabe. An atomic Dirac-Fock-Roothaan program. *Comput. Phys. Commun*, 139(2):218 – 234, 2001.

[25] PGI. *CUDA Fortran Programming Guide and Reference*, Version 2018.

[26] G. Ruetsch and M. Fatica. *CUDA Fortran for Scientists and Engineers: Best Practices for Efficient CUDA Fortran Programming*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2013.

[27] M. J. D. Powell. *The NEWUOA software for unconstrained optimization without derivatives*, pages 255–297. Springer US, Boston, MA, 2006.

[28] R. Raffenetti and K. Ruedenberg. EvenTempered Representations of Atomic SelfConsistentField Wavefunctions. Technical report, Ames Laboratory, 1973.

[29] S. Huzinaga, M. Klobukowski, and H. Tatewaki. The well-tempered GTF basis sets and their applications in the SCF calculations on N2, CO, Na2, and P2. *Can. J. Chem.*, 63(7):1812–1828, 1985.

[30] S. Huzinaga and M. Klobukowski. The well-tempered GTF basis set and the ab initio molecular calculation. *J. Mol. Struct. (THEOCHEM)*, 135:403–408, 1986.

[31] S. Huzinaga and M. Klobukowski. Well-tempered gaussian basis set expansions of Roothaan-Hartree-Fock atomic wavefunctions for lithium through mercury. *J. Mol. Struct. (THEOCHEM)*, 167(1-210), 1988.

[32] S. Huzinaga and B. Miguel. A comparison of the geometrical sequence formula and the well-tempered formulas for generating gto basis orbital exponents. *Chem. Phys. Lett.*, 175:289 – 291, 1990.

[33] S. Huzinaga and M. Klobukowski. Well-tempered gaussian basis sets for the calculation of matrix hartree-fock wavefunctions. *Chem. Phys. Lett.*, 212:260 – 264, 1993.

[34] M. Barysz. Systematic treatment of relativistic effects accurate through arbitrarily high order in α^2. *J. Chem. Phys.*, 114:9315–9324, 2001.

[35] T. Koga and A. J. Thakkar. Moments and expansion coefficients of atomic electron momentum densities: numerical Hartree - Fock calculations for hydrogen to lawrencium. *J. Phys. B: At. Mol. Opt. Phys*, 29(14):2973, 1996.

Appendix A

cudaDFRATOM Input and Output files

This appendix shows cudaDFRATOM input files for a few common calculations. The output files are also given.

A.1 Single Point Energy with predefined WTBS parameters

This input file will perform a calculation on lead using predefined well-tempered basis set (WTBS) parameters. It uses the "autogen" option to automatically generate electron configurations.

```
1  Pb
2    $contrl jobtype='energy' $end
3    $nuc znuc=82 nucmdl=3 $end
4    $bas nsym=7 bastype='wtbs' $end
5    $wtbs wtbspara= 3.5433483E-02  1.8399050E+00  4.4867042E+00  2.0048353E+00
6        nbs=35 30 30 25 25 20 20 $end
7    $econfig nclose=6 4 4 3 3 1 1 nopen=0 1 1 0 0 0 0 freeel=2 autogen=.t. $end
8    $scf maxitr=50 ixtrp=1 dfctr=0.3 $end
```

Listing A.1: *Input file for calculating the single point energy of Pb.*

```
1    &CONTRL
2    JOBTYPE = energy,
3    C =     137.0359997600000
4    /
5    &NUC
6    ZNUC =      82.00000000000000       ,
7    NUCMDL =               3,
8    RNUC =      0.000000000000000       ,
9    ALPHA =     137688400.8100000
10   /
11   &BAS
12   NSYM =              7,
13   BASTYPE = wtbs ,
14   NGROUP =            1
15   /
16   &WTBS
17   WTBSPARA =    3.5433483000000002E-002,
18               1.839905000000000       ,
19               4.486704200000000       ,
20               2.004835300000000       ,
21   NBS =             35,
22                     30,
23                     30,
24                     25,
25                     25,
26                     20,
27                     20,
28   START =            1,
29                      1,
```

```
30                        1,
31                        1,
32                        1,
33                        1,
34                        1,
35   GROUPS =             1,
36                        1,
37                        1,
38                        1,
39                        1,
40                        1,
41                        1
42   /
43   &ECONFIG
44   NCLOSE =             6,
45                        4,
46                        4,
47                        3,
48                        3,
49                        1,
50                        1,
51   NOPEN =              0,
52                        1,
53                        1,
54                        0,
55                        0,
56                        0,
57                        0,
58   FREEEL =             2,
59   AUTOGEN =  T,
60   NCONF =              3
61   /
62   nconf = 3
63    1  0.00  0.00  2.00  0.00  0.00  0.00  0.00
64    2  0.00  1.00  1.00  0.00  0.00  0.00  0.00
65    3  0.00  2.00  0.00  0.00  0.00  0.00  0.00
66   &SCF
67   MAXITR =            50,
68   IXTRP =              1,
69   DFCTR =   0.3000000000000000         ,
70   THDLL =   1.0000000000000001E-005,
71   THDSL =   9.9999999999999995E-008,
72   THDSS =   1.0000000000000001E-009
73   /
74         31560
75      166068720
76          82680
77          82680
78      221424960
79   Calculating one electron integrals
80   Calculating two electron integrals
81   Calculating guess
82   Starting SCF
83   SCF converged at iteration 21 Final energy =    -20913.71321671859
84
85
86                          1S+          2S+          3S+          4S+          5S+          6S+
87                  -3.2558910E+03 -5.8788484E+02 -1.4375419E+02 -3.3985791E+01 -6.1449434E+00 -5.6643943E-01
88
89
90    3.5433483E-02 -1.9140257E-07 -9.7159398E-06  8.4857239E-05  2.1986574E-04  5.1415800E-04  2.0636937E-02
91    6.5194589E-02  8.2570189E-07  4.1921119E-05 -3.6562370E-04 -9.4709963E-04 -2.1895500E-03  1.7096292E-01
92    1.1995578E-01 -1.9767881E-06 -1.0027465E-04  8.7018398E-04  2.2414603E-03  4.9281942E-03  4.6563303E-01
93    2.2073350E-01  3.5996723E-06  1.8260591E-04 -1.5762467E-03 -4.0630949E-03 -9.8724076E-03  4.9206077E-01
94    4.0626020E-01 -5.6936298E-06 -2.8865907E-04  2.4671919E-03  6.4082992E-03 -1.1737844E-02  1.8824719E-01
95    7.4802875E-01  8.3491287E-06  4.2257668E-04 -3.5241493E-03 -8.9348378E-03 -2.9905373E-01 -3.0010508E-01
96    1.3783189E+00 -1.1366730E-05 -5.8146938E-04  4.8783856E-03  1.5097459E-02 -7.8158376E-01 -5.1043840E-01
97    2.5427420E+00  1.5670362E-05  7.9374845E-04 -6.0504748E-03  2.7343372E-02 -5.2239957E-01 -2.8601945E-01
98    4.6995778E+00 -1.9958101E-05 -1.0286165E-03  6.5909755E-03  5.2434329E-01  7.2696910E-01  3.0687539E-01
99    8.7095624E+00  1.9050301E-05  1.1054521E-03 -2.3451833E-02  9.3510165E-01  8.5987922E-01  3.3842577E-01
100   1.6203217E+01 -2.6144911E-05 -1.4336986E-03 -2.3995069E-01 -1.1495177E-01 -5.6246536E-02 -2.2090975E-02
101   3.0302910E+01  2.8420699E-05  3.7197373E-03 -8.1531809E-01 -1.0354558E+00 -5.7885493E-01 -2.0220940E-01
102   5.7068236E+01  7.7841668E-05  8.1191015E-02 -3.4478787E-01 -3.3138799E-01 -1.7960886E-01 -6.2968065E-02
103   1.0845021E+02 -1.7007069E-04  4.1109997E-01  4.9103230E-01  3.2296539E-01  1.5765479E-01  5.4052594E-02
104   2.0847323E+02 -2.9589527E-03  5.0903323E-01  4.6725787E-01  2.7638266E-01  1.2821564E-01  4.3348064E-02
105   4.0651542E+02 -5.8530537E-02  1.5500969E-01  1.2163377E-01  6.9594624E-02  3.2609524E-02  1.1089595E-02
106   8.0667635E+02 -2.0848170E-01 -1.1977939E-01 -6.2191906E-02 -3.2239372E-02 -1.4726348E-02 -4.9831695E-03
107   1.6348107E+03 -2.9189696E-01 -1.6951366E-01 -8.7418119E-02 -4.4770915E-02 -2.0115505E-02 -6.7655030E-03
108   3.3969005E+03 -2.4399646E-01 -1.2374164E-01 -6.2334097E-02 -3.1843346E-02 -1.4364272E-02 -4.8398193E-03
109   7.2674336E+03 -1.5514532E-01 -7.1929749E-02 -3.5560267E-02 -1.8046013E-02 -8.1052706E-03 -2.7270821E-03
110   1.6080864E+04 -8.5723247E-02 -3.7776777E-02 -1.8531501E-02 -9.3982955E-03 -4.2280724E-03 -1.4235773E-03
111   3.6974132E+04 -4.4254206E-02 -1.8890041E-02 -9.2074939E-03 -4.6580915E-03 -2.0916796E-03 -7.0381814E-04
112   8.8762050E+04 -2.2211543E-02 -9.2756569E-03 -4.5077689E-03 -2.2802569E-03 -1.0247971E-03 -3.4495102E-04
113   2.2355921E+05 -1.1030466E-02 -4.5272219E-03 -2.1929860E-03 -1.1079619E-03 -4.9748954E-04 -1.6740559E-04
114   5.9356339E+05 -5.4653284E-03 -2.2126683E-03 -1.0697906E-03 -5.4040349E-04 -2.4274024E-04 -8.1695794E-05
115   1.6690414E+06 -2.6711743E-03 -1.0700322E-03 -5.1634550E-04 -2.6065392E-04 -1.1702741E-04 -3.9380312E-05
116   4.9925105E+06 -1.2803414E-03 -5.0897019E-04 -2.4533622E-04 -1.2383272E-04 -5.5608079E-05 -1.8713930E-05
117   1.5952595E+07 -5.6814462E-04 -2.2469528E-04 -1.0820468E-04 -5.4596862E-05 -2.4511059E-05 -8.2480821E-06
118   5.4660323E+07 -2.1789453E-04 -8.5878190E-05 -4.1336406E-05 -2.0856799E-05 -9.3648388E-06 -3.1514882E-06
119   2.0153596E+08 -2.1829984E-05 -8.5856049E-06 -4.1303396E-06 -2.0833500E-06 -9.3511877E-07 -3.1465061E-07
120   8.0207663E+08  1.5047170E-06  5.9151312E-07  2.8444110E-07  1.4340856E-07  6.4331467E-08  2.1641631E-08
121   3.4548830E+09 -2.3058546E-07 -9.0633416E-08 -4.3576601E-08 -2.1966863E-08 -9.8520247E-09 -3.3140430E-09
```

```
122   1.6143750E+10  3.5734594E-08  1.4045396E-08  6.7527666E-09  3.4039038E-09  1.5265427E-09  5.1348999E-10
123   8.1990099E+10 -5.0569076E-09 -1.9875952E-09 -9.5559025E-10 -4.8168458E-10 -2.1601720E-10 -7.2662291E-11
124   4.5329140E+11  5.7727825E-10  2.2689646E-10  1.0908642E-10  5.4987096E-11  2.4659548E-11  8.2947901E-12
125
126
127   3.5433483E-02  1.6954819E-09 -1.9755659E-08  1.0517252E-07  2.5786097E-07  6.0705258E-07  2.4548192E-05
128   6.5194589E-02 -6.1192070E-09  9.9944811E-08 -6.0718398E-07 -1.5143279E-06 -3.5153647E-06  2.7587966E-04
129   1.1995578E-01  1.2288718E-08 -2.9311326E-07  1.9454770E-06  4.8761399E-06  1.0747339E-05  1.0191707E-03
130   2.2073350E-01 -1.8377497E-08  6.7468779E-07 -4.7570165E-06 -1.2013960E-05 -2.9243701E-05  1.4610285E-03
131   4.0626020E-01  2.2094317E-08 -1.3755421E-06  1.0067995E-05  2.5740826E-05 -4.7401652E-05  7.5820245E-04
132   7.4802875E-01 -1.9732962E-08  2.6328827E-06 -1.9470061E-05 -4.8740114E-05 -1.6346171E-03 -1.6403132E-03
133   1.3783189E+00  7.0802976E-09 -4.7913481E-06  3.6517250E-05  1.1190759E-04 -5.7997109E-03 -3.7872066E-03
134   2.5427420E+00  3.3353393E-08  8.6948262E-06 -6.1451021E-05  2.7604343E-04 -5.2632551E-03 -2.8813285E-03
135   4.6995778E+00 -1.0558631E-07 -1.5093361E-05  9.0963356E-05  7.1885186E-03  9.9617188E-03  4.2051363E-03
136   8.7095624E+00  1.2906609E-07  2.2027770E-05 -4.3912334E-04  1.7443639E-02  1.6029779E-02  6.3081111E-03
137   1.6203217E+01 -3.6214254E-07 -3.8443478E-05 -6.1234284E-03 -2.9533854E-03 -1.4537621E-03 -5.7043669E-04
138   3.0302910E+01  5.2836853E-07  1.3285756E-04 -2.8417938E-02 -3.6003287E-02 -2.0119705E-02 -7.0279292E-03
139   5.7068236E+01  4.6114972E-06  3.9263140E-03 -1.6339710E-02 -1.5654581E-02 -8.4797945E-03 -2.9727261E-03
140   1.0845021E+02 -1.2950972E-05  2.7337032E-02  3.2407503E-02  2.1295257E-02  1.0391632E-02  3.5625331E-03
141   2.0847323E+02 -2.9073865E-04  4.6496978E-02  4.2189645E-02  2.4898158E-02  1.1543726E-02  3.9023181E-03
142   4.0651542E+02 -8.0613553E-03  1.9021333E-02  1.4724956E-02  8.4033363E-03  3.9370652E-03  1.3390133E-03
143   8.0667635E+02 -3.9844448E-02 -2.1952178E-02 -1.1434990E-02 -5.9321659E-03 -2.7091031E-03 -9.1660186E-04
144   1.6348107E+03 -7.7242278E-02 -4.2059759E-02 -2.1528276E-02 -1.1006504E-02 -4.9428729E-03 -1.6622820E-03
145   3.3969005E+03 -8.8701783E-02 -4.2001749E-02 -2.0959345E-02 -1.0683736E-02 -4.8172258E-03 -1.6229926E-03
146   7.2674336E+03 -7.6482772E-02 -3.3116809E-02 -1.6206858E-02 -8.2039085E-03 -3.6819580E-03 -1.2386094E-03
147   1.6080864E+04 -5.6100232E-02 -2.3179626E-02 -1.1261204E-02 -5.6984512E-03 -2.5626001E-03 -8.6278680E-04
148   3.6974132E+04 -3.7347045E-02 -1.5052553E-02 -7.2708423E-03 -3.6698025E-03 -1.6466392E-03 -5.5396552E-04
149   8.8762050E+04 -2.3278150E-02 -9.2698061E-03 -4.4718637E-03 -2.2583779E-03 -1.0147581E-03 -3.4157455E-04
150   2.2355921E+05 -1.3784938E-02 -5.4532697E-03 -2.6256301E-03 -1.3243959E-03 -5.9430475E-04 -1.9995104E-04
151   5.9356339E+05 -7.7805110E-03 -3.0685330E-03 -1.4775435E-03 -7.4574987E-04 -3.3497969E-04 -1.1274508E-04
152   1.6690414E+06 -4.1729867E-03 -1.6425066E-03 -7.9020592E-04 -3.9854800E-04 -1.7886230E-04 -6.0180408E-05
153   4.9925105E+06 -2.0861996E-03 -8.2044252E-04 -3.9483492E-04 -1.9924164E-04 -8.9483516E-05 -3.0116278E-05
154   1.5952595E+07 -9.3209819E-04 -3.6627633E-04 -1.7617169E-04 -8.8851596E-05 -3.9876899E-05 -1.3417320E-05
155   5.4660323E+07 -2.8134459E-04 -1.1055138E-04 -5.3198526E-05 -2.6846343E-05 -1.2058514E-05 -4.0585399E-06
156   2.0153596E+08 -5.0398556E-06 -1.9824128E-06 -9.4788639E-07 -4.7454755E-07 -2.1080489E-07 -7.0655962E-08
157   8.0207663E+08 -9.6484793E-07 -3.7958549E-07 -1.8491005E-07 -9.4673401E-08 -4.3358342E-08 -1.4697821E-08
158   3.4548830E+09  4.2903511E-07  1.6878873E-07  8.2035379E-08  4.1888247E-08  1.9114999E-08  6.4712118E-09
159   1.6143750E+10 -1.5185021E-07 -5.9740539E-08 -2.9026287E-08 -1.4815664E-08 -6.7575308E-09 -2.2872876E-09
160   8.1990099E+10  4.8899010E-08  1.9237777E-08  9.3465918E-09  4.7703943E-09  2.1756194E-09  7.3637952E-10
161   4.5329140E+11 -1.3146726E-08 -5.1721694E-09 -2.5128521E-09 -1.2825163E-09 -5.8490432E-10 -1.9797080E-10
162
163
164                        2P-            3P-            4P-            5P-            6P-
165                 -5.6348704E+02 -1.3280392E+02 -2.9211810E+01 -4.4404008E+00 -2.7507268E-01
166
167
168   3.5433483E-02 -2.1110836E-06  3.3329049E-05  8.2318337E-05 -2.3099532E-04 -9.5114313E-02
169   6.5194589E-02  8.2038732E-06 -1.2966616E-04 -3.2228822E-04  1.1078969E-03 -2.4887441E-01
170   1.1995578E-01 -1.8429299E-05  2.8944290E-04  7.0702895E-04 -1.5754355E-03 -3.8418187E-01
171   2.2073350E-01  3.2465036E-05 -5.0746384E-04 -1.2533334E-03  6.4782033E-03 -3.1103172E-01
172   4.0626020E-01 -5.0544404E-05  7.8669527E-04  1.8997745E-03  2.6096020E-02 -1.4702572E-01
173   7.4802875E-01  7.5145363E-05 -1.1223456E-03 -2.6453479E-03  2.1315146E-01  7.3403494E-02
174   1.3783189E+00 -1.0242435E-04  1.5950138E-03  5.1813384E-03  5.0964189E-02  2.0051539E-01
175   2.5427420E+00  1.4757652E-04 -2.0117491E-03  2.1606845E-02  4.8490218E-01  1.8040757E-01
176   4.6995778E+00 -1.9939967E-04  2.2268120E-03  2.6654648E-01 -1.0409201E-01 -2.2131882E-02
177   8.7095624E+00  1.8953650E-04 -9.3713975E-03  6.2243748E-01 -4.6368432E-01 -1.3477866E-01
178   1.6203217E+01 -3.7221586E-04 -9.7628883E-02  3.4575529E-01 -2.3504320E-01 -6.6197533E-02
179   3.0302910E+01  9.7479204E-04 -4.2695288E-01 -3.0935771E-01  1.3640741E-01  3.6084562E-02
180   5.7068236E+01  2.2896984E-02 -4.9060947E-01 -4.0033696E-01  1.9231613E-01  5.1750801E-02
181   1.0845021E+02  1.5476535E-01 -1.0565728E-01 -9.4601050E-02  4.4583201E-02  1.1943673E-02
182   2.0847323E+02  3.1765576E-01  1.6311570E-01  7.9749840E-02 -3.3447982E-02 -8.8056305E-03
183   4.0651542E+02  3.1357302E-01  1.8279811E-01  9.5209998E-02 -4.0980859E-02 -1.0840501E-02
184   8.0667635E+02  2.0254912E-01  1.1347400E-01  5.8154906E-02 -2.4840415E-02 -6.5613532E-03
185   1.6348107E+03  1.0544176E-01  5.7122173E-02  2.9134598E-02 -1.2463954E-02 -3.2936776E-03
186   3.3969005E+03  5.0059648E-02  2.6435568E-02  1.3361681E-02 -5.6918307E-03 -1.5028703E-03
187   7.2674336E+03  2.3694166E-02  1.2341259E-02  6.2310645E-03 -2.6582428E-03 -7.0213261E-04
188   1.6080864E+04  1.1638547E-02  6.0023349E-03  3.0182407E-03 -1.2847058E-03 -3.3918314E-04
189   3.6974132E+04  5.9982431E-03  3.0784422E-03  1.5480068E-03 -6.5951230E-04 -1.7415861E-04
190   8.8762050E+04  3.1931225E-03  1.6331902E-03  8.1995336E-04 -3.4899562E-04 -9.2142104E-05
191   2.2355921E+05  1.7236138E-03  8.8035369E-04  4.4204912E-04 -1.8822394E-04 -4.9699458E-05
192   5.9356339E+05  9.1977675E-04  4.6933992E-04  2.3554723E-04 -1.0026146E-04 -2.6471634E-05
193   1.6690414E+06  4.7923133E-04  2.4446020E-04  1.2269895E-04 -5.2235504E-05 -1.3792011E-05
194   4.9925105E+06  2.3419075E-04  1.1943149E-04  5.9934273E-05 -2.5511986E-05 -6.7358838E-06
195   1.5952595E+07  1.0466575E-04  5.3372488E-05  2.6785502E-05 -1.1402544E-05 -3.0106243E-06
196   5.4660323E+07  3.0692464E-05  1.5649976E-05  7.8534060E-06 -3.3429305E-06 -8.8262848E-07
197   2.0153596E+08  8.6822796E-07  4.4288491E-07  2.2234096E-07 -9.4676609E-08 -2.4999155E-08
198
199
200   3.5433483E-02 -8.5886264E-06 -3.8710980E-08 -1.3229008E-06 -1.4244408E-07 -1.4597307E-04
201   6.5194589E-02  4.8437004E-06 -2.3081520E-07  1.4609437E-07  2.1864265E-06 -5.1850903E-04
202   1.1995578E-01 -1.4154628E-05  6.9349778E-07 -3.8764579E-07 -4.0988158E-06 -1.0854283E-03
203   2.2073350E-01 -2.5120815E-06 -2.0262797E-06 -5.2390142E-06  2.4888258E-05 -1.1921947E-03
204   4.0626020E-01 -2.0001927E-06  3.9614954E-06  6.5357907E-06  1.3622028E-04 -7.6431931E-04
205   7.4802875E-01 -1.9153197E-05 -8.2676519E-06 -2.1970577E-06  1.5046689E-03  5.1817302E-04
206   1.3783189E+00 -4.0436086E-05  1.5053840E-05  4.2962105E-05  4.8829178E-03  1.9209398E-03
207   2.5427420E+00 -5.4843800E-05 -2.7098584E-05  2.7176505E-04  6.3096303E-03  2.3470890E-03
208   4.6995778E+00 -9.7950298E-05  3.8804793E-05  4.7013211E-03 -1.8388012E-03 -3.9119711E-04
209   8.7095624E+00 -1.4327167E-04 -2.2834376E-04  1.4966814E-02 -1.1152273E-02 -3.2417362E-03
210   1.6203217E+01 -2.5163894E-04 -3.2178309E-03  1.1306508E-02 -7.6887039E-03 -2.1654934E-03
211   3.0302910E+01 -3.4077759E-04 -1.9221810E-02 -1.3934741E-02  6.1499391E-03  1.6271672E-03
212   5.7068236E+01  8.0186654E-04 -3.0239516E-02 -2.4603559E-02  1.1813559E-02  3.1787340E-03
213   1.0845021E+02  1.2259914E-02 -8.9882096E-03 -7.9527104E-03  3.7414154E-03  1.0020520E-03
```

```
214   2.0847323E+02  3.5937378E-02  1.8702867E-02  9.1824211E-03 -3.8549441E-03 -1.0150111E-03
215   4.0651542E+02  4.8738390E-02  2.8588551E-02  1.4912210E-02 -6.4213853E-03 -1.6987496E-03
216   8.0667635E+02  4.2344757E-02  2.3814232E-02  1.2212514E-02 -5.2165181E-03 -1.3778789E-03
217   1.6348107E+03  2.8772905E-02  1.5644329E-02  7.9877030E-03 -3.4187444E-03 -9.0350250E-04
218   3.3969005E+03  1.6986116E-02  9.0076332E-03  4.5551383E-03 -1.9399597E-03 -5.1219513E-04
219   7.2674336E+03  9.3658325E-03  4.9076239E-03  2.4828687E-03 -1.0602513E-03 -2.8010111E-04
220   1.6080864E+04  4.9869445E-03  2.5880416E-03  1.3021501E-03 -5.5395659E-04 -1.4623381E-04
221   3.6974132E+04  2.6253728E-03  1.3570783E-03  6.8425087E-04 -2.9193075E-04 -7.7112040E-05
222   8.8762050E+04  1.3744607E-03  7.0676310E-04  3.5489843E-04 -1.5093380E-04 -3.9842201E-05
223   2.2355921E+05  7.1797657E-04  3.6856498E-04  1.8548117E-04 -7.9083365E-05 -2.0887093E-05
224   5.9356339E+05  3.7046632E-04  1.8956211E-04  9.5105240E-05 -4.0448225E-05 -1.0677407E-05
225   1.6690414E+06  1.8661865E-04  9.5434683E-05  4.7970253E-05 -2.0442610E-05 -5.3986815E-06
226   4.9925105E+06  8.8855319E-05  4.5355929E-05  2.2748343E-05 -9.6761026E-06 -2.5543594E-06
227   1.5952595E+07  3.7642546E-05  1.9216236E-05  9.6534660E-06 -4.1127730E-06 -1.0860871E-06
228   5.4660323E+07  1.0702166E-05  5.4565863E-06  2.7355003E-06 -1.1632623E-06 -3.0707008E-07
229   2.0153596E+08  3.3837491E-07  1.7331530E-07  8.7896073E-08 -3.7781680E-08 -9.9958460E-09
230
231
232                               2P+            3P+            4P+            5P+            6P+
233                  -4.8289919E+02 -1.1459355E+02 -2.4696490E+01 -3.5936227E+00 -2.1988025E-01
234
235
236   3.5433483E-02 -1.6353670E-06  3.0868694E-05 -1.1864655E-04  1.5855754E-04 -2.1084460E-01
237   6.5194589E-02  6.3316603E-06 -1.2026428E-04  4.6694629E-04 -1.2548766E-03 -2.8111761E-01
238   1.1995578E-01 -1.4219884E-05  2.6796577E-04 -1.0180223E-03  1.0820358E-03 -3.9737451E-01
239   2.2073350E-01  2.4992100E-05 -4.6903534E-04  1.7890741E-03 -1.0206289E-02 -2.2012706E-01
240   4.0626020E-01 -3.8612896E-05  7.2606778E-04 -2.6931726E-03 -5.9228112E-02 -7.0584944E-02
241   7.4802875E-01  5.7574583E-05 -1.0210499E-03  3.4705786E-03 -3.0443611E-01  1.2807543E-01
242   1.3783189E+00 -7.5089141E-05  1.4700244E-03 -8.3211961E-03 -5.4484929E-01  1.8731223E-01
243   2.5427420E+00  1.0885749E-04 -1.7738974E-03 -5.0169252E-02 -3.5977524E-01  2.1236805E-01
244   4.6995778E+00 -1.3997879E-04  1.6176208E-03 -3.7440022E-01  2.5348989E-01 -6.5756816E-02
245   8.7095624E+00  6.1564771E-05 -1.7286003E-02 -6.3300409E-01  4.4847514E-01 -1.1845734E-01
246   1.6203217E+01 -1.9253182E-04 -1.6403996E-01 -1.7584481E-01  1.2440143E-01 -3.2537953E-02
247   3.0302910E+01  1.9618065E-03 -5.0777453E-01  4.1846218E-01 -1.9083216E-01  4.7187098E-02
248   5.7068236E+01  4.7503498E-02 -4.2382810E-01  3.4505648E-01 -1.6162607E-01  4.0084398E-02
249   1.0845021E+02  2.2099001E-01  1.4577962E-02  1.4811851E-02 -8.1127462E-03  2.0606073E-03
250   2.0847323E+02  3.5443326E-01  2.1417794E-01 -1.1076659E-01  4.6283226E-02 -1.1300549E-02
251   4.0651542E+02  2.8912370E-01  1.7573134E-01 -9.2133021E-02  3.8900389E-02 -9.5072685E-03
252   8.0667635E+02  1.5637333E-01  9.0065826E-02 -4.6295924E-02  1.9377585E-02 -4.7315319E-03
253   1.6348107E+03  6.6617703E-02  3.7175835E-02 -1.9041634E-02  7.9919236E-03 -1.9516142E-03
254   3.3969005E+03  2.4461997E-02  1.3316345E-02 -6.7495936E-03  2.8159453E-03 -6.8726177E-04
255   7.2674336E+03  8.2393379E-03  4.4493450E-03 -2.2612571E-03  9.4762317E-04 -2.3135097E-04
256   1.6080864E+04  2.6125442E-03  1.3961041E-03 -7.0410062E-04  2.9329310E-04 -7.1567591E-05
257   3.6974132E+04  7.9905759E-04  4.2714895E-04 -2.1667865E-04  9.0808371E-05 -2.2169026E-05
258   8.8762050E+04  2.3721957E-04  1.2581238E-04 -6.3293550E-05  2.6335175E-05 -6.4253996E-06
259   2.2355921E+05  6.9144743E-05  3.6785036E-05 -1.8651687E-05  7.8206540E-06 -1.9092936E-06
260   5.9356339E+05  1.9622230E-05  1.0354569E-05 -5.1982446E-06  2.1602316E-06 -5.2700445E-07
261   1.6690414E+06  5.4240046E-06  2.8781862E-06 -1.4602187E-06  6.1295588E-07 -1.4965542E-07
262   4.9925105E+06  1.4307210E-06  7.5187181E-07 -3.7635023E-07  1.5603752E-07 -3.8058983E-08
263   1.5952595E+07  3.5694444E-07  1.8952330E-07 -9.6468744E-08  4.0629470E-08 -9.9224055E-09
264   5.4660323E+07  7.8336742E-08  4.0935008E-08 -2.0342964E-08  8.3784907E-09 -2.0424753E-09
265   2.0153596E+08  4.7974022E-09  2.6197636E-09 -1.3906157E-09  6.0777694E-10 -1.4885981E-10
266
267
268   3.5433483E-02 -3.3155956E-09  5.3478695E-08 -1.8607538E-07  2.4510510E-07 -3.2381110E-04
269   6.5194589E-02  1.6117769E-08 -2.7246598E-07  9.8672929E-07 -2.6198936E-06 -5.8561740E-04
270   1.1995578E-01 -4.6595294E-08  8.0476637E-07 -2.9071371E-06  3.0698562E-06 -1.1228754E-03
271   2.2073350E-01  1.0721820E-07 -1.8815173E-06  6.9116575E-06 -3.9145196E-05 -8.4376525E-04
272   4.0626020E-01 -2.1931450E-07  3.9092356E-06 -1.4091566E-05 -3.0799063E-04 -3.6704690E-04
273   7.4802875E-01  4.3505976E-07 -7.4083076E-06  2.4621994E-05 -2.1484003E-03  9.0376973E-04
274   1.3783189E+00 -7.6294621E-07  1.4383728E-05 -7.9917296E-05 -5.2189035E-03  1.7940892E-03
275   2.5427420E+00  1.4824094E-06 -2.3519099E-05 -6.5281515E-04 -4.6799819E-03  1.5786549E-03
276   4.6995778E+00 -2.5722484E-06  2.9232619E-05 -6.6255565E-03  4.4850584E-03 -1.1635075E-03
277   8.7095624E+00  1.6116575E-06 -4.1795075E-04 -1.5241335E-02  1.0793797E-02 -2.8508509E-03
278   1.6203217E+01 -6.4987104E-06 -5.3992894E-03 -5.7571411E-03  4.0723665E-03 -1.0651199E-03
279   3.0302910E+01  8.9154726E-05 -2.2837477E-02  1.8797085E-02 -8.5717355E-03  2.1195362E-03
280   5.7068236E+01  2.9590464E-03 -2.6050224E-02  2.1169534E-02 -9.9123723E-03  2.4582134E-03
281   1.0845021E+02  1.8924339E-02  1.3746655E-03  1.1499728E-03 -6.3925869E-04  1.6268677E-04
282   2.0847323E+02  4.1819595E-02  2.5124039E-02 -1.2984778E-02  5.4250241E-03 -1.3245584E-03
283   4.0651542E+02  4.7051829E-02  2.8364036E-02 -1.4844976E-02  6.2656237E-03 -1.5312391E-03
284   8.0667635E+02  3.5114732E-02  2.0031178E-02 -1.0273268E-02  4.2975255E-03 -1.0492687E-03
285   1.6348107E+03  2.0591420E-02  1.1372953E-02 -5.8119982E-03  2.4383402E-03 -5.9539783E-04
286   3.3969005E+03  1.0336508E-02  5.5566545E-03 -2.8130957E-03  1.1726399E-03 -2.8616504E-04
287   7.2674336E+03  4.7071277E-03  2.5160046E-03 -1.2761248E-03  5.3473105E-04 -1.3054238E-04
288   1.6080864E+04  1.9803439E-03  1.0471764E-03 -5.2638914E-04  2.1895806E-04 -5.3420806E-05
289   3.6974132E+04  7.8378096E-04  4.1574989E-04 -2.1078545E-04  8.8425502E-05 -2.1588333E-05
290   8.8762050E+04  2.9062695E-04  1.5274519E-04 -7.6497777E-05  3.1733883E-05 -7.7405204E-06
291   2.2355921E+05  1.0196314E-04  5.4097845E-05 -2.7514113E-05  1.1580086E-05 -2.8278961E-06
292   5.9356339E+05  3.3260336E-05  1.7397118E-05 -8.6605403E-06  3.5729027E-06 -8.7110633E-07
293   1.6690414E+06  1.0232548E-05  5.4559496E-06 -2.7975585E-06  1.1862834E-06 -2.8986653E-07
294   4.9925105E+06  2.8704711E-06  1.4875248E-06 -7.2996545E-07  2.9700737E-07 -7.2330628E-08
295   1.5952595E+07  7.5427076E-07  4.0779464E-07 -2.1343459E-07  9.2166263E-08 -2.2552841E-08
296   5.4660323E+07  1.4613123E-07  7.3243809E-08 -3.3993336E-08  1.3052326E-08 -3.1629306E-09
297   2.0153596E+08  1.9315900E-09  2.1567584E-09 -1.9670922E-09  1.1635497E-09 -2.9069129E-10
298
299
300                               3D-            4D-            5D-
301                  -9.6810642E+01 -1.6918558E+01 -1.1386940E+00
302
303
304   3.5433483E-02  7.5784238E-06  4.3267270E-05  7.8236958E-05
305   6.5194589E-02 -2.6993559E-05 -1.5341514E-04 -1.0607136E-03
```

```
306   1.1995578E-01  5.7267393E-05  3.1485487E-04 -7.3354507E-03
307   2.2073350E-01 -9.8181660E-05 -5.4111707E-04 -5.5260441E-02
308   4.0626020E-01  1.5037695E-04  7.8310505E-04 -1.7246183E-01
309   7.4802875E-01 -2.1738056E-04 -8.3176454E-04 -3.2770786E-01
310   1.3783189E+00  3.0219815E-04  6.3470740E-03 -3.8344465E-01
311   2.5427420E+00 -4.4994548E-04  5.7171355E-02 -2.4878259E-01
312   4.6995778E+00  3.9049198E-05  2.6346697E-01  4.0375292E-02
313   8.7095624E+00 -8.0122761E-03  4.5805697E-01  2.0149629E-01
314   1.6203217E+01 -7.4226675E-02  3.3001447E-01  1.4691305E-01
315   3.0302910E+01 -2.5949707E-01 -6.3462926E-03  1.1012835E-01
316   5.7068236E+01 -3.7266615E-01 -1.8898131E-01 -6.5016978E-02
317   1.0845021E+02 -2.8171189E-01 -1.5480799E-01 -5.4074551E-02
318   2.0847323E+02 -1.4190235E-01 -7.8386764E-02 -2.7469477E-02
319   4.0651542E+02 -5.4081638E-02 -2.9251770E-02 -1.0187547E-02
320   8.0667635E+02 -1.7643813E-02 -9.4798446E-03 -3.3057717E-03
321   1.6348107E+03 -5.2335178E-03 -2.7774063E-03 -9.6304002E-04
322   3.3969005E+03 -1.5325460E-03 -8.1584319E-04 -2.8426295E-04
323   7.2674336E+03 -4.6095060E-04 -2.4259466E-04 -8.3902621E-05
324   1.6080864E+04 -1.4789734E-04 -7.8497956E-05 -2.7352488E-05
325   3.6974132E+04 -4.7671984E-05 -2.5017391E-05 -8.6454048E-06
326   8.8762050E+04 -1.6984983E-05 -8.9980573E-06 -3.1333861E-06
327   2.2355921E+05 -4.5486942E-06 -2.3853540E-06 -8.2421615E-07
328   5.9356339E+05 -2.4306190E-06 -1.2844651E-06 -4.4653051E-07
329
330
331   3.5433483E-02  6.6378334E-08  5.0504761E-08  1.5844785E-07
332   6.5194589E-02 -5.5134818E-08 -3.8586568E-07 -2.6097866E-06
333   1.1995578E-01  3.3208145E-07  9.7973562E-07 -2.4484243E-05
334   2.2073350E-01 -2.5675922E-07 -2.5626833E-06 -2.5057139E-04
335   4.0626020E-01  1.4400230E-06  4.5493849E-06 -1.0610208E-03
336   7.4802875E-01 -8.2919598E-07 -7.4959695E-06 -2.7357957E-03
337   1.3783189E+00  5.6902067E-06  7.0748212E-05 -4.3450043E-03
338   2.5427420E+00 -2.2419726E-06  8.7781103E-04 -3.8281416E-03
339   4.6995778E+00  1.1210953E-05  5.5097183E-03  8.4595301E-04
340   8.7095624E+00 -2.0650628E-04  1.3039108E-02  5.7362336E-03
341   1.6203217E+01 -2.8411004E-03  1.2798368E-02  5.6966163E-03
342   3.0302910E+01 -1.3701878E-02 -3.4945280E-04  5.7576690E-04
343   5.7068236E+01 -2.6948928E-02 -1.3689346E-02 -4.7117273E-03
344   1.0845021E+02 -2.7850497E-02 -1.5316871E-02 -5.3508675E-03
345   2.0847323E+02 -1.9120857E-02 -1.0566583E-02 -3.7033238E-03
346   4.0651542E+02 -9.8424297E-03 -5.3236691E-03 -1.8539306E-03
347   8.0667635E+02 -4.2476636E-03 -2.2836292E-03 -7.9658186E-04
348   1.6348107E+03 -1.6035530E-03 -8.5066614E-04 -2.9479257E-04
349   3.3969005E+03 -5.6267228E-04 -3.0040086E-04 -1.0484829E-04
350   7.2674336E+03 -1.8688183E-04 -9.8013826E-05 -3.3782723E-05
351   1.6080864E+04 -6.0891217E-05 -3.2665057E-05 -1.1464903E-05
352   3.6974132E+04 -1.8933714E-05 -9.7813981E-06 -3.3354802E-06
353   8.8762050E+04 -6.0219396E-06 -3.2810341E-06 -1.1659544E-06
354   2.2355921E+05 -1.6877326E-06 -8.4801421E-07 -2.8284055E-07
355   5.9356339E+05 -6.4693359E-07 -3.5510080E-07 -1.2693531E-07
356
357
358                           3D+          4D+          5D+
359                  -9.2970175E+01 -1.6065511E+01 -1.0359352E+00
360
361
362   3.5433483E-02  7.7221792E-06  4.6214141E-05  2.1952461E-05
363   6.5194589E-02 -2.7509786E-05 -1.6424473E-04 -1.4251127E-03
364   1.1995578E-01  5.8345961E-05  3.3612523E-04 -1.0625745E-02
365   2.2073350E-01 -9.9974735E-05 -5.7481199E-04 -6.7684219E-02
366   4.0626020E-01  1.5324515E-04  8.4486783E-04 -1.9241282E-01
367   7.4802875E-01 -2.2082310E-04 -8.0559176E-04 -3.3969689E-01
368   1.3783189E+00  3.0703567E-04  7.6648169E-03 -3.7325459E-01
369   2.5427420E+00 -4.7377597E-04  6.6624936E-02 -2.2119312E-01
370   4.6995778E+00 -4.9467905E-05  2.8397802E-01  6.0585886E-02
371   8.7095624E+00 -9.6035520E-03  4.6213229E-01  2.0036287E-01
372   1.6203217E+01 -8.3937657E-02  3.0682572E-01  1.3437091E-01
373   3.0302910E+01 -2.7503581E-01 -2.9655066E-02  1.3186423E-03
374   5.7068236E+01 -3.7518565E-01 -1.9434691E-01 -6.5989400E-02
375   1.0845021E+02 -2.7089678E-01 -1.4943064E-01 -5.1260616E-02
376   2.0847323E+02 -1.3041137E-01 -7.2047242E-02 -2.4782130E-02
377   4.0651542E+02 -4.6786023E-02 -2.5268754E-02 -8.6328259E-03
378   8.0667635E+02 -1.3903256E-02 -7.4694054E-03 -2.5574685E-03
379   1.6348107E+03 -3.5609354E-03 -1.8857418E-03 -6.4088663E-04
380   3.3969005E+03 -8.3665106E-04 -4.4694872E-04 -1.5325926E-04
381   7.2674336E+03 -1.8163087E-04 -9.4900642E-05 -3.2022507E-05
382   1.6080864E+04 -3.7997791E-05 -2.0491370E-05 -7.0895064E-06
383   3.6974132E+04 -7.3043431E-06 -3.7234375E-06 -1.2330269E-06
384   8.8762050E+04 -1.4317186E-06 -7.9759591E-07 -2.8274095E-07
385   2.2355921E+05 -2.3262340E-07 -1.1047266E-07 -3.4394045E-08
386   5.9356339E+05 -5.4159164E-08 -3.1421800E-08 -1.1456088E-08
387
388
389   3.5433483E-02  1.7888758E-08  8.7508825E-08  4.1293182E-08
390   6.5194589E-02 -8.0684778E-08 -4.1663370E-07 -3.5173993E-06
391   1.1995578E-01  2.2176303E-07  1.1482697E-06 -3.5517803E-05
392   2.2073350E-01 -4.9930096E-07 -2.6490778E-06 -3.0699962E-04
393   4.0626020E-01  1.0155200E-06  5.2650364E-06 -1.1839084E-03
394   7.4802875E-01 -1.9543164E-06 -6.8268672E-06 -2.8362358E-03
395   1.3783189E+00  3.6453134E-06  8.7043689E-05 -4.2300188E-03
396   2.5427420E+00 -7.5384829E-06  1.0257309E-03 -3.4043702E-03
397   4.6995778E+00 -6.8517101E-07  5.9447614E-03  1.2688310E-03
```

```
398   8.7095624E+00 -2.7462163E-04  1.3164001E-02  5.7061308E-03
399   1.6203217E+01 -3.2669877E-03  1.1909188E-02  5.2142294E-03
400   3.0302910E+01 -1.4631652E-02 -1.5947712E-03  6.1298599E-05
401   5.7068236E+01 -2.7323725E-02 -1.4142790E-02 -4.8018676E-03
402   1.0845021E+02 -2.7076282E-02 -1.4913646E-02 -5.1146126E-03
403   2.0847323E+02 -1.7926708E-02 -9.8861261E-03 -3.3994994E-03
404   4.0651542E+02 -8.8500970E-03 -4.7696689E-03 -1.6288033E-03
405   8.0667635E+02 -3.6213113E-03 -1.9414600E-03 -6.6454224E-04
406   1.6348107E+03 -1.2728527E-03 -6.7217021E-04 -2.2827054E-04
407   3.3969005E+03 -4.0840295E-04 -2.1786340E-04 -7.4727838E-05
408   7.2674336E+03 -1.1946568E-04 -6.2106636E-05 -2.0903646E-05
409   1.6080864E+04 -3.3287385E-05 -1.8026340E-05 -6.2634465E-06
410   3.6974132E+04 -8.1967293E-06 -4.0946848E-06 -1.3349317E-06
411   8.8762050E+04 -2.0517720E-06 -1.1841925E-06 -4.3034993E-07
412   2.2355921E+05 -3.8529477E-07 -1.5977439E-07 -4.3020909E-08
413   5.9356339E+05 -1.0740559E-07 -7.0576947E-08 -2.7714538E-08
414
415
416                            4F-
417                 -5.9331805E+00
418
419
420   3.5433483E-02 -1.5255101E-05
421   6.5194589E-02  4.9086264E-05
422   1.1995578E-01 -9.8691162E-05
423   2.2073350E-01  1.2122985E-04
424   4.0626020E-01 -5.7573123E-04
425   7.4802875E-01 -3.7129185E-03
426   1.3783189E+00 -2.9632010E-02
427   2.5427420E+00 -1.0821454E-01
428   4.6995778E+00 -2.3593616E-01
429   8.7095624E+00 -3.2184810E-01
430   1.6203217E+01 -2.9865700E-01
431   3.0302910E+01 -1.9561991E-01
432   5.7068236E+01 -8.8370628E-02
433   1.0845021E+02 -3.0872542E-02
434   2.0847323E+02 -8.4857068E-03
435   4.0651542E+02 -1.9423015E-03
436   8.0667635E+02 -4.1900690E-04
437   1.6348107E+03 -8.3141076E-05
438   3.3969005E+03 -1.5517453E-05
439   7.2674336E+03 -4.6868130E-06
440
441
442   3.5433483E-02 -3.0184033E-08
443   6.5194589E-02  1.4123331E-07
444   1.1995578E-01 -3.6812892E-07
445   2.2073350E-01  6.5493014E-07
446   4.0626020E-01 -3.9507151E-06
447   7.4802875E-01 -3.4920504E-05
448   1.3783189E+00 -3.8022076E-04
449   2.5427420E+00 -1.8871046E-03
450   4.6995778E+00 -5.5934358E-03
451   8.7095624E+00 -1.0381556E-02
452   1.6203217E+01 -1.3122465E-02
453   3.0302910E+01 -1.1720703E-02
454   5.7068236E+01 -7.2258794E-03
455   1.0845021E+02 -3.4399092E-03
456   2.0847323E+02 -1.2821405E-03
457   4.0651542E+02 -3.9287197E-04
458   8.0667635E+02 -1.1098769E-04
459   1.6348107E+03 -2.6812267E-05
460   3.3969005E+03 -6.3265217E-06
461   7.2674336E+03 -1.6293580E-06
462
463
464                            4F+
465                 -5.7413469E+00
466
467
468   3.5433483E-02 -1.5556098E-05
469   6.5194589E-02  4.9990333E-05
470   1.1995578E-01 -1.0088117E-04
471   2.2073350E-01  1.1518582E-04
472   4.0626020E-01 -6.4369648E-04
473   7.4802875E-01 -4.2360900E-03
474   1.3783189E+00 -3.1861324E-02
475   2.5427420E+00 -1.1273317E-01
476   4.6995778E+00 -2.4091820E-01
477   8.7095624E+00 -3.2341441E-01
478   1.6203217E+01 -2.9517418E-01
479   3.0302910E+01 -1.9089221E-01
480   5.7068236E+01 -8.5166804E-02
481   1.0845021E+02 -2.9037579E-02
482   2.0847323E+02 -7.7172949E-03
483   4.0651542E+02 -1.6767727E-03
484   8.0667635E+02 -3.3097325E-04
485   1.6348107E+03 -5.4957097E-05
486   3.3969005E+03 -8.8754940E-06
487   7.2674336E+03 -1.4989268E-06
488
489
```

```
490    3.5433483E-02 -3.2617168E-08
491    6.5194589E-02  1.4149420E-07
492    1.1995578E-01 -3.8604183E-07
493    2.2073350E-01  5.9849482E-07
494    4.0626020E-01 -4.5010270E-06
495    7.4802875E-01 -4.0093122E-05
496    1.3783189E+00 -4.0951459E-04
497    2.5427420E+00 -1.9678245E-03
498    4.6995778E+00 -5.7167358E-03
499    8.7095624E+00 -1.0444036E-02
500    1.6203217E+01 -1.2993386E-02
501    3.0302910E+01 -1.1476039E-02
502    5.7068236E+01 -7.0050481E-03
503    1.0845021E+02 -3.2756335E-03
504    2.0847323E+02 -1.1959300E-03
505    4.0651542E+02 -3.5705524E-04
506    8.0667635E+02 -9.7043968E-05
507    1.6348107E+03 -2.1994873E-05
508    3.3969005E+03 -4.9229745E-06
509    7.2674336E+03 -1.0697561E-06
510
511
512  Calculated energy =   -2.0913713216718592E+04
513  Literature energy =   -2.0913714331849998E+04
514  Difference (cal - lit) = 1.12E-03
515  Relative error (abs((cal - lit)/lit)) = 5.33E-08
516  Basis set for original program
517 35 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
518    3.5433483000000002E-02 1.0d0 1.0d0
519    6.5194588593144476E-02 1.0d0 1.0d0
520    1.1995577607433897E-01 1.0d0 1.0d0
521    2.2073349756341545E-01 1.0d0 1.0d0
522    4.0626019965028504E-01 1.0d0 1.0d0
523    7.4802874797813979E-01 1.0d0 1.0d0
524    1.3783189024559244E+00 1.0d0 1.0d0
525    2.5427420408376111E+00 1.0d0 1.0d0
526    4.6995778331749465E+00 1.0d0 1.0d0
527    8.7095624280738555E+00 1.0d0 1.0d0
528    1.6203216735620309E+01 1.0d0 1.0d0
529    3.0302909733749487E+01 1.0d0 1.0d0
530    5.7068235542655763E+01 1.0d0 1.0d0
531    1.0845021407222320E+02 1.0d0 1.0d0
532    2.0847323449769911E+02 1.0d0 1.0d0
533    4.0651541537167549E+02 1.0d0 1.0d0
534    8.0667634870078655E+02 1.0d0 1.0d0
535    1.6348107232313521E+03 1.0d0 1.0d0
536    3.3969004983391187E+03 1.0d0 1.0d0
537    7.2674336199173958E+03 1.0d0 1.0d0
538    1.6080864381912534E+04 1.0d0 1.0d0
539    3.6974131657844009E+04 1.0d0 1.0d0
540    8.8762049980994911E+04 1.0d0 1.0d0
541    2.2355921269411492E+05 1.0d0 1.0d0
542    5.9356338843429903E+05 1.0d0 1.0d0
543    1.6690413606262361E+06 1.0d0 1.0d0
544    4.9925104726352403E+06 1.0d0 1.0d0
545    1.5952594555420512E+07 1.0d0 1.0d0
546    5.4660322670399554E+07 1.0d0 1.0d0
547    2.0153596325467077E+08 1.0d0 1.0d0
548    8.0207663178819859E+08 1.0d0 1.0d0
549    3.4548829687578516E+09 1.0d0 1.0d0
550    1.6143749848481022E+10 1.0d0 1.0d0
551    8.1990099053523590E+10 1.0d0 1.0d0
552    4.5329140391053540E+11 1.0d0 1.0d0
553 30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
554    3.5433483000000002E-02 1.0d0 1.0d0
555    6.5194588593144476E-02 1.0d0 1.0d0
556    1.1995577607433897E-01 1.0d0 1.0d0
557    2.2073349756341545E-01 1.0d0 1.0d0
558    4.0626019965028504E-01 1.0d0 1.0d0
559    7.4802874797813979E-01 1.0d0 1.0d0
560    1.3783189024559244E+00 1.0d0 1.0d0
561    2.5427420408376111E+00 1.0d0 1.0d0
562    4.6995778331749465E+00 1.0d0 1.0d0
563    8.7095624280738555E+00 1.0d0 1.0d0
564    1.6203216735620309E+01 1.0d0 1.0d0
565    3.0302909733749487E+01 1.0d0 1.0d0
566    5.7068235542655763E+01 1.0d0 1.0d0
567    1.0845021407222320E+02 1.0d0 1.0d0
568    2.0847323449769911E+02 1.0d0 1.0d0
569    4.0651541537167549E+02 1.0d0 1.0d0
570    8.0667634870078655E+02 1.0d0 1.0d0
571    1.6348107232313521E+03 1.0d0 1.0d0
572    3.3969004983391187E+03 1.0d0 1.0d0
573    7.2674336199173958E+03 1.0d0 1.0d0
574    1.6080864381912534E+04 1.0d0 1.0d0
575    3.6974131657844009E+04 1.0d0 1.0d0
576    8.8762049980994911E+04 1.0d0 1.0d0
577    2.2355921269411492E+05 1.0d0 1.0d0
578    5.9356338843429903E+05 1.0d0 1.0d0
579    1.6690413606262361E+06 1.0d0 1.0d0
580    4.9925104726352403E+06 1.0d0 1.0d0
581    1.5952594555420512E+07 1.0d0 1.0d0
```

```
582    5.4660322670399554E+07 1.0d0 1.0d0
583    2.0153596325467077E+08 1.0d0 1.0d0
584  30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
585    3.5433483000000002E-02 1.0d0 1.0d0
586    6.5194588593144476E-02 1.0d0 1.0d0
587    1.1995577607433897E-01 1.0d0 1.0d0
588    2.2073349756341545E-01 1.0d0 1.0d0
589    4.0626019965028504E-01 1.0d0 1.0d0
590    7.4802874797813979E-01 1.0d0 1.0d0
591    1.3783189024559244E+00 1.0d0 1.0d0
592    2.5427420408376111E+00 1.0d0 1.0d0
593    4.6995778331749465E+00 1.0d0 1.0d0
594    8.7095624280738555E+00 1.0d0 1.0d0
595    1.6203216735620309E+01 1.0d0 1.0d0
596    3.0302909733749487E+01 1.0d0 1.0d0
597    5.7068235542655763E+01 1.0d0 1.0d0
598    1.0845021407222320E+02 1.0d0 1.0d0
599    2.0847323449769911E+02 1.0d0 1.0d0
600    4.0651541537167549E+02 1.0d0 1.0d0
601    8.0667634870078655E+02 1.0d0 1.0d0
602    1.6348107232313521E+03 1.0d0 1.0d0
603    3.3969004983391187E+03 1.0d0 1.0d0
604    7.2674336199173958E+03 1.0d0 1.0d0
605    1.6080864381912534E+04 1.0d0 1.0d0
606    3.6974131657844009E+04 1.0d0 1.0d0
607    8.8762049980994911E+04 1.0d0 1.0d0
608    2.2355921269411492E+05 1.0d0 1.0d0
609    5.9356338843429903E+05 1.0d0 1.0d0
610    1.6690413606262361E+06 1.0d0 1.0d0
611    4.9925104726352403E+06 1.0d0 1.0d0
612    1.5952594555420512E+07 1.0d0 1.0d0
613    5.4660322670399554E+07 1.0d0 1.0d0
614    2.0153596325467077E+08 1.0d0 1.0d0
615  25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
616    3.5433483000000002E-02 1.0d0 1.0d0
617    6.5194588593144476E-02 1.0d0 1.0d0
618    1.1995577607433897E-01 1.0d0 1.0d0
619    2.2073349756341545E-01 1.0d0 1.0d0
620    4.0626019965028504E-01 1.0d0 1.0d0
621    7.4802874797813979E-01 1.0d0 1.0d0
622    1.3783189024559244E+00 1.0d0 1.0d0
623    2.5427420408376111E+00 1.0d0 1.0d0
624    4.6995778331749465E+00 1.0d0 1.0d0
625    8.7095624280738555E+00 1.0d0 1.0d0
626    1.6203216735620309E+01 1.0d0 1.0d0
627    3.0302909733749487E+01 1.0d0 1.0d0
628    5.7068235542655763E+01 1.0d0 1.0d0
629    1.0845021407222320E+02 1.0d0 1.0d0
630    2.0847323449769911E+02 1.0d0 1.0d0
631    4.0651541537167549E+02 1.0d0 1.0d0
632    8.0667634870078655E+02 1.0d0 1.0d0
633    1.6348107232313521E+03 1.0d0 1.0d0
634    3.3969004983391187E+03 1.0d0 1.0d0
635    7.2674336199173958E+03 1.0d0 1.0d0
636    1.6080864381912534E+04 1.0d0 1.0d0
637    3.6974131657844009E+04 1.0d0 1.0d0
638    8.8762049980994911E+04 1.0d0 1.0d0
639    2.2355921269411492E+05 1.0d0 1.0d0
640    5.9356338843429903E+05 1.0d0 1.0d0
641  25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
642    3.5433483000000002E-02 1.0d0 1.0d0
643    6.5194588593144476E-02 1.0d0 1.0d0
644    1.1995577607433897E-01 1.0d0 1.0d0
645    2.2073349756341545E-01 1.0d0 1.0d0
646    4.0626019965028504E-01 1.0d0 1.0d0
647    7.4802874797813979E-01 1.0d0 1.0d0
648    1.3783189024559244E+00 1.0d0 1.0d0
649    2.5427420408376111E+00 1.0d0 1.0d0
650    4.6995778331749465E+00 1.0d0 1.0d0
651    8.7095624280738555E+00 1.0d0 1.0d0
652    1.6203216735620309E+01 1.0d0 1.0d0
653    3.0302909733749487E+01 1.0d0 1.0d0
654    5.7068235542655763E+01 1.0d0 1.0d0
655    1.0845021407222320E+02 1.0d0 1.0d0
656    2.0847323449769911E+02 1.0d0 1.0d0
657    4.0651541537167549E+02 1.0d0 1.0d0
658    8.0667634870078655E+02 1.0d0 1.0d0
659    1.6348107232313521E+03 1.0d0 1.0d0
660    3.3969004983391187E+03 1.0d0 1.0d0
661    7.2674336199173958E+03 1.0d0 1.0d0
662    1.6080864381912534E+04 1.0d0 1.0d0
663    3.6974131657844009E+04 1.0d0 1.0d0
664    8.8762049980994911E+04 1.0d0 1.0d0
665    2.2355921269411492E+05 1.0d0 1.0d0
666    5.9356338843429903E+05 1.0d0 1.0d0
667  20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
668    3.5433483000000002E-02 1.0d0 1.0d0
669    6.5194588593144476E-02 1.0d0 1.0d0
670    1.1995577607433897E-01 1.0d0 1.0d0
671    2.2073349756341545E-01 1.0d0 1.0d0
672    4.0626019965028504E-01 1.0d0 1.0d0
673    7.4802874797813979E-01 1.0d0 1.0d0
```

```
674    1.3783189024559244E+00 1.0d0 1.0d0
675    2.5427420408376111E+00 1.0d0 1.0d0
676    4.6995778331749465E+00 1.0d0 1.0d0
677    8.7095624280738555E+00 1.0d0 1.0d0
678    1.6203216735620309E+01 1.0d0 1.0d0
679    3.0302909733749487E+01 1.0d0 1.0d0
680    5.7068235542655763E+01 1.0d0 1.0d0
681    1.0845021407222320E+02 1.0d0 1.0d0
682    2.0847323449769911E+02 1.0d0 1.0d0
683    4.0651541537167549E+02 1.0d0 1.0d0
684    8.0667634870078655E+02 1.0d0 1.0d0
685    1.6348107232313521E+03 1.0d0 1.0d0
686    3.3969004983391187E+03 1.0d0 1.0d0
687    7.2674336199173958E+03 1.0d0 1.0d0
688 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
689    3.5433483000000002E-02 1.0d0 1.0d0
690    6.5194588593144476E-02 1.0d0 1.0d0
691    1.1995577607433897E-01 1.0d0 1.0d0
692    2.2073349756341545E-01 1.0d0 1.0d0
693    4.0626019965028504E-01 1.0d0 1.0d0
694    7.4802874797813979E-01 1.0d0 1.0d0
695    1.3783189024559244E+00 1.0d0 1.0d0
696    2.5427420408376111E+00 1.0d0 1.0d0
697    4.6995778331749465E+00 1.0d0 1.0d0
698    8.7095624280738555E+00 1.0d0 1.0d0
699    1.6203216735620309E+01 1.0d0 1.0d0
700    3.0302909733749487E+01 1.0d0 1.0d0
701    5.7068235542655763E+01 1.0d0 1.0d0
702    1.0845021407222320E+02 1.0d0 1.0d0
703    2.0847323449769911E+02 1.0d0 1.0d0
704    4.0651541537167549E+02 1.0d0 1.0d0
705    8.0667634870078655E+02 1.0d0 1.0d0
706    1.6348107232313521E+03 1.0d0 1.0d0
707    3.3969004983391187E+03 1.0d0 1.0d0
708    7.2674336199173958E+03 1.0d0 1.0d0
```

Listing A.2: *Output file for calculating the single point energy of Pb.*

A.2 Optimizing a Basis Set for a Specific Electron Configuration

This input file will optimize a WTBS basis set for protactinium with a specific electron configuration. Note that in the output file, the "freeel" variable is set to 1. "freeel" only matters if "autogen" is set to true, the actual number of electrons in open shells is determined from the electron configurations that are read in.

```
1  91Pa
2   $contrl jobtype='bsopt' $end
3   $nuc znuc=91 nucmdl=3 $end
4   $bas nsym=7  bastype='wtbs' $end
5  $wtbs wtbspara=1.9453737E-02 1.8578241E+00 4.9471334E+00 1.4389341E+00
6         nbs=35 30 30 25 25 20 20
7         start=1 1 1 1 1 1 1 $end
8   $newuoa rhobeg=1.0e-1 rhoend=1.0e-5 iprint=2 maxfun=2000 $end
9   $econfig nclose=7 5 5 3 3 1 1 nopen=0 0 0 1 1 1 1 nconf=6 $end
10   0.0 0.0 0.0 1.0 0.0 0.0 2.0
11   0.0 0.0 0.0 1.0 0.0 1.0 1.0
12   0.0 0.0 0.0 1.0 0.0 2.0 0.0
13   0.0 0.0 0.0 0.0 1.0 0.0 2.0
14   0.0 0.0 0.0 0.0 1.0 1.0 1.0
15   0.0 0.0 0.0 0.0 1.0 2.0 0.0
16   $scf maxitr=100 ixtrp=1 dfctr=0.5 $end
```

Listing A.3: *Input file for Pa basis set optimization calculation.*

```
1  &CONTRL
2  JOBTYPE = bsopt ,
3  C =    137.0359997600000
4  /
5  &NUC
6  ZNUC =    91.00000000000000      ,
7  NUCMDL =            3,
8  RNUC =   0.000000000000000      ,
9  ALPHA =    129305395.1200000
10  /
11  &BAS
12  NSYM =             7,
13  BASTYPE = wtbs ,
```

```
14 | NGROUP =            1
15 | /
16 | &WTBS
17 | WTBSPARA =   1.9453736999999999E-002,
18 |                1.857824100000000       ,
19 |                4.947133400000000       ,
20 |                1.438934100000000       ,
21 | NBS =           35,
22 |                 30,
23 |                 30,
24 |                 25,
25 |                 25,
26 |                 20,
27 |                 20,
28 | START =          1,
29 |                  1,
30 |                  1,
31 |                  1,
32 |                  1,
33 |                  1,
34 |                  1,
35 | GROUPS =         1,
36 |                  1,
37 |                  1,
38 |                  1,
39 |                  1,
40 |                  1,
41 |                  1
42 | /
43 | &NEWUOA
44 | RHOBEG =   0.1000000000000000       ,
45 | RHOEND =   1.0000000000000001E-005,
46 | IPRINT =            2,
47 | MAXFUN =         2000
48 | /
49 | &ECONFIG
50 | NCLOSE =            7,
51 |                     5,
52 |                     5,
53 |                     3,
54 |                     3,
55 |                     1,
56 |                     1,
57 | NOPEN =             0,
58 |                     0,
59 |                     0,
60 |                     1,
61 |                     1,
62 |                     1,
63 |                     1,
64 | FREEEL =            1,
65 | AUTOGEN =  F,
66 | NCONF =             6
67 | /
68 | nconf = 6
69 |  1  0.00  0.00  0.00  1.00  0.00  0.00  2.00
70 |  2  0.00  0.00  0.00  1.00  0.00  1.00  1.00
71 |  3  0.00  0.00  0.00  1.00  0.00  2.00  0.00
72 |  4  0.00  0.00  0.00  0.00  1.00  0.00  2.00
73 |  5  0.00  0.00  0.00  0.00  1.00  1.00  1.00
74 |  6  0.00  0.00  0.00  0.00  1.00  2.00  0.00
75 | &SCF
76 | MAXITR =          100,
77 | IXTRP =             1,
78 | DFCTR =   0.5000000000000000       ,
79 | THDLL =   1.0000000000000001E-005,
80 | THDSL =   9.9999999999999995E-008,
81 | THDSS =   1.0000000000000001E-009
82 | /
83 |        31560
84 |     166068720
85 |        82680
86 |        82680
87 |     221424960
88 | Calculating one electron integrals
89 | Calculating two electron integrals
90 | Calculating guess
91 | Starting SCF
92 | SCF converged at iteration 38 Final energy =     -27274.37615333940
93 |
94 |
95 |                      1S+          2S+          3S+          4S+          5S+          6S+          7S+
96 |              -4.1678558E+03 -7.8202807E+02 -1.9970550E+02 -5.2156121E+01 -1.1997891E+01 -2.0501109E+00 -1.9881466E-01
97 |
98 |
99 |  1.9453737E-02  4.5526064E-07  6.1600046E-06 -6.1423490E-05  4.8451991E-05 -1.0511245E-04  1.3158903E-04 -1.8724908E-01
100 | 3.6141658E-02 -1.9479713E-06 -2.6351782E-05  2.6251028E-04 -2.0608864E-04  4.3702976E-04 -4.8933088E-04 -5.0769082E-01
101 | 6.7145353E-02  4.6364451E-06  6.2698403E-05 -6.2360865E-04  4.8527658E-04 -9.8904771E-04  2.4811733E-03 -4.9371359E-01
102 | 1.2474818E-01 -8.4540743E-06 -1.1426442E-04  1.1332041E-03 -8.7060948E-04  1.6324965E-03  1.8661239E-02 -1.0195880E-01
103 | 2.3178217E-01  1.3450491E-05  1.8159861E-04 -1.7957389E-03  1.3273407E-03 -2.3445935E-03  2.7533799E-01  2.9432366E-01
104 | 4.3071120E-01 -1.9941795E-05 -2.6910486E-04  2.6311082E-03 -1.9276497E-03  1.1089307E-03  7.8850453E-01  4.5209924E-01
105 | 8.0058681E-01  2.8321063E-05  3.8110568E-04 -3.7093835E-03  2.3287542E-03 -3.6107270E-02  5.0398773E-01  2.2648026E-01
```

```
106   1.4887931E+00 -3.9289907E-05 -5.2704055E-04  5.0770781E-03 -1.9095251E-03 -4.6247606E-01 -6.1984690E-01 -2.3106351E-01
107   2.7707233E+00  5.4127358E-05  7.3096927E-04 -6.5063021E-03  1.0470406E-02 -1.0705943E+00 -9.6963877E-01 -3.4091033E-01
108   5.1625848E+00 -7.1730547E-05 -9.5665668E-04  8.5020340E-03  2.3278340E-01  1.2248776E-01  4.3766720E-02  1.5356681E-02
109   9.6361639E+00  9.2150385E-05  1.1905804E-03 -1.4907331E-02  9.8537025E-01  1.2755242E+00  7.2196144E-01  2.2937092E-01
110   1.8031447E+01 -1.2623899E-04 -1.7792297E-03 -1.1538348E-01  3.8510740E-01  3.3727265E-01  1.7145582E-01  5.4245805E-02
111   3.3858337E+01  1.5373486E-04  2.6018880E-03 -6.9733818E-01 -9.8256369E-01 -6.2697726E-01 -2.9682747E-01 -9.2176854E-02
112   6.3875692E+01 -1.1535749E-04  3.8093540E-02 -6.0473604E-01 -6.4464917E-01 -3.9125068E-01 -1.8064621E-01 -5.5605763E-02
113   1.2125177E+02  1.2587653E-04  3.1220205E-01  3.3102664E-01  2.2711991E-01  1.2336342E-01  5.5360335E-02  1.6941375E-02
114   2.3200920E+02 -1.4331244E-03  5.2657623E-01  5.2989905E-01  3.2983813E-01  1.6974036E-01  7.5889627E-02  2.3305591E-02
115   4.4845175E+02 -3.5900271E-02  2.6241533E-01  2.1315943E-01  1.2658256E-01  6.4910423E-02  2.8848768E-02  8.8309114E-03
116   8.7781809E+02 -1.6237035E-01 -5.4377180E-02 -2.1340131E-02 -1.0567654E-02 -5.2701286E-03 -2.2994560E-03 -6.9793578E-04
117   1.7450942E+03 -2.6616678E-01 -1.5722966E-01 -8.1885030E-02 -4.2968044E-02 -2.1145552E-02 -9.3451415E-03 -2.8633384E-03
118   3.5348455E+03 -2.5104786E-01 -1.3570153E-01 -6.9567552E-02 -3.6553325E-02 -1.8070096E-02 -7.9768505E-03 -2.4411779E-03
119   7.3220404E+03 -1.7657870E-01 -8.7651162E-02 -4.4131448E-02 -2.3016579E-02 -1.1330730E-02 -5.0026903E-03 -1.5318434E-03
120   1.5571453E+04 -1.0699572E-01 -5.0377636E-02 -2.5130416E-02 -1.3098900E-02 -6.4564612E-03 -2.8493119E-03 -8.7210360E-04
121   3.4144729E+04 -6.0291269E-02 -2.7398442E-02 -1.3574384E-02 -7.0546649E-03 -3.4711606E-03 -1.5320152E-03 -4.6902952E-04
122   7.7552073E+04 -3.2990939E-02 -1.4620144E-02 -7.2147644E-03 -3.7489374E-03 -1.8457668E-03 -8.1446829E-04 -2.4930193E-04
123   1.8331766E+05 -1.7803523E-02 -7.7317238E-03 -3.8017386E-03 -1.9725466E-03 -9.7034370E-04 -4.2819453E-04 -1.3108227E-04
124   4.5318823E+05 -9.5990017E-03 -4.1018510E-03 -2.0118317E-03 -1.0435985E-03 -5.1349929E-04 -2.2657310E-04 -6.9353811E-05
125   1.1774969E+06 -5.0914462E-03 -2.1481685E-03 -1.0512710E-03 -5.4487705E-04 -2.6798839E-04 -1.1824674E-04 -3.6197166E-05
126   3.2312890E+06 -2.6779367E-03 -1.1191252E-03 -5.4684284E-04 -2.8337545E-04 -1.3938706E-04 -6.1499403E-05 -1.8825057E-05
127   9.4102964E+06 -1.3140471E-03 -5.4552441E-04 -2.6624977E-04 -1.3791178E-04 -6.7820854E-05 -2.9923603E-05 -9.1599121E-06
128   2.9216892E+07 -6.0990887E-04 -2.5203190E-04 -1.2291586E-04 -6.3662896E-05 -3.1309573E-05 -1.3813841E-05 -4.2284427E-06
129   9.7128202E+07 -1.6932067E-04 -6.9765935E-05 -3.4006601E-05 -1.7609042E-05 -8.6588446E-06 -3.8203370E-06 -1.1694392E-06
130   3.4711732E+08  1.0855661E-06  4.4974166E-07  2.1919545E-07  1.1319830E-07  5.5504153E-08  2.4499931E-08  7.5040942E-09
131   1.3384720E+08 -4.9489131E-07 -2.0408921E-07 -9.9444185E-08 -5.1421172E-08 -2.5249045E-08 -1.1142486E-08 -3.4118104E-09
132   5.5867382E+09  9.2667492E-08  3.8202503E-08  1.8613396E-08  9.6245329E-09  4.7258361E-09  2.0855266E-09  6.3858531E-10
133   2.5314129E+10 -1.2866632E-08 -5.3038951E-09 -2.5841696E-09 -1.3361905E-09 -6.5608747E-10 -2.8953396E-10 -8.8655105E-11
134
135
136   1.9453737E-02 -5.0363706E-09  4.1816344E-09 -7.1130274E-08  3.2626997E-08 -1.0191264E-07  1.1005684E-07 -1.6505317E-04
137   3.6141658E-02  1.8534158E-08 -2.7073422E-08  3.8148473E-07 -2.0882476E-07  5.6083489E-07 -5.6502652E-07 -6.0995107E-04
138   6.7145353E-02 -3.8743877E-08  9.3105993E-08 -1.1705575E-06  7.0835292E-07 -1.6996928E-06  4.0121203E-06 -8.0851446E-04
139   1.2474818E-01  6.2211854E-08 -2.3973054E-07  2.7968485E-06 -1.7915559E-06  3.7853494E-06  4.1745601E-05 -2.2754793E-04
140   2.3178217E-01 -8.5290474E-08  5.3161181E-07 -5.8924309E-06  3.8006383E-06 -7.3563981E-06  8.3761902E-04  8.9545248E-04
141   4.3071120E-01  1.0208035E-07 -1.0912541E-06  1.1564757E-05 -7.6428230E-06  4.9288760E-06  3.2706490E-03  1.8751516E-03
142   8.0058681E-01 -1.0113876E-07  2.1306620E-06 -2.1943188E-05  1.2659057E-05 -2.0469811E-04  2.8494050E-03  1.2804352E-03
143   1.4887931E+00  5.7963374E-08 -4.0508702E-06  4.0571690E-05 -1.3989995E-03 -3.5664257E-03 -4.7796997E-03 -1.7817799E-03
144   2.7707233E+00  8.0268709E-08  7.7097835E-06 -7.0519182E-05  1.0916315E-04 -1.1264127E-02 -1.0199493E-02 -3.5858463E-03
145   5.1625848E+00 -3.7857508E-07 -1.3828225E-05  1.2511857E-04  3.3482341E-03  1.7666122E-03  6.3456555E-04  2.2254868E-04
146   9.6361639E+00  9.5854938E-07  2.3593188E-05 -2.9713533E-04  1.9342317E-02  2.5017145E-02  1.4157942E-02  4.4979590E-03
147   1.8031447E+01 -2.3080850E-06 -4.8277108E-05 -3.1055434E-03  1.0304204E-02  9.0112601E-03  4.5792045E-03  1.4487425E-03
148   3.3858337E+01  4.2624324E-06  9.6726112E-05 -2.5736242E-02 -3.6158839E-02 -2.3065390E-02 -1.0918611E-02 -3.3905903E-03
149   6.3875692E+01 -3.9222807E-06  1.9604252E-03 -3.0450329E-02 -3.2337647E-02 -1.9610198E-02 -9.0525363E-03 -2.7863962E-03
150   1.2125177E+02  6.0105809E-06  2.2053270E-02  2.3275860E-02  1.5984258E-02  8.6821874E-03  3.8967421E-03  1.1925690E-03
151   2.3200920E+02 -1.4693522E-04  5.0989297E-02  5.0595306E-02  3.1409317E-02  1.6152076E-02  7.2202503E-03  2.2172770E-03
152   4.4845175E+02 -5.3310399E-03  3.4317940E-02  2.7410756E-02  1.6222173E-02  8.3130239E-03  3.6937536E-03  1.1306051E-03
153   8.7781809E+02 -3.3138881E-02 -1.1199777E-02 -4.7571173E-03 -2.4170697E-03 -1.2082964E-03 -5.2904485E-04 -1.6088394E-04
154   1.7450942E+03 -7.4346622E-02 -4.0725194E-02 -2.1062736E-02 -1.1035434E-02 -5.4284824E-03 -2.3988415E-03 -7.3499143E-04
155   3.5348455E+03 -9.4791276E-02 -4.7154793E-02 -2.3907837E-02 -1.2529869E-02 -6.1904548E-03 -2.7322183E-03 -8.3610751E-04
156   7.3220404E+03 -8.8657364E-02 -4.0506359E-02 -2.0147306E-02 -1.0475076E-02 -5.1518739E-03 -2.2742441E-03 -6.9638073E-04
157   1.5571453E+04 -6.9734773E-02 -3.0375011E-02 -1.4973564E-02 -7.7832300E-03 -3.8340378E-03 -1.6916478E-03 -5.1773708E-04
158   3.4144729E+04 -4.9557458E-02 -2.1009571E-02 -1.0294910E-02 -5.3347585E-03 -2.6224758E-03 -1.1572758E-03 -3.5430612E-04
159   7.7552073E+04 -3.2965670E-02 -1.3787754E-02 -6.7413567E-03 -3.4955699E-03 -1.7203202E-03 -7.5897747E-04 -2.3230106E-04
160   1.8331766E+05 -2.0881919E-02 -8.6643540E-03 -4.2282845E-03 -2.1893129E-03 -1.0762097E-03 -4.7486954E-04 -1.4537414E-04
161   4.5318823E+05 -1.2669461E-02 -5.2369602E-03 -2.5546250E-03 -1.3236088E-03 -6.5117130E-04 -2.8728442E-04 -8.7932535E-05
162   1.1774969E+06 -7.3479864E-03 -3.0292815E-03 -1.4766257E-03 -7.6444915E-04 -3.7581084E-04 -1.6581658E-04 -5.0760481E-05
163   3.2312890E+06 -4.0327727E-03 -1.6606011E-03 -8.0945657E-04 -4.1927555E-04 -2.0623615E-04 -9.0988010E-05 -2.7850367E-05
164   9.4102964E+06 -2.0258641E-03 -8.3326185E-04 -4.0601852E-04 -2.1018983E-04 -1.0333633E-04 -4.5593595E-05 -1.3957078E-05
165   2.9216892E+07 -8.4844613E-04 -3.4885394E-04 -1.7000306E-04 -8.8049493E-05 -4.3308889E-05 -1.9107141E-05 -5.8484891E-06
166   9.7128202E+07 -1.5105779E-04 -6.2100174E-05 -3.0252716E-05 -1.5655586E-05 -7.6940850E-06 -3.3949229E-06 -1.0393242E-06
167   3.4711732E+08  1.1704689E-05  4.8007844E-06  2.3357655E-06  1.2056324E-06  5.9103384E-07  2.6088264E-07  7.9906891E-08
168   1.3384720E+08 -3.9489441E-06 -1.6195221E-06 -7.8782411E-07 -4.0646724E-07 -1.9917466E-07 -8.7921529E-08 -2.6932237E-08
169   5.5867382E+09  1.3761821E-06  5.6435969E-07  2.7452198E-07  1.4162013E-07  6.9388206E-08  3.0630487E-08  9.3829787E-09
170   2.5314129E+10 -3.9881149E-07 -1.6354648E-07 -7.9553252E-08 -4.1038958E-08 -2.0107018E-08 -8.8759999E-09 -2.7189798E-09
171
172
173                    2P-            3P-            4P-            5P-            6P-
174             -7.5278375E+02 -1.8645886E+02 -4.6168113E+01 -9.6061333E+00 -1.2849029E+00
175
176
177   1.9453737E-02  9.0965057E-07  1.2896703E-05 -4.3994948E-05 -2.0416597E-05 -2.6310064E-04
178   3.6141658E-02 -3.5075358E-06 -4.9693762E-05  1.6870164E-04  7.1672253E-05 -5.6527686E-05
179   6.7145353E-02  7.8332869E-06  1.1075391E-04 -3.7507757E-04 -2.0535337E-04 -6.1665581E-03
180   1.2474818E-01 -1.3812446E-05 -1.9489646E-04  6.5040608E-04  2.7390373E-04 -3.6031104E-02
181   2.3178217E-01  2.1716136E-05  3.0504688E-04 -1.0047874E-03 -4.9867871E-04 -2.4163851E-01
182   4.3071120E-01 -3.2242716E-05 -4.4888289E-04  1.4554968E-03  2.6902937E-03 -5.3380006E-01
183   8.0058681E-01  4.6495084E-05  6.3916964E-04 -1.9025030E-03 -2.9228815E-02 -4.3202646E-01
184   1.4887931E+00 -6.4267822E-05 -9.0331287E-04  2.3062452E-03  2.8313912E-01  1.3477329E-01
185   2.7707233E+00  9.7062275E-05  1.1336035E-03 -7.9029460E-03  6.8725142E-01  4.8905387E-01
186   5.1625848E+00 -1.2442494E-04 -1.5944229E-03 -1.1019740E-01  3.1341941E-01  2.1541172E-01
187   9.6361639E+00  1.4800346E-04  3.8354015E-03 -5.4657109E-01 -4.8407881E-01 -2.2646971E-01
188   1.8031447E+01 -5.4346015E-04  4.5783363E-02 -5.6337875E-01 -4.5481993E-01 -2.1160347E-01
189   3.3858337E+01  4.9628044E-04  3.2042697E-01  1.3351790E-01  4.2390518E-02  1.5892391E-02
190   6.3875692E+01  1.0051412E-02  5.3506410E-01  4.6050660E-01  2.4979948E-01  1.0529281E-01
191   1.2125177E+02  1.0530040E-01  2.2896636E-01  1.9198472E-01  1.0300065E-01  4.3646851E-02
192   2.3200920E+02  2.7676550E-01 -1.0764924E-01 -4.5965987E-02 -2.0689177E-02 -8.4760406E-03
193   4.4845175E+02  3.2207155E-01 -1.8629447E-01 -9.9267957E-02 -4.7649979E-02 -1.9600133E-02
194   8.7781809E+02  2.3379262E-01 -1.3364107E-01 -7.0874802E-02 -3.4056843E-02 -1.4050746E-02
195   1.7450942E+03  1.3377488E-01 -7.3952268E-02 -3.8920165E-02 -1.8638224E-02 -7.6700437E-03
196   3.5348455E+03  6.9162687E-02 -3.7313961E-02 -1.9500814E-02 -9.3318162E-03 -3.8449565E-03
197   7.3220404E+03  3.5186458E-02 -1.8660812E-02 -9.7170006E-03 -4.6416309E-03 -1.9096388E-03
```

```
198   1.5571453E+04   1.8505689E-02  -9.7166701E-03  -5.0445737E-03  -2.4093916E-03  -9.9205398E-04
199   3.4144729E+04   1.0053157E-02  -5.2431315E-03  -2.7184682E-03  -1.2973726E-03  -5.3378266E-04
200   7.7552073E+04   5.7074514E-03  -2.9660586E-03  -1.5361338E-03  -7.3311748E-04  -3.0174582E-04
201   1.8331766E+05   3.2157856E-03  -1.6675210E-03  -8.6327145E-04  -4.1187645E-04  -1.6947378E-04
202   4.5318823E+05   1.8747197E-03  -9.7110390E-04  -5.0256289E-04  -2.3978410E-04  -9.8678508E-05
203   1.1774969E+06   1.0050904E-03  -5.2031431E-04  -2.6924530E-04  -1.2845081E-04  -5.2855503E-05
204   3.2312890E+06   5.8000988E-04  -3.0016914E-04  -1.5531049E-04  -7.4096328E-05  -3.0491384E-05
205   9.4102964E+06   2.4716101E-04  -1.2788647E-04  -6.6168171E-05  -3.1566674E-05  -1.2989360E-05
206   2.9216892E+07   1.4083690E-04  -7.2866088E-05  -3.7699289E-05  -1.7985330E-05  -7.4010218E-06
207
208
209   1.9453737E-02  -6.3405875E-06  -2.1301527E-08   1.0700958E-06   1.1834573E-07  -1.4440979E-07
210   3.6141658E-02   3.1756298E-06  -6.2079715E-08  -3.0754875E-07   2.5434252E-08  -1.7462909E-07
211   6.7145353E-02  -1.0272372E-05   1.8208140E-07   1.0384013E-06  -1.7584224E-07  -1.2767581E-05
212   1.2474818E-01  -2.8077997E-06  -5.9212891E-07   2.3316708E-06   7.8600651E-07  -1.0380398E-04
213   2.3178217E-01  -1.4868391E-05   1.1352799E-06  -1.2643904E-06  -1.5375906E-06  -9.4873522E-04
214   4.3071120E-01  -1.6397545E-05  -2.5374135E-06   1.0584723E-05   1.4608141E-05  -2.8579440E-03
215   8.0058681E-01  -3.0646694E-05   4.5490830E-06  -8.3293775E-06   2.1429967E-04  -3.1528512E-03
216   1.4887931E+00  -4.6806616E-05  -9.3651340E-06   3.0953812E-05   2.8198685E-03   1.3426030E-03
217   2.7707233E+00  -7.4042998E-05   1.5121921E-05  -9.3897732E-05   9.3362928E-03   6.6422439E-03
218   5.1625848E+00  -1.2240603E-04  -3.0538176E-05  -2.0241203E-03   5.8098961E-03   3.9909902E-03
219   9.6361639E+00  -1.8777740E-04   9.6634033E-05  -1.3818484E-02  -1.2248394E-02  -5.7314004E-03
220   1.8031447E+01  -3.2211346E-04   1.5905883E-03  -1.9458426E-02  -1.5710492E-02  -7.3093403E-03
221   3.3858337E+01  -4.7314779E-04   1.5261706E-02   6.4268708E-03   2.0740600E-03   7.8182821E-04
222   6.3875692E+01  -1.4529518E-04   3.4917787E-02   2.9967022E-02   1.6249261E-02   6.8487906E-03
223   1.2125177E+02   8.2853578E-03   2.0507632E-02   1.7060952E-02   9.1405048E-03   3.8723373E-03
224   2.3200920E+02   3.2555735E-02  -1.2945668E-02  -5.5915260E-03  -2.5270209E-03  -1.0363673E-03
225   4.4845175E+02   5.2184871E-02  -3.0492878E-02  -1.6290141E-02  -7.8244849E-03  -3.2187218E-03
226   8.7781809E+02   5.0599337E-02  -2.9106109E-02  -1.5456043E-02  -7.4297312E-03  -3.0657907E-03
227   1.7450942E+03   3.7447604E-02  -2.0808041E-02  -1.0964739E-02  -5.2520017E-03  -2.1611221E-03
228   3.5348455E+03   2.3852880E-02  -1.2946615E-02  -6.7744329E-03  -3.2432913E-03  -1.3367386E-03
229   7.3220404E+03   1.4124865E-02  -7.5432175E-03  -3.9350911E-03  -1.8801817E-03  -7.7334285E-04
230   1.5571453E+04   8.0835488E-03  -4.2798586E-03  -2.2258142E-03  -1.0638691E-03  -4.3828285E-04
231   3.4144729E+04   4.5718285E-03  -2.4029105E-03  -1.2485434E-03  -5.9598784E-04  -2.4511095E-04
232   7.7552073E+04   2.5752205E-03  -1.3482502E-03  -6.9931356E-04  -3.3399060E-04  -1.3755417E-04
233   1.8331766E+05   1.4456696E-03  -7.5366895E-04  -3.9079113E-04  -1.8646403E-04  -7.6690789E-05
234   4.5318823E+05   8.0607364E-04  -4.1939263E-04  -2.1722188E-04  -1.0369475E-04  -4.2695616E-05
235   1.1774969E+06   4.3778461E-04  -2.2721593E-04  -1.1767658E-04  -5.6138179E-05  -2.3091594E-05
236   3.2312890E+06   2.3188841E-04  -1.2025115E-04  -6.2238374E-05  -2.9702038E-05  -1.2227164E-05
237   9.4102964E+06   1.0715191E-04  -5.5493574E-05  -2.8723429E-05  -1.3701658E-05  -5.6364792E-06
238   2.9216892E+07   4.8678287E-05  -2.5204121E-05  -1.3040938E-05  -6.2224697E-06  -2.5611349E-06
239
240
241                        2P+            3P+            4P+            5P+            6P+
242                   -6.1939659E+02  -1.5549253E+02  -3.8038859E+01  -7.7228257E+00  -9.5478329E-01
243
244
245   1.9453737E-02   1.7909768E-06  -1.7357814E-05   1.1478498E-05   2.3943494E-05   2.9857062E-04
246   3.6141658E-02  -6.9040715E-06   6.6855639E-05  -4.3412352E-05  -7.7232160E-05   1.4984484E-03
247   6.7145353E-02   1.5412243E-05  -1.4884694E-04   9.7117396E-05   2.4491676E-04   1.5742169E-02
248   1.2474818E-01  -2.7149733E-05   2.6157013E-04  -1.5994485E-04  -2.4121140E-04   1.0098719E-01
249   2.3178217E-01   4.2620264E-05  -4.0809632E-04   2.4000122E-04   3.2527832E-04   3.6830260E-01
250   4.3071120E-01  -6.3065591E-05   5.9706844E-04  -3.3728483E-04  -5.5370295E-03   5.3672586E-01
251   8.0058681E-01   9.0321565E-05  -8.4149659E-04   2.6508045E-04  -7.4346300E-02   2.4351525E-01
252   1.4887931E+00  -1.2406804E-04   1.1748943E-03   3.8248978E-04  -4.2661051E-01  -2.9947901E-01
253   2.7707233E+00   1.8019397E-04  -1.3762562E-03   1.6246402E-02  -6.8213430E-01  -4.3831897E-01
254   5.1625848E+00  -2.2496858E-04   1.8294319E-03   2.1741344E-01  -4.4410348E-02  -4.9284966E-02
255   9.6361639E+00   2.4499503E-04  -7.7437667E-03   6.5460596E-01   5.9530979E-01   2.6628797E-01
256   1.8031447E+01  -4.6779083E-04  -1.0330252E-01   3.8926786E-01   3.0640830E-01   1.3425171E-01
257   3.3858337E+01   1.0638953E-03  -4.4550532E-01  -3.4215046E-01  -1.7300060E-01  -6.9214175E-02
258   6.3875692E+01   2.9507066E-02  -5.0175633E-01  -4.3575268E-01  -2.3448067E-01  -9.4050509E-02
259   1.2125177E+02   1.8226800E-01  -6.9382970E-02  -7.7621230E-02  -4.2661336E-02  -1.7339228E-02
260   2.3200920E+02   3.4067653E-01   1.9950681E-01   1.0498458E-01   4.9759817E-02   1.9538359E-02
261   4.4845175E+02   3.0831828E-01   1.9261829E-01   1.0475926E-01   5.0149212E-02   1.9658628E-02
262   8.7781809E+02   1.7951817E-01   1.0697797E-01   5.7260854E-02   2.7288808E-02   1.0708997E-02
263   1.7450942E+03   8.1558104E-02   4.6938340E-02   2.4943775E-02   1.1869221E-02   4.6482124E-03
264   3.5348455E+03   3.1965643E-02   1.7965602E-02   9.4703531E-03   4.4956032E-03   1.7627453E-03
265   7.3220404E+03   1.1545310E-02   6.4093369E-03   3.3749812E-03   1.6022521E-03   6.2708462E-04
266   1.5571453E+04   3.9616034E-03   2.1806178E-03   1.1431679E-03   5.4188309E-04   2.1244923E-04
267   3.4144729E+04   1.3204201E-03   7.2399824E-04   3.8013189E-04   1.8034013E-04   7.0555585E-05
268   7.7552073E+04   4.3056878E-04   2.3498357E-04   1.2289346E-04   5.8215787E-05   2.2825835E-05
269   1.8331766E+05   1.3832870E-04   7.5368358E-05   3.9519284E-05   1.8743120E-05   7.3308702E-06
270   4.5318823E+05   4.3643664E-05   2.3691871E-05   1.2370523E-05   5.8573101E-06   2.2969541E-06
271   1.1774969E+06   1.3350438E-05   7.2447862E-06   3.7964614E-06   1.8004317E-06   7.0393529E-07
272   3.2312890E+06   4.0383316E-06   2.1845507E-06   1.1390236E-06   5.3906370E-07   2.1147717E-07
273   9.4102964E+06   1.0537445E-06   5.7076680E-07   2.9927507E-07   1.4197777E-07   5.5462585E-08
274   2.9216892E+07   3.6303539E-07   1.9603153E-07   1.0213144E-07   4.8322849E-08   1.8961593E-08
275
276
277   1.9453737E-02   8.5985209E-10  -2.8531377E-08   6.7111681E-09   2.2295759E-08   3.3737755E-07
278   3.6141658E-02  -6.6001057E-09   1.3488641E-07  -4.4891603E-08  -1.0228987E-07   2.3326122E-06
279   6.7145353E-02   2.4004173E-08  -3.8144270E-07   1.5760424E-07   4.8058133E-07   3.3263312E-05
280   1.2474818E-01  -6.3775406E-08   8.6999191E-07  -3.7850697E-07  -6.3081234E-07   2.9104377E-04
281   2.3178217E-01   1.4544617E-07  -1.7872119E-06   8.1338376E-07   1.1769234E-06   1.4466308E-03
282   4.3071120E-01  -3.0617935E-07   3.4765381E-06  -1.6118020E-06  -2.9500368E-05   2.8739407E-03
283   8.0058681E-01   6.1585337E-07  -6.5587616E-06   1.6483973E-06  -5.4304212E-04   1.7774231E-03
284   1.4887931E+00  -1.1769394E-06   1.2308622E-05   4.2310532E-06  -4.2471669E-03  -2.9813022E-03
285   2.7707233E+00   2.3724256E-06  -1.9586095E-05   2.2018911E-04  -9.2641796E-03  -5.9520048E-03
286   5.1625848E+00  -4.0748728E-06   3.5224260E-05   4.0345279E-03  -8.1852768E-04  -9.1076808E-04
287   9.6361639E+00   6.0844709E-06  -1.9845084E-04   1.6583775E-02   1.5073627E-02   6.7421216E-03
288   1.8031447E+01  -1.6058505E-05  -3.5882024E-03   1.3464819E-02   1.0591992E-02   4.6402525E-03
289   3.3858337E+01   5.0629269E-05  -2.1201098E-02  -1.6269013E-02  -8.2292556E-03  -3.2926206E-03
```

96

```
290   6.3875692E+01  1.9519698E-03 -3.2659281E-02 -2.8297397E-02 -1.5219819E-02 -6.1041034E-03
291   1.2125177E+02  1.6551983E-02 -6.0201350E-03 -6.7830986E-03 -3.7297295E-03 -1.5162420E-03
292   2.3200920E+02  4.2504697E-02  2.4747700E-02  1.3022969E-02  6.1730569E-03  2.4238896E-03
293   4.4845175E+02  5.2751818E-02  3.2633957E-02  1.7711917E-02  8.4747501E-03  3.3217453E-03
294   8.7781809E+02  4.1993602E-02  2.4737198E-02  1.3204440E-02  6.2884236E-03  2.4674815E-03
295   1.7450942E+03  2.5938621E-02  1.4742301E-02  7.8113383E-03  3.7142919E-03  1.4542888E-03
296   3.5348455E+03  1.3675392E-02  7.5863835E-03  3.9858465E-03  1.8904523E-03  7.4119349E-04
297   7.3220404E+03  6.5496213E-03  3.5904302E-03  1.8851858E-03  8.9437714E-04  3.4992445E-04
298   1.5571453E+04  2.9184844E-03  1.5871405E-03  8.2924399E-04  3.9271030E-04  1.5398430E-04
299   3.4144729E+04  1.2315589E-03  6.6836029E-04  3.5024392E-04  1.6610133E-04  6.4944690E-05
300   7.7552073E+04  4.9178426E-04  2.6591115E-04  1.3860521E-04  6.5589715E-05  2.5734476E-05
301   1.8331766E+05  1.8727871E-04  1.0140133E-04  5.3170924E-05  2.5225700E-05  9.8532403E-06
302   4.5318823E+05  6.7172912E-05  3.6241344E-05  1.8847394E-05  8.9106871E-06  3.5009323E-06
303   1.1774969E+06  2.2854770E-05  1.2377549E-05  6.5065854E-06  3.0902551E-06  1.2047168E-06
304   3.2312890E+06  7.2396045E-06  3.8984144E-06  2.0189998E-06  9.5287306E-07  3.7546967E-07
305   9.4102964E+06  2.1116140E-06  1.1457808E-06  6.0568710E-07  2.8834299E-07  1.1195682E-07
306   2.9216892E+07  6.0709444E-07  3.2630424E-07  1.6820133E-07  7.9223167E-08  3.1323327E-08
307
308
309                           3D-           4D-           5D-           6D-
310                   -1.3454997E+02 -2.8440237E+01 -4.0958645E+00 -1.9118266E-01
311
312
313   1.9453737E-02  1.7089039E-06 -9.8760413E-06  2.7995910E-05  1.4932202E-02
314   3.6141658E-02 -6.0185033E-06  3.4969254E-05 -1.1368619E-04  9.0689119E-02
315   6.7145353E-02  1.2797964E-05 -7.2799441E-05  1.7505638E-04  2.1700399E-01
316   1.2474818E-01 -2.1957719E-05  1.2433456E-04 -4.9191642E-04  3.0663088E-01
317   2.3178217E-01  3.4032852E-05 -1.8495013E-04 -8.3400901E-04  3.1911490E-01
318   4.3071120E-01 -4.9957127E-05  2.9296060E-04 -1.9959124E-02  2.2961461E-01
319   8.0058681E-01  7.1491914E-05 -2.7828719E-04 -1.3222270E-01  6.4324811E-02
320   1.4887931E+00 -9.7118725E-05  9.6601056E-05 -3.8240896E-01 -1.1281170E-01
321   2.7707233E+00  1.0969179E-04 -1.7210460E-02 -4.8283065E-01 -1.5779838E-01
322   5.1625848E+00 -1.2595221E-05 -1.5375034E-01 -1.7055544E-01 -5.8127068E-02
323   9.6361639E+00  2.8679170E-03 -4.3440372E-01  2.2451693E-01  5.6934617E-02
324   1.8031447E+01  4.4723198E-02 -4.3473355E-01  2.5425237E-01  6.6990629E-02
325   3.3858337E+01  2.0918124E-01 -9.0538223E-02  6.8020582E-02  1.8184440E-02
326   6.3875692E+01  3.6808534E-01  1.7719762E-01 -7.2767881E-02 -1.8121532E-02
327   1.2125177E+02  3.1529005E-01  1.8203443E-01 -8.0174035E-02 -2.0259119E-02
328   2.3200920E+02  1.7091567E-01  9.9398001E-02 -4.3637462E-02 -1.1003657E-02
329   4.4845175E+02  6.9494681E-02  3.9939466E-02 -1.7558094E-02 -4.4348522E-03
330   8.7781809E+02  2.3873371E-02  1.3493698E-02 -5.8836773E-03 -1.4818314E-03
331   1.7450942E+03  7.6257647E-03  4.2969254E-03 -1.8824352E-03 -4.7542439E-04
332   3.5348455E+03  2.3509435E-03  1.3111106E-03 -5.6913149E-04 -1.4316013E-04
333   7.3220404E+03  7.7673255E-04  4.3465144E-04 -1.9037934E-04 -4.8106375E-05
334   1.5571453E+04  2.4819454E-04  1.3762045E-04 -5.9590324E-05 -1.4976820E-05
335   3.4144729E+04  1.0097160E-04  5.6322678E-05 -2.4649097E-05 -6.2274833E-06
336   7.7552073E+04  2.3593046E-05  1.3051841E-05 -5.6438896E-06 -1.4177230E-06
337   1.8331766E+05  1.9141952E-05  1.0646356E-05 -4.6448722E-06 -1.1718580E-06
338
339
340   1.9453737E-02 -2.3494679E-08  2.6208878E-09  4.6634915E-08  2.0101141E-05
341   3.6141658E-02 -1.9322798E-08  6.5634513E-08 -1.9694197E-07  1.6643639E-04
342   6.7145353E-02 -3.7623054E-08 -1.3504919E-07  4.5212975E-07  5.4281184E-04
343   1.2474818E-01 -1.7982510E-07  4.7295761E-07 -1.6027518E-06  1.0454788E-03
344   2.3178217E-01 -1.0906279E-07 -6.8682944E-07 -3.8029510E-06  1.4830538E-03
345   4.3071120E-01 -8.6077940E-07  2.1426726E-06 -1.2615206E-04  1.4546706E-03
346   8.0058681E-01 -6.0012341E-07 -1.6582379E-06 -1.1417297E-03  5.5549240E-04
347   1.4887931E+00 -3.7513449E-06  2.5787163E-06 -4.5032764E-03 -1.3286172E-03
348   2.7707233E+00 -3.9610437E-06 -2.7331159E-04 -7.7561746E-03 -2.5348380E-03
349   5.1625848E+00 -1.2698453E-05 -3.3672176E-03 -3.7359314E-03 -1.2732659E-03
350   9.6361639E+00  5.8900205E-05 -1.3005742E-02  6.7252198E-03  1.7056459E-03
351   1.8031447E+01  1.7787270E-03 -1.7785899E-02  1.0400561E-02  2.7403220E-03
352   3.3858337E+01  1.1643180E-02 -5.0457069E-03  3.7911647E-03  1.0135104E-03
353   6.3875692E+01  2.8130069E-02  1.3583480E-02 -5.5843513E-03 -1.3909320E-03
354   1.2125177E+02  3.2908758E-02  1.9024063E-02 -8.3813030E-03 -2.1179794E-03
355   2.3200920E+02  2.4220728E-02  1.4094371E-02 -6.1882584E-03 -1.5604250E-03
356   4.4845175E+02  1.3211997E-02  7.5954655E-03 -3.3396728E-03 -8.4360128E-04
357   8.7781809E+02  5.9478115E-03  3.3626162E-03 -1.4658614E-03 -3.6912643E-04
358   1.7450942E+03  2.3895830E-03  1.3480030E-03 -5.9112469E-04 -1.4935917E-04
359   3.5348455E+03  8.7767933E-04  4.8937629E-04 -2.1203365E-04 -5.3281030E-05
360   7.3220404E+03  3.1659229E-04  1.7794360E-04 -7.8319996E-05 -1.9834947E-05
361   1.5571453E+04  1.0762715E-04  5.9443293E-05 -2.5514972E-05 -6.3839772E-06
362   3.4144729E+04  3.8715731E-05  2.1846633E-05 -9.7070014E-06 -2.4699387E-06
363   7.7552073E+04  1.1319909E-05  6.1760611E-06 -2.6064317E-06 -6.4670721E-07
364   1.8331766E+05  5.6586060E-06  3.1874249E-06 -1.4152137E-06 -3.6001496E-07
365
366
367                           3D+           4D+           5D+           6D+
368                   -1.2828009E+02 -2.6929354E+01 -3.8109062E+00 -1.8232759E-01
369
370
371   1.9453737E-02 -1.7210672E-06  7.3132785E-06 -2.7776140E-05 -1.9530883E-02
372   3.6141658E-02  6.0569545E-06 -2.6038313E-05  1.1990627E-04 -1.0438260E-01
373   6.7145353E-02 -1.2879308E-05  5.3836184E-05 -1.6720252E-04 -2.3373561E-01
374   1.2474818E-01  2.2080100E-05 -9.2557207E-05  5.8753465E-04 -3.1566074E-01
375   2.3178217E-01 -3.4204831E-05  1.3382386E-04  1.4429520E-03 -3.1354910E-01
376   4.3071120E-01  5.0047368E-05 -2.2717258E-04  2.5351517E-02 -2.0760139E-01
377   8.0058681E-01 -7.0646065E-05  2.1103922E-04  1.5247930E-01 -3.9034285E-02
378   1.4887931E+00  9.8359381E-05  2.7872922E-04  4.0535099E-01  1.2157603E-01
379   2.7707233E+00 -1.0556997E-04  2.1603697E-02  4.6918787E-01  1.4784763E-01
380   5.1625848E+00 -4.9550393E-05  1.7560741E-01  1.2765354E-01  4.3641198E-02
381   9.6361639E+00 -3.7085325E-03  4.5397419E-01 -2.4344193E-01 -6.0722751E-02
```

97

```
382    1.8031447E+01 -5.3425366E-02  4.1453450E-01 -2.4127529E-01 -6.1822179E-02
383    3.3858337E+01 -2.2898645E-01  5.5727504E-02 -4.9313322E-02 -1.2973639E-02
384    6.3875692E+01 -3.7665564E-01 -1.9087229E-01  7.9486784E-02  1.9348904E-02
385    1.2125177E+02 -3.0417809E-01 -1.7732290E-01  7.7728580E-02  1.9124179E-02
386    2.3200920E+02 -1.5582420E-01 -9.0941139E-02  3.9705444E-02  9.7488921E-03
387    4.4845175E+02 -5.9172363E-02 -3.4051989E-02  1.4873841E-02  3.6575156E-03
388    8.7781809E+02 -1.8361206E-02 -1.0392119E-02  4.5029277E-03  1.1039906E-03
389    1.7450942E+03 -5.0293490E-03 -2.8410282E-03  1.2378659E-03  3.0452018E-04
390    3.5348455E+03 -1.2433202E-03 -6.9394483E-04  2.9877826E-04  7.3078820E-05
391    7.3220404E+03 -2.9812421E-04 -1.6799494E-04  7.3538561E-05  1.8143947E-05
392    1.5571453E+04 -6.5763503E-05 -3.6294888E-05  1.5444178E-05  3.7554984E-06
393    3.4144729E+04 -1.5152583E-05 -8.6075592E-06  3.8213515E-06  9.4970447E-07
394    7.7552073E+04 -2.7567414E-06 -1.4902716E-06  6.1579429E-07  1.4740009E-07
395    1.8331766E+05 -8.5925459E-07 -4.8980789E-07  2.1885493E-07  5.4570737E-08
396
397
398    1.9453737E-02 -4.7665861E-09  8.1805294E-09 -3.6249172E-08 -2.6296851E-05
399    3.6141658E-02  1.9195897E-08 -4.2306839E-08  2.1627369E-07 -1.9156671E-04
400    6.7145353E-02 -4.8849001E-08  1.2343567E-07 -4.1049585E-07 -5.8467694E-04
401    1.2474818E-01  1.0380432E-07 -2.9641068E-07  1.9900451E-06 -1.0762744E-03
402    2.3178217E-01 -2.0414601E-07  5.9173652E-07  6.7275140E-06 -1.4572199E-03
403    4.3071120E-01  3.8613103E-07 -1.3933250E-06  1.6059381E-04 -1.3152239E-03
404    8.0058681E-01 -7.1410391E-07  1.7536461E-06  1.3171861E-03 -3.3708597E-04
405    1.4887931E+00  1.3136455E-06  3.3901664E-06  4.7746749E-03  1.4319939E-03
406    2.7707233E+00 -1.9267861E-06  3.4719011E-04  7.5386331E-03  2.3753685E-03
407    5.1625848E+00 -7.4723028E-07  3.8540012E-03  2.7970369E-03  9.5627200E-04
408    9.6361639E+00 -1.1191286E-04  1.3605693E-02 -7.2944336E-03 -1.8194942E-03
409    1.8031447E+01 -2.1949542E-03  1.6976115E-02 -9.8769744E-03 -2.5306567E-03
410    3.3858337E+01 -1.2885923E-02  3.0958817E-03 -2.7485193E-03 -7.2321262E-04
411    6.3875692E+01 -2.9036735E-02 -1.4706197E-02  6.1253089E-03  1.4911026E-03
412    1.2125177E+02 -3.2140078E-02 -1.8703353E-02  8.1956820E-03  2.0163590E-03
413    2.3200920E+02 -2.2569729E-02 -1.3141606E-02  5.7348291E-03  1.4079636E-03
414    4.4845175E+02 -1.1722481E-02 -6.7283794E-03  2.9373860E-03  7.2227047E-04
415    8.7781809E+02 -4.9597391E-03 -2.7988715E-03  1.2118459E-03  2.9706335E-04
416    1.7450942E+03 -1.8420135E-03 -1.0375926E-03  4.5193033E-04  1.1118511E-04
417    3.5348455E+03 -6.1075133E-04 -3.3963629E-04  1.4600229E-04  3.5690622E-05
418    7.3220404E+03 -1.9417130E-04 -1.0924950E-04  4.7892364E-05  1.1828134E-05
419    1.5571453E+04 -5.5414419E-05 -3.0380139E-05  1.2843419E-05  3.1128767E-06
420    3.4144729E+04 -1.6218355E-05 -9.2679551E-06  4.1602860E-06  1.0398485E-06
421    7.7552073E+04 -3.6486776E-06 -1.9252975E-06  7.6724542E-07  1.7990722E-07
422    1.8331766E+05 -1.2882697E-06 -7.5080026E-07  3.4612983E-07  8.7631389E-08
423
424
425                           4F-           5F-
426                  -1.4283914E+01 -2.9213544E-01
427
428
429    1.9453737E-02  6.6519343E-06 -7.5234628E-05
430    3.6141658E-02 -2.1360454E-05 -2.6536652E-03
431    6.7145353E-02  4.6289376E-05 -1.8069774E-02
432    1.2474818E-01 -6.6350273E-05 -5.9145321E-02
433    2.3178217E-01  1.2967025E-04 -1.2997399E-01
434    4.3071120E-01 -1.3368066E-04 -2.2013932E-01
435    8.0058681E-01  4.0938667E-04 -2.9387764E-01
436    1.4887931E+00  3.0219209E-03 -3.0617901E-01
437    2.7707233E+00  3.7821876E-02 -2.2297888E-01
438    5.1625848E+00  1.5394501E-01 -7.7968302E-02
439    9.6361639E+00  3.0647180E-01  5.1295935E-02
440    1.8031447E+01  3.4548477E-01  9.1788477E-02
441    3.3858337E+01  2.5411508E-01  7.2862946E-02
442    6.3875692E+01  1.2634752E-01  3.7217372E-02
443    1.2125177E+02  4.5722954E-02  1.3327572E-02
444    2.3200920E+02  1.3450450E-02  3.9475329E-03
445    4.4845175E+02  3.1290776E-03  9.0396660E-04
446    8.7781809E+02  7.5391295E-04  2.2220654E-04
447    1.7450942E+03  1.2847135E-04  3.6367372E-05
448    3.5348455E+03  4.9214287E-05  1.4522570E-05
449
450
451    1.9453737E-02  9.7802298E-09 -1.1425213E-07
452    3.6141658E-02 -4.5904641E-08 -5.5196843E-06
453    6.7145353E-02  1.2937723E-07 -5.1249869E-05
454    1.2474818E-01 -2.6772194E-07 -2.2864402E-04
455    2.3178217E-01  6.6021547E-07 -6.8490365E-04
456    4.3071120E-01 -1.0452365E-06 -1.5812801E-03
457    8.0058681E-01  3.7845584E-06 -2.8778807E-03
458    1.4887931E+00  3.9665323E-05 -4.0883228E-03
459    2.7707233E+00  6.8727947E-04 -4.0609157E-03
460    5.1625848E+00  3.8233705E-03 -1.9368644E-03
461    9.6361639E+00  1.0396309E-02  1.7415482E-03
462    1.8031447E+01  1.6009999E-02  4.2547758E-03
463    3.3858337E+01  1.6086541E-02  4.6134703E-03
464    6.3875692E+01  1.0916987E-02  3.2160372E-03
465    1.2125177E+02  5.3763349E-03  1.5670944E-03
466    2.3200920E+02  2.1331360E-03  6.2619880E-04
467    4.4845175E+02  6.6181052E-04  1.9105941E-04
468    8.7781809E+02  2.0336879E-04  6.0086793E-05
469    1.7450942E+03  4.5163176E-05  1.2694798E-05
470    3.5348455E+03  1.7481944E-05  5.2199366E-06
471
472
473                           4F+           5F+
```

98

```
474                -1.3899400E+01 -2.7004390E-01
475
476
477    1.9453737E-02 -7.5710755E-06  1.4358729E-04
478    3.6141658E-02  2.4162114E-05  3.7346339E-03
479    6.7145353E-02 -5.2886361E-05  2.2106892E-02
480    1.2474818E-01  7.4984490E-05  6.6652471E-02
481    2.3178217E-01 -1.4619715E-04  1.3906611E-01
482    4.3071120E-01  1.5321002E-04  2.2654846E-01
483    8.0058681E-01 -4.5908422E-04  2.9384540E-01
484    1.4887931E+00 -3.4599894E-03  3.0112350E-01
485    2.7707233E+00 -4.0854397E-02  2.1487666E-01
486    5.1625848E+00 -1.6027303E-01  6.9670258E-02
487    9.6361639E+00 -3.1187985E-01 -5.4403079E-02
488    1.8031447E+01 -3.4421117E-01 -9.0715310E-02
489    3.3858337E+01 -2.4827600E-01 -7.0373982E-02
490    6.3875692E+01 -1.2151771E-01 -3.5284216E-02
491    1.2125177E+02 -4.2830578E-02 -1.2308746E-02
492    2.3200920E+02 -1.2088313E-02 -3.4969605E-03
493    4.4845175E+02 -2.6662822E-03 -7.5896215E-04
494    8.7781809E+02 -5.7540121E-04 -1.6751694E-04
495    1.7450942E+03 -8.8544591E-05 -2.4524420E-05
496    3.5348455E+03 -2.3576924E-05 -6.9470986E-06
497
498
499    1.9453737E-02 -1.2101571E-08  2.1907964E-07
500    3.6141658E-02  5.1973800E-08  7.7720385E-06
501    6.7145353E-02 -1.5341845E-07  6.2702951E-05
502    1.2474818E-01  2.9575690E-07  2.5768755E-04
503    2.3178217E-01 -7.7984903E-07  7.3285185E-04
504    4.3071120E-01  1.1153267E-06  1.6274525E-03
505    8.0058681E-01 -4.5192354E-06  2.8778425E-03
506    1.4887931E+00 -4.6192042E-05  4.0215028E-03
507    2.7707233E+00 -7.4463276E-04  3.9142810E-03
508    5.1625848E+00 -3.9866749E-03  1.7314970E-03
509    9.6361639E+00 -1.0595697E-02 -1.8487122E-03
510    1.8031447E+01 -1.5983972E-02 -4.2120077E-03
511    3.3858337E+01 -1.5775294E-02 -4.4706439E-03
512    6.3875692E+01 -1.0569570E-02 -3.0682566E-03
513    1.2125177E+02 -5.1013080E-03 -1.4655355E-03
514    2.3200920E+02 -1.9715140E-03 -5.7017825E-04
515    4.4845175E+02 -5.9357692E-04 -1.6884944E-04
516    8.7781809E+02 -1.7454005E-04 -5.0836243E-05
517    1.7450942E+03 -3.6449349E-05 -1.0060536E-05
518    3.5348455E+03 -1.2858668E-05 -3.8037142E-06
519
520
521    Calculated energy =   -2.7274376153339403E+04
522    Literature energy =   -2.7274378906250000E+04
523    Difference (cal - lit) = 2.75E-03
524    Relative error (abs((cal - lit)/lit)) = 1.01E-07
525    SCF converged at iteration 38 Final energy =     -27274.37615333940
526    SCF converged at iteration 40 Final energy =     -27274.37652373574
527    SCF converged at iteration 39 Final energy =     -27274.36557271042
528    SCF converged at iteration 38 Final energy =     -27274.37328631207
529    SCF converged at iteration 38 Final energy =     -27274.37681712200
530    SCF converged at iteration 36 Final energy =     -27274.37555675471
531    SCF converged at iteration 38 Final energy =     -27274.02325483119
532    SCF converged at iteration 38 Final energy =     -27274.37703303956
533    SCF converged at iteration 38 Final energy =     -27274.37463387263
534    SCF converged at iteration 40 Final energy =     -27274.36624593227
535    SCF converged at iteration 39 Final energy =     -27274.36325682023
536    SCF converged at iteration 38 Final energy =     -27274.37675780536
537    SCF converged at iteration 40 Final energy =     -27274.37705991919
538    SCF converged at iteration 40 Final energy =     -27274.37711582447
539    SCF converged at iteration 39 Final energy =     -27274.36526340017
540    SCF converged at iteration 40 Final energy =     -27274.37525627186
541    SCF converged at iteration 40 Final energy =     -27274.37554430689
542    SCF converged at iteration 39 Final energy =     -27274.37525819958
543    SCF converged at iteration 40 Final energy =     -27274.37618439044
544    SCF converged at iteration 40 Final energy =     -27274.37569386304
545       interm  rho nf F del 0.010000000    20  -27274.377 115 824 5   -27274.377 115 824 5
546    :::>   1.1000000       1.0000000       0.90000000       1.0000000
547    SCF converged at iteration 40 Final energy =     -27274.37624394280
548    SCF converged at iteration 40 Final energy =     -27274.37681479764
549    SCF converged at iteration 40 Final energy =     -27274.37660684840
550    SCF converged at iteration 40 Final energy =     -27274.37710938528
551    SCF converged at iteration 40 Final energy =     -27274.37709990601
552    SCF converged at iteration 40 Final energy =     -27274.37712130963
553    SCF converged at iteration 40 Final energy =     -27274.37716226352
554    SCF converged at iteration 40 Final energy =     -27274.37717442678
555    SCF converged at iteration 40 Final energy =     -27274.37716723333
556    SCF converged at iteration 40 Final energy =     -27274.37709635337
557    SCF converged at iteration 40 Final energy =     -27274.37717330824
558    SCF converged at iteration 40 Final energy =     -27274.37700224991
559    SCF converged at iteration 40 Final energy =     -27274.37717427240
560    SCF converged at iteration 40 Final energy =     -27274.37679601249
561    SCF converged at iteration 40 Final energy =     -27274.37717971213
562    SCF converged at iteration 40 Final energy =     -27274.37716637889
563    SCF converged at iteration 40 Final energy =     -27274.37680382172
564    SCF converged at iteration 40 Final energy =     -27274.37718793943
565    SCF converged at iteration 40 Final energy =     -27274.37718724257
```

```
566   SCF converged at iteration 40 Final energy =      -27274.37697998060
567   SCF converged at iteration 40 Final energy =      -27274.37719820330
568   SCF converged at iteration 40 Final energy =      -27274.37720555374
569   SCF converged at iteration 40 Final energy =      -27274.37721015110
570   SCF converged at iteration 40 Final energy =      -27274.37721989711
571   SCF converged at iteration 40 Final energy =      -27274.37723141619
572   SCF converged at iteration 40 Final energy =      -27274.37724506508
573   SCF converged at iteration 40 Final energy =      -27274.37727356602
574   SCF converged at iteration 40 Final energy =      -27274.37726115421
575   SCF converged at iteration 40 Final energy =      -27274.37679573947
576   SCF converged at iteration 40 Final energy =      -27274.37729431578
577   SCF converged at iteration 40 Final energy =      -27274.37731317497
578   SCF converged at iteration 40 Final energy =      -27274.37732184126
579   SCF converged at iteration 40 Final energy =      -27274.37734102829
580   SCF converged at iteration 40 Final energy =      -27274.37737425937
581   SCF converged at iteration 41 Final energy =      -27274.37741422374
582   SCF converged at iteration 40 Final energy =      -27274.37747058523
583   SCF converged at iteration 41 Final energy =      -27274.37143110862
584   SCF converged at iteration 41 Final energy =      -27274.37751839181
585   SCF converged at iteration 41 Final energy =      -27274.37753025332
586   SCF converged at iteration 40 Final energy =      -27274.37668602075
587   SCF converged at iteration 41 Final energy =      -27274.37739254861
588   SCF converged at iteration 41 Final energy =      -27274.37729343396
589   SCF converged at iteration 41 Final energy =      -27274.37753046981
590   SCF converged at iteration 41 Final energy =      -27274.37738289865
591   SCF converged at iteration 41 Final energy =      -27274.37745108623
592   SCF converged at iteration 41 Final energy =      -27274.37745035628
593   SCF converged at iteration 41 Final energy =      -27274.37752038428
594   SCF converged at iteration 41 Final energy =      -27274.37747841063
595   SCF converged at iteration 41 Final energy =      -27274.37753539678
596   SCF converged at iteration 41 Final energy =      -27274.37753189082
597   SCF converged at iteration 41 Final energy =      -27274.37753338237
598   SCF converged at iteration 41 Final energy =      -27274.37739192332
599   SCF converged at iteration 41 Final energy =      -27274.37753340209
600   SCF converged at iteration 41 Final energy =      -27274.37753764559
601   SCF converged at iteration 41 Final energy =      -27274.37753647355
602   SCF converged at iteration 41 Final energy =      -27274.37743857712
603   SCF converged at iteration 41 Final energy =      -27274.37753417706
604   SCF converged at iteration 41 Final energy =      -27274.37746150104
605   SCF converged at iteration 41 Final energy =      -27274.37753562298
606   SCF converged at iteration 41 Final energy =      -27274.37727249676
607   SCF converged at iteration 41 Final energy =      -27274.37753843775
608   SCF converged at iteration 41 Final energy =      -27274.37743420742
609   SCF converged at iteration 41 Final energy =      -27274.37744247752
610   SCF converged at iteration 41 Final energy =      -27274.37753800378
611   SCF converged at iteration 41 Final energy =      -27274.37752144962
612   SCF converged at iteration 41 Final energy =      -27274.37753749272
613        interm  rho nf F del 0.001000000     86  -27274.377 538 437 7       -0.000 422 613 3
614   :::>    1.6627416       1.0021141       1.0254277       1.0623435
615   SCF converged at iteration 41 Final energy =      -27274.37753764903
616   SCF converged at iteration 41 Final energy =      -27274.37753839733
617   SCF converged at iteration 41 Final energy =      -27274.37753881774
618   SCF converged at iteration 41 Final energy =      -27274.37753908531
619   SCF converged at iteration 41 Final energy =      -27274.37753966563
620   SCF converged at iteration 41 Final energy =      -27274.37754040643
621   SCF converged at iteration 41 Final energy =      -27274.37754022898
622   SCF converged at iteration 41 Final energy =      -27274.37751653413
623   SCF converged at iteration 41 Final energy =      -27274.37754015865
624   SCF converged at iteration 41 Final energy =      -27274.37753751040
625   SCF converged at iteration 41 Final energy =      -27274.37754101662
626   SCF converged at iteration 41 Final energy =      -27274.37754084213
627   SCF converged at iteration 41 Final energy =      -27274.37753869069
628   SCF converged at iteration 41 Final energy =      -27274.37754104751
629   SCF converged at iteration 41 Final energy =      -27274.37753933119
630   SCF converged at iteration 41 Final energy =      -27274.37754116962
631   SCF converged at iteration 41 Final energy =      -27274.37754138718
632   SCF converged at iteration 41 Final energy =      -27274.37754179706
633   SCF converged at iteration 41 Final energy =      -27274.37754246896
634   SCF converged at iteration 41 Final energy =      -27274.37754335791
635   SCF converged at iteration 41 Final energy =      -27274.37754042695
636   SCF converged at iteration 41 Final energy =      -27274.37748754261
637   SCF converged at iteration 41 Final energy =      -27274.37754395899
638   SCF converged at iteration 41 Final energy =      -27274.37749778102
639   SCF converged at iteration 41 Final energy =      -27274.37754367125
640   SCF converged at iteration 41 Final energy =      -27274.37753077486
641   SCF converged at iteration 41 Final energy =      -27274.37754428152
642   SCF converged at iteration 41 Final energy =      -27274.37754418491
643   SCF converged at iteration 41 Final energy =      -27274.37754109361
644   SCF converged at iteration 41 Final energy =      -27274.37754440276
645   SCF converged at iteration 41 Final energy =      -27274.37754461948
646   SCF converged at iteration 41 Final energy =      -27274.37754500144
647   SCF converged at iteration 41 Final energy =      -27274.37754557010
648   SCF converged at iteration 41 Final energy =      -27274.37754603123
649   SCF converged at iteration 41 Final energy =      -27274.37754609415
650   SCF converged at iteration 41 Final energy =      -27274.37754610645
651   SCF converged at iteration 41 Final energy =      -27274.37754254067
652   SCF converged at iteration 41 Final energy =      -27274.37754561090
653   SCF converged at iteration 41 Final energy =      -27274.37754609332
654   SCF converged at iteration 41 Final energy =      -27274.37754602444
655   SCF converged at iteration 41 Final energy =      -27274.37754608405
656   SCF converged at iteration 41 Final energy =      -27274.37754392067
657   SCF converged at iteration 41 Final energy =      -27274.37754610813
```

```
SCF converged at iteration 41 Final energy =      -27274.37754439248
SCF converged at iteration 41 Final energy =      -27274.37754610794
SCF converged at iteration 41 Final energy =      -27274.37754526297
SCF converged at iteration 41 Final energy =      -27274.37754610373
SCF converged at iteration 41 Final energy =      -27274.37754374482
SCF converged at iteration 41 Final energy =      -27274.37754610522
SCF converged at iteration 41 Final energy =      -27274.37754610387
SCF converged at iteration 41 Final energy =      -27274.37754610267
     interm  rho nf F del 0.000100000   137   -27274.377 546 108 1     -0.000 007 670 4
 :::>   1.6853380      1.0025643      1.0865944      1.1572454
SCF converged at iteration 41 Final energy =      -27274.37754610732
SCF converged at iteration 41 Final energy =      -27274.37754605688
SCF converged at iteration 41 Final energy =      -27274.37754610859
SCF converged at iteration 41 Final energy =      -27274.37754610906
SCF converged at iteration 41 Final energy =      -27274.37754610936
SCF converged at iteration 41 Final energy =      -27274.37754610938
SCF converged at iteration 41 Final energy =      -27274.37754610937
SCF converged at iteration 41 Final energy =      -27274.37754609040
SCF converged at iteration 41 Final energy =      -27274.37754610926
SCF converged at iteration 41 Final energy =      -27274.37754608270
SCF converged at iteration 41 Final energy =      -27274.37754610941
SCF converged at iteration 41 Final energy =      -27274.37754610939
SCF converged at iteration 41 Final energy =      -27274.37754608488
SCF converged at iteration 41 Final energy =      -27274.37754610936
SCF converged at iteration 41 Final energy =      -27274.37754608941
SCF converged at iteration 41 Final energy =      -27274.37754610772
SCF converged at iteration 41 Final energy =      -27274.37754610937
SCF converged at iteration 41 Final energy =      -27274.37754608604
SCF converged at iteration 41 Final energy =      -27274.37754610927
     interm  rho nf F del 0.000010000   156   -27274.377 546 109 4     -0.000 000 001 3
 :::>   1.6849066      1.0025845      1.0874081      1.1585546
SCF converged at iteration 41 Final energy =      -27274.37754610938
SCF converged at iteration 41 Final energy =      -27274.37754610940
SCF converged at iteration 41 Final energy =      -27274.37754610942
SCF converged at iteration 41 Final energy =      -27274.37754610938
SCF converged at iteration 41 Final energy =      -27274.37754610911
SCF converged at iteration 41 Final energy =      -27274.37754610940
SCF converged at iteration 41 Final energy =      -27274.37754610917
SCF converged at iteration 41 Final energy =      -27274.37754610942
SCF converged at iteration 41 Final energy =      -27274.37754610939
SCF converged at iteration 41 Final energy =      -27274.37754610940
SCF converged at iteration 41 Final energy =      -27274.37754610939
SCF converged at iteration 41 Final energy =      -27274.37754610930
SCF converged at iteration 41 Final energy =      -27274.37754610940
SCF converged at iteration 41 Final energy =      -27274.37754610910
SCF converged at iteration 41 Final energy =      -27274.37754610940
SCF converged at iteration 41 Final energy =      -27274.37754610916
SCF converged at iteration 41 Final energy =      -27274.37754610941

    At the return from NEWUOA     number of function values =    173
    F= -27274.37754610942
       1.6848935      1.0025849      1.0874118      1.1585570
New parameters are
 3.2777475E-02  1.8626263E+00  5.3795715E+00  1.6670872E+00
SCF converged at iteration 41 Final energy =      -27274.37754610942


                    1S+           2S+           3S+           4S+           5S+           6S+           7S+
              -4.1678552E+03 -7.8202739E+02 -1.9970482E+02 -5.2155429E+01 -1.1997194E+01 -2.0494257E+00 -1.9849676E-01


 3.2777475E-02  6.3766410E-07  5.7309076E-06 -6.1014939E-05  1.3503449E-04 -8.9549947E-05  2.0217706E-04  7.7763953E-01
 6.1052208E-02 -2.7212503E-06 -2.4451140E-05  2.5997072E-04 -5.7257873E-04  3.6084676E-04 -1.8505829E-03  1.9084131E-01
 1.1371780E-01  6.4661788E-06  5.8087512E-05 -6.1594002E-04  1.3479564E-03 -7.5938581E-04 -1.0442761E-02  4.3509449E-01
 2.1181678E-01 -1.1776869E-05 -1.0570257E-04  1.1185370E-03 -2.4045368E-03  1.2148510E-03 -2.1151300E-01 -4.1956006E-01
 3.9455421E-01  1.8769658E-05  1.6866127E-04 -1.7636196E-03  3.7907496E-03 -1.2618495E-04 -7.3718664E-01 -3.5322441E-01
 7.3499994E-01 -2.7775219E-05 -2.4890013E-04  2.6020802E-03 -5.2448412E-03  2.1094051E-02 -6.2243418E-01 -3.3766148E-01
 1.3694267E+00  3.9581146E-05  3.5469575E-04 -3.6690519E-03  6.5256712E-03  3.6524111E-01  4.7977497E-01  2.0741439E-01
 2.5522453E+00 -5.5531012E-05 -5.0644472E-04  4.8195743E-03 -1.3768212E-02  1.0628665E+00  1.0136423E+00  3.4555769E-01
 4.7592004E+00  7.5042533E-05  6.7454424E-04 -6.6562007E-03 -1.5840398E-01  1.2336517E-01  1.2190063E-01  4.7160748E-02
 8.8821010E+00 -9.9650704E-05 -8.8184744E-04  1.1265810E-02 -9.1814825E-01 -1.2574109E+00 -7.1892239E-01 -2.3332330E-01
 1.6598538E+01  1.3971887E-04  1.4607049E-03  7.8898658E-02 -5.7101352E-01 -5.1418558E-01 -2.6643838E-01 -8.2270513E-02
 3.1079528E+01 -1.7602025E-04 -2.0695834E-03  6.1053018E-01  8.6852650E-01  5.6179744E-01  2.6866146E-01  8.2644473E-02
 5.8358006E+01  1.6963083E-04 -2.3283094E-02  7.0220007E-01  7.7741233E-01  4.7333116E-01  2.1834120E-01  6.7834903E-02
 1.1000966E+02 -2.3821652E-04 -2.5386103E-01 -2.1304165E-01 -1.4183476E-01 -7.6052348E-02 -3.3602641E-02 -1.0499887E-02
 2.0848751E+02  8.5338160E-04 -5.1752274E-01 -5.4684268E-01 -3.4531663E-01 -1.7912344E-01 -8.0359590E-02 -2.4606349E-02
 3.9793692E+02  2.2723041E-02 -3.2469759E-01 -2.6916770E-01 -1.6072880E-01 -8.2191235E-02 -3.6464966E-02 -1.1231440E-02
 7.6660359E+02  1.2996824E-01  7.6465413E-03 -8.8975780E-03 -6.2835928E-03 -3.2962445E-03 -1.5389625E-03 -4.5630116E-04
 1.4944323E+03  2.4681387E-01  1.4585997E-01  7.6029536E-02  3.9881367E-02  1.9666704E-02  8.7101348E-03  2.6628490E-03
 2.9570622E+03  2.5643870E-01  1.4221656E-01  7.3229143E-02  3.8524236E-02  1.9030766E-02  8.3940292E-03  2.5765332E-03
 5.9602824E+03  1.9274715E-01  9.7608608E-02  4.9362514E-02  2.5773385E-02  1.2700084E-02  5.6108718E-03  1.7187114E-03
 1.2287274E+04  1.2229619E-01  5.8398368E-02  2.9197147E-02  1.5227645E-02  7.5035816E-03  3.3103362E-03  1.0152861E-03
 2.6025475E+04  7.1426092E-02  3.2800158E-02  1.6284612E-02  8.4675426E-03  4.1682198E-03  1.8402301E-03  5.6389575E-04
 5.6920168E+04  4.0099771E-02  1.7926580E-02  8.8580193E-03  4.6043650E-03  2.2666591E-03  1.0000444E-03  3.0662309E-04
 1.2924147E+05  2.2205546E-02  9.7155189E-03  4.7836464E-03  2.4828387E-03  1.2216568E-03  5.3917832E-04  1.6524649E-04
 3.0640048E+05  1.2181181E-02  5.2395866E-03  2.5724532E-03  1.3347501E-03  6.5674013E-04  2.8975990E-04  8.8829643E-05
 7.6297421E+05  6.6597693E-03  2.8244368E-03  1.3834892E-03  7.1723201E-04  3.5280484E-04  1.5568362E-04  4.7717415E-05
 2.0076699E+06  3.5575042E-03  1.4928906E-03  7.2995848E-04  3.7832779E-04  1.8609143E-04  8.2104714E-05  2.5168451E-05
 5.6164426E+06  1.8566153E-03  7.7288988E-04  3.7740417E-04  1.9551278E-04  9.6154484E-05  4.2426773E-05  1.3004305E-05
 1.6802886E+07  8.8408903E-04  3.6609607E-04  1.7860420E-04  9.2513170E-05  4.5497769E-05  2.0073628E-05  6.1532077E-06
 5.4065393E+07  3.6372280E-04  1.5003603E-04  7.3149434E-05  3.7880674E-05  1.8628161E-05  8.2191072E-06  2.5192723E-06
 1.8809445E+08  3.9084666E-05  1.6088047E-05  7.8407006E-06  4.0605376E-06  1.9968838E-06  8.8097218E-07  2.7005784E-07
```

```
750   7.1100931E+08 -2.6913120E-06 -1.1074607E-06 -5.3969183E-07 -2.7956824E-07 -1.3750197E-07 -6.0650654E-08 -1.8595851E-08
751   2.9330971E+09  4.2284342E-07  1.7399041E-07  8.4788201E-08  4.3925941E-08  2.1605380E-08  9.5292259E-09  2.9219333E-09
752   1.3255936E+10 -6.6036810E-08 -2.7172566E-08 -1.3241564E-08 -6.8602231E-09 -3.3743106E-09 -1.4882350E-09 -4.5634558E-10
753   6.5852665E+10  8.2684455E-09  3.4022696E-09  1.6579716E-09  8.5897268E-10  4.2250091E-10  1.8634247E-10  5.7139514E-11
754
755
756   3.2777475E-02  6.6583018E-09  5.1997131E-09 -9.0280402E-08  1.4001238E-07 -1.1702732E-07  2.2257972E-07  8.8973711E-04
757   6.1052208E-02 -2.7234862E-08 -3.3234803E-08  4.8543240E-07 -8.3875912E-07  6.1956519E-07 -2.8561455E-06  2.9801286E-04
758   1.1371780E-01  6.5584587E-08  1.1357878E-07 -1.4921476E-06  2.7503848E-06 -1.7436024E-06 -2.2330833E-05  9.2721339E-04
759   2.1181678E-01 -1.2687135E-07 -2.9131507E-07  3.5750657E-06 -6.7804012E-06  3.7545795E-06 -6.1508893E-04 -1.2202708E-03
760   3.9455421E-01  2.2292090E-07  6.4801348E-07 -7.5181259E-06  1.4716319E-05 -8.4814951E-07 -2.9266569E-03 -1.4022390E-03
761   7.3499994E-01 -3.7623323E-07 -1.3240078E-06  1.4888317E-05 -2.7928998E-05  1.1483385E-04 -3.3719703E-03 -1.8292450E-03
762   1.3694267E+00  6.2820434E-07  2.6019932E-06 -2.8314609E-05  4.7559797E-05  2.7010967E-03  3.5481729E-03  1.5339608E-03
763   2.5522453E+00 -1.0548532E-06 -5.1099838E-06  5.0406770E-05 -1.3805043E-04  1.0733461E-02  1.0233838E-02  3.4886688E-03
764   4.7592004E+00  1.7607038E-06  9.3408006E-06 -9.4370031E-05 -2.1881955E-03  1.6938343E-03  1.6747161E-03  6.4816516E-04
765   8.8821010E+00 -2.9467977E-06 -1.6757433E-05  2.1620569E-04 -1.7304956E-02 -2.3680505E-02 -1.3537694E-02 -4.3935092E-03
766   1.6598538E+01  5.1797725E-06  3.8010920E-05  2.0360874E-03 -1.4680955E-02 -1.3203591E-02 -6.8396882E-03 -2.1117539E-03
767   3.1079528E+01 -8.5036007E-06 -7.3720904E-05  2.2195179E-02  3.0642804E-02  1.9816957E-02  9.4758500E-03  2.9149436E-03
768   5.8358006E+01  1.1461839E-05 -1.1455299E-03  3.3856539E-02  3.7345896E-02  2.2720121E-02  1.0478503E-02  3.2554645E-03
769   1.1000966E+02 -2.0387677E-05 -1.7093763E-02 -1.4384488E-02 -9.6242069E-03 -5.1671769E-03 -2.2851648E-03 -7.1377344E-04
770   2.0848751E+02  8.9042294E-05 -4.7607546E-02 -4.9636309E-02 -3.3268197E-02 -1.6208787E-02 -7.2705507E-03 -2.2261834E-03
771   3.9793692E+02  3.1750790E-03 -4.0355126E-02 -3.2905184E-02 -1.9582457E-02 -1.0005800E-02 -4.4379426E-03 -1.3669924E-03
772   7.6660359E+02  2.4905109E-02  2.4571143E-03 -6.4702145E-04 -5.6630997E-04 -3.0559609E-04 -1.4848940E-04 -4.2804830E-05
773   1.4944323E+03  6.4312504E-02  3.5404032E-02  1.8368843E-02  9.6269446E-03  4.7461617E-03  2.1018145E-03  6.4257151E-04
774   2.9570622E+03  8.9937328E-02  4.5939928E-02  2.3407135E-02  1.2284219E-02  6.0646469E-03  2.6744338E-03  8.2093827E-04
775   5.9602824E+03  8.9537818E-02  4.1711583E-02  2.0842689E-02  1.0849360E-02  5.3416257E-03  2.3596508E-03  7.2272164E-04
776   1.2287274E+04  7.3753937E-02  3.2516176E-02  1.6061406E-02  8.3530855E-03  4.1130977E-03  1.8140569E-03  5.5641213E-04
777   2.6025475E+04  5.4420421E-02  2.3238701E-02  1.1407274E-02  5.9138670E-03  2.9087461E-03  1.2840890E-03  3.9342287E-04
778   5.6920168E+04  3.7450532E-02  1.5729658E-02  7.6951575E-03  3.9906895E-03  1.9634095E-03  8.6601592E-04  2.6555365E-04
779   1.2924147E+05  2.4513497E-02  1.0196669E-02  4.9794708E-03  2.5787438E-03  1.2680510E-03  5.5964584E-04  1.7149460E-04
780   3.0640048E+05  1.5383160E-02  6.3679065E-03  3.1066330E-03  1.6096318E-03  7.9171441E-04  3.4923746E-04  1.0707400E-04
781   7.6297421E+05  9.2503621E-03  3.8170961E-03  1.8612148E-03  9.6364657E-04  4.7383792E-04  2.0909878E-04  6.4081524E-05
782   2.0076699E+06  5.2989334E-03  2.1831696E-03  1.0641632E-03  5.5118940E-04  2.7107876E-04  1.1958448E-04  3.6660831E-05
783   5.6164426E+06  2.8243198E-03  1.1621686E-03  5.6638269E-04  2.9323102E-04  1.4418577E-04  6.3623385E-05  1.9499446E-05
784   1.6802886E+07  1.3403351E-03  5.5119574E-04  2.6858785E-04  1.3909881E-04  6.8406847E-05  3.0177984E-05  9.2513068E-06
785   5.4065393E+07  4.2517766E-04  1.7477765E-04  8.5163139E-05  4.4087413E-05  2.1677739E-05  9.5657566E-06  2.9316483E-06
786   1.8809445E+08  9.8714923E-06  4.0808855E-06  1.9896300E-06  1.0343690E-06  5.0954248E-07  2.2422999E-07  6.8918938E-08
787   7.1100931E+08  1.1624993E-06  4.7251887E-07  2.3006947E-07  1.1756714E-07  5.7473477E-08  2.5582624E-08  7.7693109E-09
788   2.9330971E+09 -5.6493630E-07 -2.3038071E-07 -1.1220197E-07 -5.7527294E-08 -2.8164677E-08 -1.2508763E-08 -3.8077421E-09
789   1.3255936E+10  2.0123625E-07  8.2100500E-08  3.9986755E-08  2.0510704E-08  1.0043772E-08  4.4594273E-09  1.3578940E-09
790   6.5852665E+10 -5.6896317E-08 -2.3214553E-08 -1.1306646E-08 -5.8000751E-09 -2.8403099E-09 -1.2610270E-09 -3.8400415E-10
791
792
793                     2P-            3P-            4P-            5P-            6P-
794             -7.5278343E+02 -1.8645827E+02 -4.6167441E+01 -9.6054318E+00 -1.2842044E+00
795
796
797   3.2777475E-02 -7.1419616E-07  7.8103090E-06  8.0014713E-05  4.1421115E-05 -3.0036559E-04
798   6.1052208E-02  2.7478595E-06 -2.9964405E-05 -3.0698594E-04 -2.0084087E-04 -4.6908852E-03
799   1.1371780E-01 -6.1266438E-06  6.6795018E-05  6.7484733E-04  3.4852567E-04 -2.5326033E-02
800   2.1181678E-01  1.0806498E-05 -1.1735248E-04 -1.1748191E-03 -7.1990125E-04 -1.9859318E-01
801   3.9455421E-01 -1.7050462E-05  1.8367136E-04  1.8136919E-03  2.4591053E-03 -5.0663001E-01
802   7.3499994E-01  2.5467950E-05 -2.7268255E-04 -2.5270870E-03  1.8786243E-02 -4.8412064E-01
803   1.3694267E+00 -3.5833116E-05  4.0248005E-04  3.3854562E-03  2.2862081E-01  5.3893134E-02
804   2.5522453E+00  5.7665574E-05 -4.8877642E-04 -7.5621053E-03  6.5921867E-01  4.7808254E-01
805   4.7592004E+00 -7.1829380E-05  8.1151373E-04 -7.6853322E-02  4.1932271E-01  2.7962063E-01
806   8.8821010E+00  9.0786449E-05 -2.2127713E-03 -4.8837158E-01 -4.1452975E-01 -1.8996887E-01
807   1.6598538E+01 -2.6568400E-04 -3.2082164E-02 -6.1426897E-01 -5.0407795E-01 -2.3595179E-01
808   3.1079528E+01  3.3486473E-04 -2.6807434E-01  3.5412768E-02 -1.9851858E-02 -1.1221864E-02
809   5.8358006E+01  6.0909364E-03 -5.3331952E-01  4.5960941E-01  2.4801552E-01  1.0435215E-01
810   1.1000966E+02  8.1427606E-02 -2.9005669E-01  2.4006230E-01  1.2924157E-01  5.4795638E-02
811   2.0848751E+02  2.4884351E-01  6.9516352E-02 -2.1149330E-02 -8.3830283E-03 -3.3844827E-03
812   3.9793692E+02  3.2142412E-01  1.8355468E-01 -9.7197522E-02 -4.6461116E-02 -1.9084088E-02
813   7.6660359E+02  2.4991232E-01  1.4391088E-01 -7.6545227E-02 -3.6856495E-02 -1.5215454E-02
814   1.4944323E+03  1.4966669E-01  8.3245213E-02 -4.3851270E-02 -2.0984900E-02 -8.6331107E-03
815   2.9570622E+03  7.9577568E-02  4.3190468E-02 -2.2617427E-02 -1.0837131E-02 -4.4669823E-03
816   5.9602824E+03  4.1165686E-02  2.1906788E-02 -1.1412507E-02 -5.4487875E-03 -2.2411963E-03
817   1.2287274E+04  2.1764771E-02  1.1463227E-02 -5.9575944E-03 -2.8476002E-03 -1.1727872E-03
818   2.6025475E+04  1.1962129E-02  6.2485980E-03 -3.2404186E-03 -1.5460261E-03 -6.3599606E-04
819   5.6920168E+04  6.8009560E-03  3.5390295E-03 -1.8337079E-03 -8.7543173E-04 -3.6036597E-04
820   1.2924147E+05  3.9458576E-03  2.0472703E-03 -1.0599356E-03 -5.0564610E-04 -2.0804375E-04
821   3.0640048E+05  2.2955149E-03  1.1896197E-03 -6.1574652E-04 -2.9382355E-04 -1.2092274E-04
822   7.6297421E+05  1.3185566E-03  6.8270858E-04 -3.5328364E-04 -1.6853589E-04 -6.9348191E-05
823   2.0076699E+06  7.3321294E-04  3.7951322E-04 -1.9637484E-04 -9.3691920E-05 -3.8555676E-05
824   5.6164426E+06  3.8556276E-04  1.9951148E-04 -1.0322691E-04 -4.9245293E-05 -2.0263741E-05
825   1.6802886E+07  1.7968610E-04  9.2969270E-05 -4.8101095E-05 -2.2948263E-05 -9.4433293E-06
826   5.4065393E+07  5.8601596E-05  3.0318645E-05 -1.5686187E-05 -7.4832266E-06 -3.0792590E-06
827
828
829   3.2777475E-02 -9.3316637E-06  7.0000665E-08  1.7578676E-06  2.5432750E-07 -2.2427559E-07
830   6.1052208E-02  4.5611594E-06 -1.0155629E-07 -1.4049180E-06 -4.7480480E-07 -9.5509282E-06
831   1.1371780E-01 -1.5158182E-05  3.0331219E-07  4.4901492E-06  1.2252677E-06 -6.9349211E-05
832   2.1181678E-01 -4.2451680E-06 -4.7050070E-07 -3.5964861E-06 -2.5070434E-06 -7.4556346E-04
833   3.9455421E-01 -2.2395358E-05  1.1587154E-06  1.3135576E-05  1.2921192E-05 -2.5959546E-03
834   7.3499994E-01 -2.4101698E-05 -1.9000805E-06 -1.3264295E-05  1.3219801E-04 -3.3854786E-03
835   1.3694267E+00 -4.7031223E-05  4.3178993E-06  4.0361802E-05  2.1838455E-03  5.1556783E-04
836   2.5522453E+00 -6.8311062E-05 -6.2725455E-06 -8.6157072E-05  8.5954352E-03  6.2321848E-03
837   4.7592004E+00 -1.1523798E-04  1.5580022E-05 -1.3497218E-03  7.4633643E-03  4.9748472E-03
838   8.8821010E+00 -1.1775408E-04 -5.3566907E-05 -1.1851927E-02 -1.0070723E-02 -4.6165537E-03
839   1.6598538E+01 -2.9859355E-04 -1.0682392E-03 -2.0366327E-02 -1.6714489E-02 -7.8238944E-03
840   3.1079528E+01 -4.4960812E-04 -1.2235754E-02  1.7031350E-03 -8.3772517E-04 -4.8139265E-04
841   5.8358006E+01 -3.6322307E-04 -3.3281971E-02  2.8614634E-02  1.5437728E-02  6.4952597E-03
```

```
842   1.1000966E+02  5.8633180E-03 -2.4755158E-02  2.0357568E-02  1.0947083E-02  4.6403333E-03
843   2.0848751E+02  2.7663532E-02  7.9050943E-03 -2.4590451E-03 -9.8714652E-04 -3.9984555E-04
844   3.9793692E+02  4.9278632E-02  2.8466864E-02 -1.5119849E-02 -7.2324632E-03 -2.9709995E-03
845   7.6660359E+02  5.1169865E-02  2.9666664E-02 -1.5802650E-02 -7.6127992E-03 -3.1434241E-03
846   1.4944323E+03  3.9687448E-02  2.2187704E-02 -1.1701297E-02 -5.5999814E-03 -2.3035144E-03
847   2.9570622E+03  2.6194603E-02  1.4300717E-02 -7.4988679E-03 -3.5955499E-03 -1.4826020E-03
848   5.9602824E+03  1.5931698E-02  8.5328085E-03 -4.4517913E-03 -2.1251481E-03 -8.7380651E-04
849   1.2287274E+04  9.3288421E-03  4.9546810E-03 -2.5802520E-03 -1.2347919E-03 -5.0889786E-04
850   2.6025475E+04  5.3842381E-03  2.8338848E-03 -1.4721886E-03 -7.0212833E-04 -2.8866928E-04
851   5.6920168E+04  3.0926864E-03  1.6225415E-03 -8.4241915E-04 -4.0273993E-04 -1.6592330E-04
852   1.2924147E+05  1.7734094E-03  9.2533153E-04 -4.7969863E-04 -2.2872350E-04 -9.4046964E-05
853   3.0640048E+05  1.0104040E-03  5.2642764E-04 -2.7285039E-04 -1.3034549E-04 -5.3681772E-05
854   7.6297421E+05  5.6708693E-04  2.9446980E-04 -1.5247942E-04 -7.2704287E-05 -2.9900411E-05
855   2.0076699E+06  3.0829826E-04  1.5999996E-04 -8.2847907E-05 -3.9556961E-05 -1.6286587E-05
856   5.6164426E+06  1.5771287E-04  8.1695277E-05 -4.2278166E-05 -2.0160475E-05 -8.2925202E-06
857   1.6802886E+07  7.0609820E-05  3.6575817E-05 -1.8929933E-05 -9.0358362E-06 -3.7196921E-06
858   5.4065393E+07  2.1734831E-05  1.1245924E-05 -5.8182964E-06 -2.7744208E-06 -1.1412212E-06
859
860
861                         2P+           3P+           4P+           5P+           6P+
862                -6.1939589E+02 -1.5549183E+02 -3.8038164E+01 -7.7221173E+00 -9.5408788E-01
863
864
865   3.2777475E-02 -2.4336490E-06  1.6754306E-05 -5.5892525E-05 -6.0387643E-05  1.2386475E-03
866   6.1052208E-02  9.3576855E-06 -6.4207914E-05  2.1493368E-04  3.0280989E-04  1.1874241E-02
867   1.1371780E-01 -2.0842321E-05  1.4278822E-04 -4.6618926E-04 -4.7530162E-04  7.7427164E-02
868   2.1181678E-01  3.6697396E-05 -2.4977203E-04  8.0604051E-04  9.0129034E-04  3.2545555E-01
869   3.9455421E-01 -5.7637303E-05  3.8792609E-04 -1.2273094E-03 -4.6882426E-03  5.3826085E-01
870   7.3499994E-01  8.5352209E-05 -5.6655569E-04  1.5800164E-03 -5.2087664E-02  3.1320799E-01
871   1.3694267E+00 -1.2053399E-04  8.0999454E-04 -1.6846906E-03 -3.6568845E-01 -2.3816734E-01
872   2.5522453E+00  1.7779718E-04 -9.4218169E-04  1.2903151E-02 -6.9361708E-01 -4.5706220E-01
873   4.7592004E+00 -2.2790063E-04  1.3715452E-03  1.6619670E-01 -1.6575956E-01 -1.1313649E-01
874   8.8821010E+00  2.7867735E-04 -4.9912777E-03  6.1780391E-01  5.6618084E-01  2.5261361E-01
875   1.6598538E+01 -5.3063638E-04 -7.6655861E-02  4.7545603E-01  3.7594386E-01  1.6531536E-01
876   3.1079528E+01  8.1926120E-04 -3.9443725E-01 -2.6325439E-01 -1.2640907E-01 -5.0282867E-02
877   5.8358006E+01  1.9516933E-02 -5.3111228E-01 -4.6566357E-01 -2.5019762E-01 -1.0025868E-01
878   1.1000966E+02  1.5033101E-01 -1.3594093E-01 -1.2662521E-01 -6.8504877E-02 -2.7729878E-02
879   2.0848751E+02  3.2133747E-01  1.7641008E-01  9.0446983E-02  4.2739129E-02  1.6796378E-02
880   3.9793692E+02  3.2072249E-01  2.0124520E-01  1.0945733E-01  5.2312345E-02  2.0490470E-02
881   7.6660359E+02  2.0050509E-01  1.2068503E-01  6.4829782E-02  3.0961770E-02  1.2158730E-02
882   1.4944323E+03  9.6326899E-02  5.5757202E-02  2.9651607E-02  1.4096148E-02  5.5177320E-03
883   2.9570622E+03  3.9654466E-02  2.2415263E-02  1.1842862E-02  5.6316919E-03  2.2095082E-03
884   5.9602824E+03  1.4971482E-02  8.3261254E-03  4.3824962E-03  2.0776849E-03  8.1267403E-04
885   1.2287274E+04  5.3674900E-03  2.9641098E-03  1.5566798E-03  7.3927073E-04  2.9002714E-04
886   2.6025475E+04  1.8661840E-03  1.0233010E-03  5.3671985E-04  2.5416668E-04  9.9365917E-05
887   5.6920168E+04  6.3461305E-04  3.4729890E-04  1.8196989E-04  8.6386092E-05  3.3896608E-05
888   1.2924147E+05  2.1249834E-04  1.1571279E-04  6.0588224E-05  2.8673602E-05  1.1205637E-05
889   3.0640048E+05  6.9752539E-05  3.7968687E-05  1.9866465E-05  9.4297765E-06  3.7008695E-06
890   7.6297421E+05  2.2397871E-05  1.2139640E-05  6.3493838E-06  3.0032946E-06  1.1732181E-06
891   2.0076699E+06  6.9131086E-06  3.7504542E-06  1.9605527E-06  9.3071395E-07  3.6543963E-07
892   5.6164426E+06  2.0594701E-06  1.1127777E-06  5.8164270E-07  2.7491428E-07  1.0730215E-07
893   1.6802886E+07  5.5135401E-07  2.9875011E-07  1.5608727E-07  7.4164034E-08  2.9160511E-08
894   5.4065393E+07  1.4879784E-07  8.0244195E-08  4.1931736E-08  1.9799755E-08  7.7181920E-09
895
896
897   3.2777475E-02 -1.3207878E-09  3.4965780E-08 -7.5446197E-08 -8.3100511E-08  1.8324502E-06
898   6.1052208E-02  1.0946486E-08 -1.6565764E-07  4.0833249E-07  5.8897280E-07  2.3927737E-05
899   1.1371780E-01 -4.0803138E-08  4.7032525E-07 -1.2299178E-06 -1.2621203E-06  2.1304902E-04
900   2.1181678E-01  1.0985682E-07 -1.0728611E-06  2.9362395E-06  3.3058064E-06  1.2220435E-03
901   3.9455421E-01 -2.5276752E-07  2.2024214E-06 -6.1486831E-06 -2.3905280E-05  2.7585314E-03
902   7.3499994E-01  5.3549677E-07 -4.2892502E-06  1.0840453E-05 -3.6457970E-04  2.1905696E-03
903   1.3694267E+00 -1.0650882E-06  8.2210039E-06 -1.5772498E-05 -3.4916695E-03 -2.2739744E-03
904   2.5522453E+00  2.2017354E-06 -1.3001460E-05  1.6782632E-04 -9.0413702E-03 -5.9569861E-03
905   4.7592004E+00 -3.9021382E-06  2.5481600E-05  2.9612675E-03 -2.9460697E-03 -2.0109624E-03
906   8.8821010E+00  6.5774673E-06 -1.2317379E-04  1.5028180E-02  1.3765672E-02  6.1415481E-03
907   1.6598538E+01 -1.7395834E-05 -2.5543939E-03  1.5787682E-02  1.2475252E-02  5.4850895E-03
908   3.1079528E+01  3.7061187E-05 -1.7988429E-02 -1.2006348E-02 -5.7712025E-03 -2.2961420E-03
909   5.8358006E+01  1.2348390E-03 -3.3073785E-02 -2.8932698E-02 -1.5538574E-02 -6.2260449E-03
910   1.1000966E+02  1.3011629E-02 -1.1426030E-02 -1.0640642E-02 -5.7553138E-03 -2.3297340E-03
911   2.0848751E+02  3.8062766E-02  2.0811109E-02  1.0682682E-02  5.0499788E-03  1.9847382E-03
912   3.9793692E+02  5.1851468E-02  3.2230331E-02  1.7496779E-02  8.3583775E-03  3.2735798E-03
913   7.6660359E+02  4.4072294E-02  2.6230738E-02  1.4053605E-02  6.7075174E-03  2.6337808E-03
914   1.4944323E+03  2.8624950E-02  1.6364740E-02  8.6772510E-03  4.1218999E-03  1.6130929E-03
915   2.9570622E+03  1.5770993E-02  8.8005603E-03  4.6349164E-03  2.2024914E-03  8.6408133E-04
916   5.9602824E+03  7.8639115E-03  4.3171518E-03  2.2652863E-03  1.0729240E-03  4.1949616E-04
917   1.2287274E+04  3.6559185E-03  1.9947426E-03  1.0443832E-03  4.9574149E-04  1.9453650E-04
918   2.6025475E+04  1.6113476E-03  8.7355432E-04  4.5694978E-04  2.1613580E-04  8.4425446E-05
919   5.6920168E+04  6.7497388E-04  3.6607705E-04  1.9133996E-04  9.0852090E-05  3.5684824E-05
920   1.2924147E+05  2.6989658E-04  1.4571912E-04  7.6161518E-05  3.5980911E-05  1.4034697E-05
921   3.0640048E+05  1.0215312E-04  5.5349847E-05  2.8914350E-05  1.3745415E-05  5.4085524E-06
922   7.6297421E+05  3.6629771E-05  1.9732887E-05  1.0312021E-05  4.8624567E-06  1.8917339E-06
923   2.0076699E+06  1.2254330E-05  6.6477539E-06  3.4717529E-06  1.6546375E-06  6.5339148E-07
924   5.6164426E+06  3.8316748E-06  2.0582388E-06  1.0757435E-06  5.0527204E-07  1.9549628E-07
925   1.6802886E+07  1.0835857E-06  5.8992873E-07  3.0797781E-07  1.4759391E-07  5.8723790E-08
926   5.4065393E+07  2.3568386E-07  1.2586494E-07  6.5813885E-08  3.0642650E-08  1.1707548E-08
927
928
929                         3D-           4D-           5D-           6D-
930                -1.3454927E+02 -2.8439539E+01 -4.0951554E+00 -1.9053474E-01
931
932
933   3.2777475E-02  3.2651074E-07 -2.6571957E-05 -7.1019724E-05  8.9491015E-02
```

```
934   6.1052208E-02 -1.1894050E-06  9.1843839E-05  1.7552722E-04  1.8654044E-01
935   1.1371780E-01  2.4625831E-06 -1.9297146E-04 -5.2234218E-04  3.0652101E-01
936   2.1181678E-01 -4.0986129E-06  3.2129889E-04 -2.7417100E-04  3.2033054E-01
937   3.9455421E-01  6.1043128E-06 -5.0589439E-04 -1.4563385E-02  2.5061410E-01
938   7.3499994E-01 -8.9341665E-06  6.0227820E-04 -1.0696676E-01  8.9685454E-02
939   1.3694267E+00  9.7845165E-06 -7.2930002E-04 -3.4928272E-01 -9.3657675E-02
940   2.5522453E+00  6.1678263E-06  1.2297876E-02 -4.9298781E-01 -1.6224526E-01
941   4.7592004E+00 -9.5122511E-05  1.2306050E-01 -2.2981896E-01 -7.5638494E-02
942   8.8821010E+00 -1.7044696E-03  4.0358527E-01  1.9020216E-01  4.6878442E-02
943   1.6598538E+01 -3.3490722E-02  4.5955502E-01  2.6898485E-01  7.0909914E-02
944   3.1079528E+01 -1.7976659E-01  1.4192470E-01  9.4582711E-02  2.5010988E-02
945   5.8358006E+01 -3.5445132E-01 -1.5547080E-01 -6.1021210E-02 -1.5051975E-02
946   1.1000966E+02 -3.3068223E-01 -1.8988192E-01 -8.3626235E-02 -2.1126171E-02
947   2.0848751E+02 -1.9048277E-01 -1.1074452E-01 -4.8575649E-02 -1.2239840E-02
948   3.9793692E+02 -8.1562455E-02 -4.7076496E-02 -2.0731398E-02 -5.2359922E-03
949   7.6660359E+02 -2.9130071E-02 -1.6474611E-02 -7.1776740E-03 -1.8063717E-03
950   1.4944323E+03 -9.6393649E-03 -5.4487794E-03 -2.3916031E-03 -6.0404305E-04
951   2.9570622E+03 -3.0667014E-03 -1.7080815E-03 -7.4001605E-04 -1.8594553E-04
952   5.9602824E+03 -1.0114887E-03 -5.6807403E-04 -2.4954728E-04 -6.3083197E-05
953   1.2287274E+04 -3.5232458E-04 -1.9484679E-04 -8.4131236E-05 -2.1116516E-05
954   2.6025475E+04 -1.2123834E-04 -6.7931142E-05 -2.9858569E-05 -7.5513692E-06
955   5.6920168E+04 -5.1206264E-05 -2.8281789E-05 -1.2218172E-05 -3.0678055E-06
956   1.2924147E+05 -1.2423186E-05 -6.9580184E-06 -3.0604736E-06 -7.7428947E-07
957   3.0640048E+05 -1.0166505E-05 -5.6313512E-06 -2.4435258E-06 -6.1470251E-07
958
959
960   3.2777475E-02  5.1272654E-08 -7.1337975E-08 -9.8695082E-08  1.5640040E-04
961   6.1052208E-02  9.4368734E-09  1.9756252E-07  4.1408779E-07  4.4494135E-04
962   1.1371780E-01  1.5149148E-07 -6.8163196E-07 -1.6165852E-06  9.9782587E-04
963   2.1181678E-01  1.7093311E-07  1.2680016E-06 -1.1699505E-06  1.4231482E-03
964   3.9455421E-01  5.8354045E-07 -3.3044203E-06 -8.8017777E-05  1.5196087E-03
965   7.3499994E-01  9.5286576E-07  4.2711904E-06 -8.8497625E-04  7.4213549E-04
966   1.3694267E+00  2.5613533E-06 -9.4734702E-06 -3.9447915E-03 -1.0579134E-03
967   2.5522453E+00  5.1356596E-06  1.8658384E-04 -7.6009953E-03 -2.5015174E-03
968   4.7592004E+00  9.4183882E-06  2.5869202E-03 -4.8349100E-03 -1.5912555E-03
969   8.8821010E+00 -2.4846844E-05  1.1600744E-02  5.4711831E-03  1.3487348E-03
970   1.6598538E+01 -1.2668094E-03  1.8043647E-02  1.0560150E-02  2.7838624E-03
971   3.1079528E+01 -9.5791864E-03  7.5931119E-03  5.0588871E-03  1.3376943E-03
972   5.8358006E+01 -2.5906001E-02 -1.1405232E-02 -4.4834319E-03 -1.1062167E-03
973   1.1000966E+02 -3.2939316E-02 -1.8940614E-02 -8.3445574E-03 -2.1081864E-03
974   2.0848751E+02 -2.5687730E-02 -1.4944831E-02 -6.5558307E-03 -1.6518885E-03
975   3.9793692E+02 -1.4730133E-02 -8.5051338E-03 -3.7462797E-03 -9.4625040E-04
976   7.6660359E+02 -6.8893687E-03 -3.8966271E-03 -1.6970918E-03 -4.2702250E-04
977   1.4944323E+03 -2.8874401E-03 -1.6343513E-03 -7.1820842E-04 -1.8148297E-04
978   2.9570622E+03 -1.0971088E-03 -6.1041945E-04 -2.6378718E-04 -6.6204885E-05
979   5.9602824E+03 -4.1131546E-04 -2.3233303E-04 -1.0267240E-04 -2.6017839E-05
980   1.2287274E+04 -1.4526502E-04 -7.9726162E-05 -3.4004690E-05 -8.4885749E-06
981   2.6025475E+04 -5.3097540E-05 -3.0262169E-05 -1.3565060E-05 -3.4583031E-06
982   5.6920168E+04 -1.8273467E-05 -9.8440828E-06 -4.1025343E-06 -1.0137506E-06
983   1.2924147E+05 -6.0357247E-06 -3.5020021E-06 -1.6064965E-06 -4.1341700E-07
984   3.0640048E+05 -2.7989375E-06 -1.5096828E-06 -6.3118668E-07 -1.5620812E-07
985
986
987                         3D+            4D+            5D+            6D+
988            -1.2827939E+02 -2.6928655E+01 -3.8101969E+00 -1.8169381E-01
989
990
991   3.2777475E-02 -4.4802587E-07  2.4275314E-05 -7.5317133E-05 -1.0707794E-01
992   6.1052208E-02  1.6224646E-06 -8.3565772E-05  1.6617400E-04 -1.9937753E-01
993   1.1371780E-01 -3.3669023E-06  1.7587412E-04 -5.8806415E-04 -3.1991976E-01
994   2.1181678E-01  5.6351388E-06 -2.9003014E-04 -7.3087509E-04 -3.1570572E-01
995   3.9455421E-01 -8.3286912E-06  4.6510025E-04 -1.8676569E-02 -2.3125690E-01
996   7.3499994E-01  1.1432832E-05 -5.5581886E-04 -1.2503947E-01 -6.3047095E-02
997   1.3694267E+00 -1.5216574E-05  4.6523033E-04 -3.7416157E-01  1.0510482E-01
998   2.5522453E+00 -6.0920422E-06 -1.5488351E-02 -4.8552690E-01  1.5427095E-01
999   4.7592004E+00  1.3841937E-04 -1.4245950E-01 -1.8761814E-01  6.0618295E-02
1000  8.8821010E+00  2.2622365E-03 -4.2633114E-01  2.1406810E-01 -5.2331763E-02
1001  1.6598538E+01  4.0492708E-02 -4.4457077E-01  2.5889309E-01 -6.6387969E-02
1002  3.1079528E+01  1.9887657E-01 -1.0644157E-01  7.5275550E-02 -1.9454885E-02
1003  5.8358006E+01  3.6589928E-01  1.7260955E-01 -6.9603820E-02  1.6825684E-02
1004  1.1000966E+02  3.2180831E-01  1.8693843E-01 -8.1980897E-02  2.0168569E-02
1005  2.0848751E+02  1.7529852E-01  1.0234078E-01 -4.4647524E-02  1.0953251E-02
1006  3.9793692E+02  7.0461699E-02  4.0733158E-02 -1.7825555E-02  4.3835269E-03
1007  7.6660359E+02  2.2927985E-02  1.2979953E-02 -5.6181141E-03  1.3760728E-03
1008  1.4944323E+03  6.6296509E-03  3.7587050E-03 -1.6416421E-03  4.0398079E-04
1009  2.9570622E+03  1.7162680E-03  9.5520955E-04 -4.0992766E-04  1.0009302E-04
1010  5.9602824E+03  4.3584798E-04  2.4710517E-04 -1.0871783E-04  2.6857714E-05
1011  1.2287274E+04  1.0119312E-04  5.5350430E-05 -2.3344938E-05  5.6550027E-06
1012  2.6025475E+04  2.4512440E-05  1.4120973E-05 -6.3442033E-06  1.5822968E-06
1013  5.6920168E+04  5.1949414E-06  2.7463397E-06 -1.1095423E-06  2.6310949E-07
1014  1.2924147E+05  1.1521615E-06  6.8995617E-07 -3.2368921E-07  8.2241922E-08
1015  3.0640048E+05  3.0400933E-07  1.5808078E-07 -6.2498362E-08  1.4655020E-08
1016
1017
1018  3.2777475E-02 -2.5112890E-09  4.1934348E-08 -1.3098950E-07 -1.8714305E-04
1019  6.1052208E-02  9.5532465E-09 -1.9775127E-07  3.9426617E-07 -4.7556643E-04
1020  1.1371780E-01 -2.2638410E-08  5.6940672E-07 -1.9101004E-06 -1.0414533E-03
1021  2.1181678E-01  4.5049675E-08 -1.2834203E-06 -3.2547256E-06 -1.4026322E-03
1022  3.9455421E-01 -8.2213089E-08  2.8124617E-06 -1.1324607E-04 -1.4022465E-03
1023  7.3499994E-01  1.4317164E-07 -4.5883166E-06 -1.0349402E-03 -5.2172390E-04
1024  1.3694267E+00 -2.4505475E-07  5.2366233E-06 -4.2269885E-03  1.1873505E-03
1025  2.5522453E+00  1.5406900E-08 -2.3897014E-04 -7.4874326E-03  2.3788950E-03
```

104

```
1026    4.7592004E+00   2.7528844E-06  -3.0018144E-03  -3.9485289E-03   1.2757083E-03
1027    8.8821010E+00   6.5489371E-05  -1.2268290E-02   6.1592757E-03  -1.5057738E-03
1028    1.6598538E+01   1.5963932E-03  -1.7471504E-02   1.0170578E-02  -2.6079155E-03
1029    3.1079528E+01   1.0724050E-02  -5.6943677E-03   4.0295907E-03  -1.0414543E-03
1030    5.8358006E+01   2.6975934E-02   1.2723409E-02  -5.1328540E-03   1.2409059E-03
1031    1.1000966E+02   3.2419769E-02   1.8801788E-02  -8.2428979E-03   2.0278041E-03
1032    2.0848751E+02   2.4114276E-02   1.4046415E-02  -6.1248998E-03   1.5024841E-03
1033    3.9793692E+02   1.3197018E-02   7.6097968E-03  -3.3285423E-03   8.1849208E-04
1034    7.6660359E+02   5.8221774E-03   3.3282557E-03  -1.4212514E-03   3.4805303E-04
1035    1.4944323E+03   2.2707250E-03   1.2839997E-03  -5.6067292E-04   1.3798793E-04
1036    2.9570622E+03   7.8370384E-04   4.3440158E-04  -1.8606588E-04   4.5400882E-05
1037    5.9602824E+03   2.6332078E-04   1.4918807E-04  -6.5774960E-05   1.6268067E-05
1038    1.2287274E+04   7.8488463E-05   4.2535433E-05  -1.7781357E-05   4.2895664E-06
1039    2.6025475E+04   2.4632134E-05   1.4332594E-05  -6.5269824E-06   1.6378398E-06
1040    5.6920168E+04   6.0733526E-06   3.0796907E-06  -1.1747621E-06   2.7012610E-07
1041    1.2924147E+05   1.8431209E-06   1.1573922E-06  -5.7003982E-07   1.4770355E-07
1042    3.0640048E+05   4.3591834E-07   2.0024127E-07  -6.4421032E-08   1.3266370E-08
1043
1044
1045                             4F-         5F-
1046                      -1.4283216E+01 -2.9142420E-01
1047
1048
1049    3.2777475E-02  -1.0725471E-05   1.8335167E-03
1050    6.1052208E-02   3.7371756E-05   1.4299458E-02
1051    1.1371780E-01  -6.5117205E-05   5.1474113E-02
1052    2.1181678E-01   1.3410944E-04   1.1826153E-01
1053    3.9455421E-01  -1.5304599E-04   2.0834794E-01
1054    7.3499994E-01   3.9034287E-04   2.8704100E-01
1055    1.3694267E+00   1.8711942E-03   3.1098500E-01
1056    2.5522453E+00   2.9485902E-02   2.3916164E-01
1057    4.7592004E+00   1.3401088E-01   9.8299863E-02
1058    8.8821010E+00   2.9018994E-01  -3.8916120E-02
1059    1.6598538E+01   3.4836741E-01  -9.0866778E-02
1060    3.1079528E+01   2.6972459E-01  -7.6747815E-02
1061    5.8358006E+01   1.4163449E-01  -4.1716302E-02
1062    1.1000966E+02   5.3462215E-02  -1.5589750E-02
1063    2.0848751E+02   1.6463172E-02  -4.8330990E-03
1064    3.9793692E+02   4.0383670E-03  -1.1670891E-03
1065    7.6660359E+02   9.6404242E-04  -2.8455384E-04
1066    1.4944323E+03   2.0990334E-04  -5.9438911E-05
1067    2.9570622E+03   4.2739189E-05  -1.2950119E-05
1068    5.9602824E+03   1.5565905E-05  -4.3943503E-06
1069
1070
1071    3.2777475E-02  -2.0783188E-08   3.6300411E-06
1072    6.1052208E-02   9.3085995E-08   3.8674853E-05
1073    1.1371780E-01  -2.3659580E-07   1.8998293E-04
1074    2.1181678E-01   6.3262257E-07   5.9574736E-04
1075    3.9455421E-01  -1.0880802E-06   1.4323838E-03
1076    7.3499994E-01   3.4104055E-06   2.6933700E-03
1077    1.3694267E+00   2.3452890E-05   3.9826033E-03
1078    2.5522453E+00   5.1386501E-04   4.1805895E-03
1079    4.7592004E+00   3.1956728E-03   2.3449523E-03
1080    8.8821010E+00   9.4515168E-03  -1.2688828E-03
1081    1.6598538E+01   1.5493324E-02  -4.0424851E-03
1082    3.1079528E+01   1.6367583E-02  -4.6582595E-03
1083    5.8358006E+01   1.1712012E-02  -3.4499746E-03
1084    1.1000966E+02   6.0012126E-03  -1.7499308E-03
1085    2.0848751E+02   2.4902691E-03  -7.3123932E-04
1086    3.9793692E+02   8.0917742E-04  -2.3369431E-04
1087    7.6660359E+02   2.5236369E-04  -7.4645254E-05
1088    1.4944323E+03   6.5321088E-05  -1.8336818E-05
1089    2.9570622E+03   1.7865583E-05  -5.4973542E-06
1090    5.9602824E+03   5.2672434E-06  -1.4302192E-06
1091
1092
1093                             4F+         5F+
1094                      -1.3898700E+01 -2.6933365E-01
1095
1096
1097    3.2777475E-02   1.1675980E-05  -2.6792931E-03
1098    6.1052208E-02  -4.1377000E-05  -1.7785577E-02
1099    1.1371780E-01   7.0857124E-05  -5.8567071E-02
1100    2.1181678E-01  -1.4708650E-04  -1.2737380E-01
1101    3.9455421E-01   1.6844262E-04  -2.1546329E-01
1102    7.3499994E-01  -4.2699706E-04  -2.8794124E-01
1103    1.3694267E+00  -2.1767351E-03  -3.0639932E-01
1104    2.5522453E+00  -3.2039098E-02  -2.3140727E-01
1105    4.7592004E+00  -1.4005668E-01  -8.9645492E-02
1106    8.8821010E+00  -2.9608946E-01   4.2772328E-02
1107    1.6598538E+01  -3.4809129E-01   9.0168542E-02
1108    3.1079528E+01  -2.6411894E-01   7.4324210E-02
1109    5.8358006E+01  -1.3656877E-01   3.9659289E-02
1110    1.1000966E+02  -5.0325596E-02   1.4467214E-02
1111    2.0848751E+02  -1.4919582E-02   4.3176435E-03
1112    3.9793692E+02  -3.4681534E-03   9.8742386E-04
1113    7.6660359E+02  -7.6780228E-04   2.2383383E-04
1114    1.4944323E+03  -1.3866054E-04   3.8326704E-05
1115    2.9570622E+03  -2.6734026E-05   8.1305216E-06
1116    5.9602824E+03  -5.2648888E-06   1.3970692E-06
1117
```

```
   3.2777475E-02  2.4030104E-08 -5.3099163E-06
   6.1052208E-02 -1.1468771E-07 -4.8102747E-05
   1.1371780E-01  2.6715090E-07 -2.1618653E-04
   2.1181678E-01 -7.5061919E-07 -6.4167480E-04
   3.9455421E-01  1.1736847E-06 -1.4814308E-03
   7.3499994E-01 -4.0316229E-06 -2.7020456E-03
   1.3694267E+00 -2.7854269E-05 -3.9245322E-03
   2.5522453E+00 -5.6048759E-04 -4.0459078E-03
   4.7592004E+00 -3.3449889E-03 -2.1394404E-03
   8.8821010E+00 -9.6582637E-03  1.3958154E-03
   1.6598538E+01 -1.5510672E-02  4.0174278E-03
   3.1079528E+01 -1.6082860E-02  4.5249404E-03
   5.8358006E+01 -1.1361507E-02  3.2985776E-03
   1.1000966E+02 -5.7154982E-03  1.6425083E-03
   2.0848751E+02 -2.3118320E-03  6.6885753E-04
   3.9793692E+02 -7.2978324E-04  2.0764163E-04
   7.6660359E+02 -2.1946679E-04  6.4005928E-05
   1.4944323E+03 -5.2827097E-05  1.4546466E-05
   2.9570622E+03 -1.4013535E-05  4.2861177E-06
   5.9602824E+03 -3.3747959E-06  8.7670867E-07


 Calculated energy =   -2.7274377546109416E+04
 Literature energy =   -2.7274378906250000E+04
 Difference (cal - lit) = 1.36E-03
 Relative error (abs((cal - lit)/lit)) = 4.99E-08
 Basis set for original program
35 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   3.2777474514072306E-02 1.0d0 1.0d0
   6.1052208024089318E-02 1.0d0 1.0d0
   1.1371779544172864E-01 1.0d0 1.0d0
   2.1181678224305334E-01 1.0d0 1.0d0
   3.9455421330447510E-01 1.0d0 1.0d0
   7.3499993917294237E-01 1.0d0 1.0d0
   1.3694267119018748E+00 1.0d0 1.0d0
   2.5522453323068639E+00 1.0d0 1.0d0
   4.7592004160127050E+00 1.0d0 1.0d0
   8.8821010163462422E+00 1.0d0 1.0d0
   1.6598538397889143E+01 1.0d0 1.0d0
   3.1079527707765703E+01 1.0d0 1.0d0
   5.8358006063602978E+01 1.0d0 1.0d0
   1.1000966104096939E+02 1.0d0 1.0d0
   2.0848750508658674E+02 1.0d0 1.0d0
   3.9793692474094433E+02 1.0d0 1.0d0
   7.6660358942891128E+02 1.0d0 1.0d0
   1.4944323346681333E+03 1.0d0 1.0d0
   2.9570621576747585E+03 1.0d0 1.0d0
   5.9602824302208819E+03 1.0d0 1.0d0
   1.2287273922270530E+04 1.0d0 1.0d0
   2.6025474503929181E+04 1.0d0 1.0d0
   5.6920167949008617E+04 1.0d0 1.0d0
   1.2924146561822579E+05 1.0d0 1.0d0
   3.0640047899595002E+05 1.0d0 1.0d0
   7.6297421342751326E+05 1.0d0 1.0d0
   2.0076698631188064E+06 1.0d0 1.0d0
   5.6164425941082174E+06 1.0d0 1.0d0
   1.6802885560418982E+07 1.0d0 1.0d0
   5.4065392869313173E+07 1.0d0 1.0d0
   1.8809445122304142E+08 1.0d0 1.0d0
   7.1100930633447683E+08 1.0d0 1.0d0
   2.9330970963636436E+09 1.0d0 1.0d0
   1.3255935848588299E+10 1.0d0 1.0d0
   6.5852664686153687E+10 1.0d0 1.0d0
30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   3.2777474514072306E-02 1.0d0 1.0d0
   6.1052208024089318E-02 1.0d0 1.0d0
   1.1371779544172864E-01 1.0d0 1.0d0
   2.1181678224305334E-01 1.0d0 1.0d0
   3.9455421330447510E-01 1.0d0 1.0d0
   7.3499993917294237E-01 1.0d0 1.0d0
   1.3694267119018748E+00 1.0d0 1.0d0
   2.5522453323068639E+00 1.0d0 1.0d0
   4.7592004160127050E+00 1.0d0 1.0d0
   8.8821010163462422E+00 1.0d0 1.0d0
   1.6598538397889143E+01 1.0d0 1.0d0
   3.1079527707765703E+01 1.0d0 1.0d0
   5.8358006063602978E+01 1.0d0 1.0d0
   1.1000966104096939E+02 1.0d0 1.0d0
   2.0848750508658674E+02 1.0d0 1.0d0
   3.9793692474094433E+02 1.0d0 1.0d0
   7.6660358942891128E+02 1.0d0 1.0d0
   1.4944323346681333E+03 1.0d0 1.0d0
   2.9570621576747585E+03 1.0d0 1.0d0
   5.9602824302208819E+03 1.0d0 1.0d0
   1.2287273922270530E+04 1.0d0 1.0d0
   2.6025474503929181E+04 1.0d0 1.0d0
   5.6920167949008617E+04 1.0d0 1.0d0
   1.2924146561822579E+05 1.0d0 1.0d0
   3.0640047899595002E+05 1.0d0 1.0d0
   7.6297421342751326E+05 1.0d0 1.0d0
   2.0076698631188064E+06 1.0d0 1.0d0
```

```
1210    5.6164425941082174E+06 1.0d0 1.0d0
1211    1.6802885560418982E+07 1.0d0 1.0d0
1212    5.4065392869313173E+07 1.0d0 1.0d0
1213 30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1214    3.2777474514072306E-02 1.0d0 1.0d0
1215    6.1052208024089318E-02 1.0d0 1.0d0
1216    1.1371779544172864E-01 1.0d0 1.0d0
1217    2.1181678224305334E-01 1.0d0 1.0d0
1218    3.9455421330447510E-01 1.0d0 1.0d0
1219    7.3499993917294237E-01 1.0d0 1.0d0
1220    1.3694267119018748E+00 1.0d0 1.0d0
1221    2.5522453323068639E+00 1.0d0 1.0d0
1222    4.7592004160127050E+00 1.0d0 1.0d0
1223    8.8821010163462422E+00 1.0d0 1.0d0
1224    1.6598538397889143E+01 1.0d0 1.0d0
1225    3.1079527707765703E+01 1.0d0 1.0d0
1226    5.8358006063602978E+01 1.0d0 1.0d0
1227    1.1000966104096939E+02 1.0d0 1.0d0
1228    2.0848750508658674E+02 1.0d0 1.0d0
1229    3.9793692474094433E+02 1.0d0 1.0d0
1230    7.6660358942891128E+02 1.0d0 1.0d0
1231    1.4944323346681333E+03 1.0d0 1.0d0
1232    2.9570621576747585E+03 1.0d0 1.0d0
1233    5.9602824302208819E+03 1.0d0 1.0d0
1234    1.2287273922270530E+04 1.0d0 1.0d0
1235    2.6025474503929181E+04 1.0d0 1.0d0
1236    5.6920167949008617E+04 1.0d0 1.0d0
1237    1.2924146561822579E+05 1.0d0 1.0d0
1238    3.0640047899595002E+05 1.0d0 1.0d0
1239    7.6297421342751326E+05 1.0d0 1.0d0
1240    2.0076698631188064E+06 1.0d0 1.0d0
1241    5.6164425941082174E+06 1.0d0 1.0d0
1242    1.6802885560418982E+07 1.0d0 1.0d0
1243    5.4065392869313173E+07 1.0d0 1.0d0
1244 25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1245    3.2777474514072306E-02 1.0d0 1.0d0
1246    6.1052208024089318E-02 1.0d0 1.0d0
1247    1.1371779544172864E-01 1.0d0 1.0d0
1248    2.1181678224305334E-01 1.0d0 1.0d0
1249    3.9455421330447510E-01 1.0d0 1.0d0
1250    7.3499993917294237E-01 1.0d0 1.0d0
1251    1.3694267119018748E+00 1.0d0 1.0d0
1252    2.5522453323068639E+00 1.0d0 1.0d0
1253    4.7592004160127050E+00 1.0d0 1.0d0
1254    8.8821010163462422E+00 1.0d0 1.0d0
1255    1.6598538397889143E+01 1.0d0 1.0d0
1256    3.1079527707765703E+01 1.0d0 1.0d0
1257    5.8358006063602978E+01 1.0d0 1.0d0
1258    1.1000966104096939E+02 1.0d0 1.0d0
1259    2.0848750508658674E+02 1.0d0 1.0d0
1260    3.9793692474094433E+02 1.0d0 1.0d0
1261    7.6660358942891128E+02 1.0d0 1.0d0
1262    1.4944323346681333E+03 1.0d0 1.0d0
1263    2.9570621576747585E+03 1.0d0 1.0d0
1264    5.9602824302208819E+03 1.0d0 1.0d0
1265    1.2287273922270530E+04 1.0d0 1.0d0
1266    2.6025474503929181E+04 1.0d0 1.0d0
1267    5.6920167949008617E+04 1.0d0 1.0d0
1268    1.2924146561822579E+05 1.0d0 1.0d0
1269    3.0640047899595002E+05 1.0d0 1.0d0
1270 25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1271    3.2777474514072306E-02 1.0d0 1.0d0
1272    6.1052208024089318E-02 1.0d0 1.0d0
1273    1.1371779544172864E-01 1.0d0 1.0d0
1274    2.1181678224305334E-01 1.0d0 1.0d0
1275    3.9455421330447510E-01 1.0d0 1.0d0
1276    7.3499993917294237E-01 1.0d0 1.0d0
1277    1.3694267119018748E+00 1.0d0 1.0d0
1278    2.5522453323068639E+00 1.0d0 1.0d0
1279    4.7592004160127050E+00 1.0d0 1.0d0
1280    8.8821010163462422E+00 1.0d0 1.0d0
1281    1.6598538397889143E+01 1.0d0 1.0d0
1282    3.1079527707765703E+01 1.0d0 1.0d0
1283    5.8358006063602978E+01 1.0d0 1.0d0
1284    1.1000966104096939E+02 1.0d0 1.0d0
1285    2.0848750508658674E+02 1.0d0 1.0d0
1286    3.9793692474094433E+02 1.0d0 1.0d0
1287    7.6660358942891128E+02 1.0d0 1.0d0
1288    1.4944323346681333E+03 1.0d0 1.0d0
1289    2.9570621576747585E+03 1.0d0 1.0d0
1290    5.9602824302208819E+03 1.0d0 1.0d0
1291    1.2287273922270530E+04 1.0d0 1.0d0
1292    2.6025474503929181E+04 1.0d0 1.0d0
1293    5.6920167949008617E+04 1.0d0 1.0d0
1294    1.2924146561822579E+05 1.0d0 1.0d0
1295    3.0640047899595002E+05 1.0d0 1.0d0
1296 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1297    3.2777474514072306E-02 1.0d0 1.0d0
1298    6.1052208024089318E-02 1.0d0 1.0d0
1299    1.1371779544172864E-01 1.0d0 1.0d0
1300    2.1181678224305334E-01 1.0d0 1.0d0
1301    3.9455421330447510E-01 1.0d0 1.0d0
```

```
1302    7.3499993917294237E-01 1.0d0 1.0d0
1303    1.3694267119018748E+00 1.0d0 1.0d0
1304    2.5522453323068639E+00 1.0d0 1.0d0
1305    4.7592004160127050E+00 1.0d0 1.0d0
1306    8.8821010163462422E+00 1.0d0 1.0d0
1307    1.6598538397889143E+01 1.0d0 1.0d0
1308    3.1079527707765703E+01 1.0d0 1.0d0
1309    5.8358006063602978E+01 1.0d0 1.0d0
1310    1.1000966104096939E+02 1.0d0 1.0d0
1311    2.0848750508658674E+02 1.0d0 1.0d0
1312    3.9793692474094433E+02 1.0d0 1.0d0
1313    7.6660358942891128E+02 1.0d0 1.0d0
1314    1.4944323346681333E+03 1.0d0 1.0d0
1315    2.9570621576747585E+03 1.0d0 1.0d0
1316    5.9602824302208819E+03 1.0d0 1.0d0
1317 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1318    3.2777474514072306E-02 1.0d0 1.0d0
1319    6.1052208024089318E-02 1.0d0 1.0d0
1320    1.1371779544172864E-01 1.0d0 1.0d0
1321    2.1181678224305334E-01 1.0d0 1.0d0
1322    3.9455421330447510E-01 1.0d0 1.0d0
1323    7.3499993917294237E-01 1.0d0 1.0d0
1324    1.3694267119018748E+00 1.0d0 1.0d0
1325    2.5522453323068639E+00 1.0d0 1.0d0
1326    4.7592004160127050E+00 1.0d0 1.0d0
1327    8.8821010163462422E+00 1.0d0 1.0d0
1328    1.6598538397889143E+01 1.0d0 1.0d0
1329    3.1079527707765703E+01 1.0d0 1.0d0
1330    5.8358006063602978E+01 1.0d0 1.0d0
1331    1.1000966104096939E+02 1.0d0 1.0d0
1332    2.0848750508658674E+02 1.0d0 1.0d0
1333    3.9793692474094433E+02 1.0d0 1.0d0
1334    7.6660358942891128E+02 1.0d0 1.0d0
1335    1.4944323346681333E+03 1.0d0 1.0d0
1336    2.9570621576747585E+03 1.0d0 1.0d0
1337    5.9602824302208819E+03 1.0d0 1.0d0
```

Listing A.4: *Output file for Pa basis set optimization calculation.*

A.3   Single Point Energy using a predefined Basis Set

This input file will perform a single point energy calculation on protactinium using the basis optimized in Section A.2.

```
1  91Pa
2   $contrl jobtype='energy' $end
3   $nuc znuc=91 nucmdl=3 $end
4   $bas nsym=7  bastype='rdin' $end
5  35
6     3.2777474514072306E-02
7     6.1052208024089318E-02
8     1.1371779544172864E-01
9     2.1181678224305334E-01
10    3.9455421330447510E-01
11    7.3499993917294237E-01
12    1.3694267119018748E+00
13    2.5522453323068639E+00
14    4.7592004160127050E+00
15    8.8821010163462422E+00
16    1.6598538397889143E+01
17    3.1079527707765703E+01
18    5.8358006063602978E+01
19    1.1000966104096939E+02
20    2.0848750508658674E+02
21    3.9793692474094433E+02
22    7.6660358942891128E+02
23    1.4944323346681333E+03
24    2.9570621576747585E+03
25    5.9602824302208819E+03
26    1.2287273922270530E+04
27    2.6025474503929181E+04
28    5.6920167949008617E+04
29    1.2924146561822579E+05
30    3.0640047899595002E+05
31    7.6297421342751326E+05
32    2.0076698631188064E+06
33    5.6164425941082174E+06
34    1.6802885560418982E+07
35    5.4065392869313173E+07
36    1.8809445122304142E+08
37    7.1100930633447683E+08
38    2.9330970963636436E+09
39    1.3255935848588299E+10
40    6.5852664686153687E+10
```

108

```
41  30
42      3.2777474514072306E-02
43      6.1052208024089318E-02
44      1.1371779544172864E-01
45      2.1181678224305334E-01
46      3.9455421330447510E-01
47      7.3499993917294237E-01
48      1.3694267119018748E+00
49      2.5522453323068639E+00
50      4.7592004160127050E+00
51      8.8821010163462422E+00
52      1.6598538397889143E+01
53      3.1079527707765703E+01
54      5.8358006063602978E+01
55      1.1000966104096939E+02
56      2.0848750508658674E+02
57      3.9793692474094433E+02
58      7.6660358942891128E+02
59      1.4944323346681333E+03
60      2.9570621576747585E+03
61      5.9602824302208819E+03
62      1.2287273922270530E+04
63      2.6025474503929181E+04
64      5.6920167949008617E+04
65      1.2924146561822579E+05
66      3.0640047899595002E+05
67      7.6297421342751326E+05
68      2.0076698631188064E+06
69      5.6164425941082174E+06
70      1.6802885560418982E+07
71      5.4065392869313173E+07
72  30
73      3.2777474514072306E-02
74      6.1052208024089318E-02
75      1.1371779544172864E-01
76      2.1181678224305334E-01
77      3.9455421330447510E-01
78      7.3499993917294237E-01
79      1.3694267119018748E+00
80      2.5522453323068639E+00
81      4.7592004160127050E+00
82      8.8821010163462422E+00
83      1.6598538397889143E+01
84      3.1079527707765703E+01
85      5.8358006063602978E+01
86      1.1000966104096939E+02
87      2.0848750508658674E+02
88      3.9793692474094433E+02
89      7.6660358942891128E+02
90      1.4944323346681333E+03
91      2.9570621576747585E+03
92      5.9602824302208819E+03
93      1.2287273922270530E+04
94      2.6025474503929181E+04
95      5.6920167949008617E+04
96      1.2924146561822579E+05
97      3.0640047899595002E+05
98      7.6297421342751326E+05
99      2.0076698631188064E+06
100     5.6164425941082174E+06
101     1.6802885560418982E+07
102     5.4065392869313173E+07
103 25
104     3.2777474514072306E-02
105     6.1052208024089318E-02
106     1.1371779544172864E-01
107     2.1181678224305334E-01
108     3.9455421330447510E-01
109     7.3499993917294237E-01
110     1.3694267119018748E+00
111     2.5522453323068639E+00
112     4.7592004160127050E+00
113     8.8821010163462422E+00
114     1.6598538397889143E+01
115     3.1079527707765703E+01
116     5.8358006063602978E+01
117     1.1000966104096939E+02
118     2.0848750508658674E+02
119     3.9793692474094433E+02
120     7.6660358942891128E+02
121     1.4944323346681333E+03
122     2.9570621576747585E+03
123     5.9602824302208819E+03
124     1.2287273922270530E+04
125     2.6025474503929181E+04
126     5.6920167949008617E+04
127     1.2924146561822579E+05
128     3.0640047899595002E+05
129 25
130     3.2777474514072306E-02
131     6.1052208024089318E-02
132     1.1371779544172864E-01
```

```
133    2.1181678224305334E-01
134    3.9455421330447510E-01
135    7.3499993917294237E-01
136    1.3694267119018748E+00
137    2.5522453323068639E+00
138    4.7592004160127050E+00
139    8.8821010163462422E+00
140    1.6598538397889143E+01
141    3.1079527707765703E+01
142    5.8358006063602978E+01
143    1.1000966104096939E+02
144    2.0848750508658674E+02
145    3.9793692474094433E+02
146    7.6660358942891128E+02
147    1.4944323346681333E+03
148    2.9570621576747585E+03
149    5.9602824302208819E+03
150    1.2287273922270530E+04
151    2.6025474503929181E+04
152    5.6920167949008617E+04
153    1.2924146561822579E+05
154    3.0640047899595002E+05
155  20
156    3.2777474514072306E-02
157    6.1052208024089318E-02
158    1.1371779544172864E-01
159    2.1181678224305334E-01
160    3.9455421330447510E-01
161    7.3499993917294237E-01
162    1.3694267119018748E+00
163    2.5522453323068639E+00
164    4.7592004160127050E+00
165    8.8821010163462422E+00
166    1.6598538397889143E+01
167    3.1079527707765703E+01
168    5.8358006063602978E+01
169    1.1000966104096939E+02
170    2.0848750508658674E+02
171    3.9793692474094433E+02
172    7.6660358942891128E+02
173    1.4944323346681333E+03
174    2.9570621576747585E+03
175    5.9602824302208819E+03
176  20
177    3.2777474514072306E-02
178    6.1052208024089318E-02
179    1.1371779544172864E-01
180    2.1181678224305334E-01
181    3.9455421330447510E-01
182    7.3499993917294237E-01
183    1.3694267119018748E+00
184    2.5522453323068639E+00
185    4.7592004160127050E+00
186    8.8821010163462422E+00
187    1.6598538397889143E+01
188    3.1079527707765703E+01
189    5.8358006063602978E+01
190    1.1000966104096939E+02
191    2.0848750508658674E+02
192    3.9793692474094433E+02
193    7.6660358942891128E+02
194    1.4944323346681333E+03
195    2.9570621576747585E+03
196    5.9602824302208819E+03
197  $econfig nclose=7 5 5 3 3 1 1 nopen=0 0 0 1 1 1 1 nconf=6 $end
198  0.0 0.0 0.0 1.0 0.0 0.0 2.0
199  0.0 0.0 0.0 1.0 0.0 1.0 1.0
200  0.0 0.0 0.0 1.0 0.0 2.0 0.0
201  0.0 0.0 0.0 0.0 1.0 0.0 2.0
202  0.0 0.0 0.0 0.0 1.0 1.0 1.0
203  0.0 0.0 0.0 0.0 1.0 2.0 0.0
204  $scf maxitr=100 ixtrp=1 dfctr=0.5 $end
```

Listing A.5: *Input file for calculating the single point energy of Pa using a predefined basis set.*

```
1   &CONTRL
2   JOBTYPE = energy,
3   C =    137.0359997600000
4   /
5   &NUC
6   ZNUC =    91.00000000000000      ,
7   NUCMDL =             3,
8   RNUC =    0.000000000000000      ,
9   ALPHA =    129305395.1200000
10  /
11  &BAS
12  NSYM =             7,
13  BASTYPE = rdin ,
```

```
14 | NGROUP =             1
15 | /
16 | &ECONFIG
17 | NCLOSE =             7,
18 |                      5,
19 |                      5,
20 |                      3,
21 |                      3,
22 |                      1,
23 |                      1,
24 | NOPEN =              0,
25 |                      0,
26 |                      0,
27 |                      1,
28 |                      1,
29 |                      1,
30 |                      1,
31 | FREEEL =             1,
32 | AUTOGEN =  F,
33 | NCONF =              6
34 | /
35 | nconf = 6
36 | 1  0.00  0.00  0.00  1.00  0.00  0.00  2.00
37 | 2  0.00  0.00  0.00  1.00  0.00  1.00  1.00
38 | 3  0.00  0.00  0.00  1.00  0.00  2.00  0.00
39 | 4  0.00  0.00  0.00  0.00  1.00  0.00  2.00
40 | 5  0.00  0.00  0.00  0.00  1.00  1.00  1.00
41 | 6  0.00  0.00  0.00  0.00  1.00  2.00  0.00
42 | &SCF
43 | MAXITR =           100,
44 | IXTRP =              1,
45 | DFCTR =   0.5000000000000000        ,
46 | THDLL =   1.0000000000000001E-005,
47 | THDSL =   9.9999999999999995E-008,
48 | THDSS =   1.0000000000000001E-009
49 | /
50 |        31560
51 |      166068720
52 |         82680
53 |         82680
54 |      221424960
55 | Calculating one electron integrals
56 | Calculating two electron integrals
57 | Calculating guess
58 | Starting SCF
59 | SCF converged at iteration 41 Final energy =     -27274.37754610942
60 |
61 |
62 |                        1S+           2S+           3S+           4S+           5S+           6S+           7S+
63 |               -4.1678552E+03 -7.8202739E+02 -1.9970482E+02 -5.2155429E+01 -1.1997194E+01 -2.0494257E+00 -1.9849676E-01
64 |
65 |
66 | 3.2777475E-02  6.3766410E-07  5.7309076E-06 -6.1014939E-05  1.3503449E-04 -8.9549947E-05  2.0217706E-04  7.7763953E-01
67 | 6.1052208E-02 -2.7212503E-06 -2.4451140E-05  2.5997072E-04 -5.7257873E-04  3.6084676E-04 -1.8505829E-03  1.9084131E-01
68 | 1.1371780E-01  6.4661788E-06  5.8087512E-05 -6.1594002E-04  1.3479564E-03 -7.5938581E-04 -1.0442761E-02  4.3509449E-01
69 | 2.1181678E-01 -1.1776869E-05 -1.0570257E-04  1.1185370E-03 -2.4045368E-03  1.2148510E-03 -2.1151300E-01 -4.1956006E-01
70 | 3.9455421E-01  1.8769658E-05  1.6866127E-04 -1.7636196E-03  3.7907496E-03 -1.2618495E-04 -7.3718664E-01 -3.5322441E-01
71 | 7.3499994E-01 -2.7775219E-05 -2.4890013E-04  2.6020802E-03 -5.2448412E-03  2.1094051E-02 -6.2243418E-01 -3.3766148E-01
72 | 1.3694267E+00  3.9581146E-05  3.5469575E-04 -3.6690519E-03  6.5256712E-03  3.6524111E-01  4.7977497E-01  2.0741439E-01
73 | 2.5522453E+00 -5.5531012E-05 -5.0644472E-04  4.8195743E-03 -1.3768212E-02  1.0628665E+00  1.0136423E+00  3.4555769E-01
74 | 4.7592004E+00  7.5042533E-05  6.7454424E-04 -6.6562007E-03 -1.5840398E-01  1.2336517E-01  1.2190063E-01  4.7160748E-02
75 | 8.8821010E+00 -9.9650704E-05 -8.8184744E-04  1.1265810E-02 -9.1814825E-01 -1.2574109E+00 -7.1892239E-01 -2.3332330E-01
76 | 1.6598538E+01  1.3971887E-04  1.4607049E-03  7.8898658E-02 -5.7101352E-01 -5.1418558E-01 -2.6643838E-01 -8.2270513E-02
77 | 3.1079528E+01 -1.7602025E-04 -2.0695834E-03  6.1053018E-01  8.6852650E-01  5.6179744E-01  2.6866146E-01  8.2644473E-02
78 | 5.8358006E+01  1.6963083E-04 -2.3283094E-02  7.0220007E-01  7.7741233E-01  4.7333116E-01  2.1834120E-01  6.7834903E-02
79 | 1.1000966E+02 -2.3821652E-04 -2.5386103E-01 -2.1304165E-01 -1.4183476E-01 -7.6052348E-02 -3.3602641E-02 -1.0499887E-02
80 | 2.0848751E+02  8.5338160E-04 -5.1752274E-01 -5.4684268E-01 -3.4531663E-01 -1.7912344E-01 -8.0359590E-02 -2.4606349E-02
81 | 3.9793692E+02  2.2723041E-02 -3.2469759E-01 -2.6916770E-01 -1.6072880E-01 -8.2191235E-02 -3.6464966E-02 -1.1231440E-02
82 | 7.6660359E+02  1.2996824E-01  7.6465413E-03 -8.8975780E-03 -6.2835928E-03 -3.2962445E-03 -1.5389625E-03 -4.5630116E-04
83 | 1.4944323E+03  2.4681387E-01  1.4585997E-01  7.6029536E-02  3.9881367E-02  1.9666704E-02  8.7101348E-03  2.6628490E-03
84 | 2.9570622E+03  2.5643870E-01  1.4221656E-01  7.3229143E-02  3.8524236E-02  1.9030766E-02  8.3940292E-03  2.5765332E-03
85 | 5.9602824E+03  1.9274715E-01  9.7608608E-02  4.9362514E-02  2.5773385E-02  1.2700084E-02  5.6108718E-03  1.7187114E-03
86 | 1.2287274E+04  1.2229619E-01  5.8398368E-02  2.9197147E-02  1.5227645E-02  7.5035816E-03  3.3103362E-03  1.0152861E-03
87 | 2.6025475E+04  7.1426092E-02  3.2800158E-02  1.6284612E-02  8.4675426E-03  4.1682198E-03  1.8402301E-03  5.6389575E-04
88 | 5.6920168E+04  4.0099771E-02  1.7926580E-02  8.8580193E-03  4.6043650E-03  2.2666591E-03  1.0000444E-03  3.0662309E-04
89 | 1.2924147E+05  2.2205546E-02  9.7155189E-03  4.7836464E-03  2.4828387E-03  1.2216568E-03  5.3917832E-04  1.6524649E-04
90 | 3.0640048E+05  1.2181181E-02  5.2395866E-03  2.5724532E-03  1.3347501E-03  6.5674013E-04  2.8975990E-04  8.8829643E-05
91 | 7.6297421E+05  6.6597693E-03  2.8244368E-03  1.3834892E-03  7.1723201E-04  3.5280484E-04  1.5568362E-04  4.7717415E-05
92 | 2.0076699E+06  3.5575042E-03  1.4928906E-03  7.2995848E-04  3.7832779E-04  1.8609143E-04  8.2104714E-05  2.5168451E-05
93 | 5.6164426E+06  1.8566153E-03  7.7288988E-04  3.7740417E-04  1.9551278E-04  9.6154484E-05  4.2426773E-05  1.3004305E-05
94 | 1.6802886E+07  8.8408903E-04  3.6609607E-04  1.7860420E-04  9.2513170E-05  4.5497769E-05  2.0073628E-05  6.1532077E-06
95 | 5.4065393E+07  3.6372280E-04  1.5003603E-04  7.3149434E-05  3.7880674E-05  1.8628161E-05  8.2191072E-06  2.5192723E-06
96 | 1.8809445E+08  3.9084666E-05  1.6088047E-05  7.8407006E-06  4.0605376E-06  1.9968838E-06  8.8097218E-07  2.7005784E-07
97 | 7.1100931E+08 -2.6913120E-06 -1.1074607E-06 -5.3969183E-07 -2.7956824E-07 -1.3750197E-07 -6.0650654E-08 -1.8595851E-08
98 | 2.9330971E+09  4.2284342E-07  1.7399041E-07  8.4788201E-08  4.3925941E-08  2.1605380E-08  9.5292259E-09  2.9219333E-09
99 | 1.3255936E+10 -6.6036810E-08 -2.7172566E-08 -1.3241564E-08 -6.8602231E-09 -3.3743106E-09 -1.4882350E-09 -4.5634558E-10
100| 6.5852665E+10  8.2684455E-09  3.4022696E-09  1.6579716E-09  8.5897268E-10  4.2250091E-10  1.8634247E-10  5.7139514E-11
101|
102|
103| 3.2777475E-02  6.6583018E-09  5.1997131E-09 -9.0280402E-08  1.4001238E-07 -1.1702732E-07  2.2257972E-07  8.8973711E-04
104| 6.1052208E-02 -2.7234862E-08 -3.3234803E-08  4.8543240E-07 -8.3875912E-07  6.1956519E-07 -2.8561455E-06  2.9801286E-04
105| 1.1371780E-01  6.5584587E-08  1.1357878E-07 -1.4921476E-06  2.7503848E-06 -1.7436024E-06 -2.2330833E-05  9.2721339E-04
```

```
106   2.1181678E-01 -1.2687135E-07 -2.9131507E-07  3.5750657E-06 -6.7804012E-06  3.7545795E-06 -6.1508893E-04 -1.2202708E-03
107   3.9455421E-01  2.2292090E-07  6.4801348E-07 -7.5181259E-06  1.4716319E-05 -8.4814951E-07 -2.9266569E-03 -1.4022390E-03
108   7.3499994E-01 -3.7623323E-07 -1.3240078E-06 -1.4888317E-05 -2.7928998E-05  1.1483385E-04 -3.3719703E-03 -1.8292450E-03
109   1.3694267E+00  6.2820434E-07  2.6019932E-06 -2.8314609E-05  4.7559797E-05  2.7010967E-03  3.5481729E-03  1.5339608E-03
110   2.5522453E+00 -1.0548532E-06 -5.1099838E-06  5.0406770E-05 -1.3805043E-04  1.0733461E-02  1.0233838E-02  3.4886688E-03
111   4.7592004E+00  1.7607038E-06  9.3408006E-06 -9.4370031E-05 -2.1881955E-03  1.6938343E-03  1.6747161E-03  6.4816516E-04
112   8.8821010E+00 -2.9467977E-06 -1.6757433E-05  2.1620569E-04 -1.7304956E-02 -2.3680505E-02 -1.3537694E-02 -4.3935092E-03
113   1.6598538E+01  5.1797725E-06  3.8010920E-05  2.0360874E-03 -1.4680955E-02 -1.3203591E-02 -6.8396882E-03 -2.1117539E-03
114   3.1079528E+01 -8.5036007E-06 -7.3720904E-05  2.1595179E-02  3.0642804E-02  1.9816957E-02  9.4758500E-03  2.9149436E-03
115   5.8358006E+01  1.1461839E-05 -1.1455299E-03  3.3856539E-02  3.7345896E-02  2.2720121E-02  1.0478503E-02  3.2554645E-03
116   1.1000966E+02 -2.0387677E-05 -1.7093763E-02 -1.4384488E-02 -9.6242069E-03 -5.1671769E-03 -2.2851648E-03 -7.1377344E-04
117   2.0848751E+02  8.9042294E-05 -4.7607546E-02 -4.9636309E-02 -3.1268197E-02 -1.6208787E-02 -7.2705507E-03 -2.2261834E-03
118   3.9793692E+02  3.1750790E-03 -4.0355126E-02 -3.2905184E-02 -1.9582457E-02 -1.0005800E-02 -4.4379426E-03 -1.3669924E-03
119   7.6660359E+02  2.4905109E-02  2.4571143E-02 -6.4702145E-04 -5.6630997E-04 -3.0559609E-04 -1.4848940E-04 -4.2804830E-05
120   1.4944323E+03  6.4312504E-02  3.5404032E-02  1.8368843E-02  9.6269446E-03  4.7461617E-03  2.1018145E-03  6.4257151E-04
121   2.9570622E+03  8.9937328E-02  4.5939928E-02  2.3407135E-02  1.2284219E-02  6.0646469E-03  2.6744338E-03  8.2093827E-04
122   5.9602824E+03  8.9537818E-02  4.1711583E-02  2.0842689E-02  1.0849360E-02  5.3416257E-03  2.3596508E-03  7.2272164E-04
123   1.2287274E+04  7.3753937E-02  3.2516176E-02  1.6061406E-02  8.3530855E-03  4.1130977E-03  1.8140569E-03  5.5641213E-04
124   2.6025475E+04  5.4420421E-02  2.3238701E-02  1.1407274E-02  5.9138670E-03  2.9087461E-03  1.2840890E-03  3.9342287E-04
125   5.6920168E+04  3.7450532E-02  1.5729658E-02  7.6951575E-03  3.9906895E-03  1.9634095E-03  8.6601592E-04  2.6555365E-04
126   1.2924147E+05  2.4513497E-02  1.0196669E-02  4.9794708E-03  2.5787438E-03  1.2680510E-03  5.5964584E-04  1.7149460E-04
127   3.0640048E+05  1.5383160E-02  6.3679065E-03  3.1066330E-03  1.6096318E-03  7.9171441E-04  3.4923746E-04  1.0707400E-04
128   7.6297421E+05  9.2503621E-03  3.8170961E-03  1.8612148E-03  9.6364657E-04  4.7383792E-04  2.0909878E-04  6.4081524E-05
129   2.0076699E+06  5.2989334E-03  2.1831696E-03  1.0641632E-03  5.5118940E-04  2.7107876E-04  1.1958448E-04  3.6660831E-05
130   5.6164426E+06  2.8243198E-03  1.1621686E-03  5.6638269E-04  2.9323102E-04  1.4418577E-04  6.3623385E-05  1.9499446E-05
131   1.6802886E+07  1.3403351E-03  5.5119574E-04  2.6858785E-04  1.3909881E-04  6.8406847E-05  3.0177984E-05  9.2513068E-06
132   5.4065393E+07  4.2517766E-04  1.7477765E-04  8.5163139E-05  4.4087413E-05  2.1677739E-05  9.5657566E-06  2.9316483E-06
133   1.8809445E+08  9.8714923E-06  4.0808855E-06  1.9896300E-06  1.0343690E-06  5.0954248E-07  2.2422999E-07  6.8918938E-08
134   7.1100931E+08  1.1624993E-06  4.7251887E-07  2.3006947E-07  1.1756714E-07  5.7473477E-08  2.5582624E-08  7.7693109E-09
135   2.9330971E+09 -5.6493630E-07 -2.3038071E-07 -1.1220197E-07 -5.7527294E-08 -2.8164677E-08 -1.2508763E-08 -3.8077421E-09
136   1.3255936E+10  2.0123625E-07  8.2100500E-08  3.9986755E-08  2.0510704E-08  1.0043772E-08  4.4594273E-09  1.3578940E-09
137   6.5852665E+10 -5.6896317E-08 -2.3214553E-08 -1.1306646E-08 -5.8000751E-09 -2.8403099E-09 -1.2610270E-09 -3.8400415E-10
138
139
140                                2P-          3P-          4P-          5P-          6P-
141                         -7.5278343E+02 -1.8645827E+02 -4.6167441E+01 -9.6054318E+00 -1.2842044E+00
142
143
144   3.2777475E-02 -7.1419616E-07  7.8103090E-06  8.0014713E-05  4.1421115E-05 -3.0036559E-04
145   6.1052208E-02  2.7478595E-06 -2.9964405E-05 -3.0698594E-04 -2.0084087E-04 -4.6908852E-03
146   1.1371780E-01 -6.1266438E-06  6.6795018E-05  6.7484733E-04  3.4852567E-04 -2.5326033E-02
147   2.1181678E-01  1.0806498E-05 -1.1735248E-04 -1.1748191E-03 -7.1990125E-04 -1.9859318E-01
148   3.9455421E-01 -1.7050462E-05  1.8367136E-04  1.8136919E-03  2.4591053E-03 -5.0663001E-01
149   7.3499994E-01  2.5467950E-05 -2.7268255E-04 -2.5270870E-03  1.8786243E-02 -4.8412064E-01
150   1.3694267E+00 -3.5833116E-05  4.0248005E-04  3.3854562E-03  2.2862081E-01  5.3893134E-02
151   2.5522453E+00  5.7665574E-05 -4.8877642E-04 -7.5621053E-03  6.5921867E-01  4.7808254E-01
152   4.7592004E+00 -7.1829380E-05  8.1151373E-04 -7.6853322E-02  4.1932271E-01  2.7962063E-01
153   8.8821010E+00  9.0786449E-05 -2.2127713E-03 -4.8837158E-01 -4.1452975E-01 -1.8996887E-01
154   1.6598538E+01 -2.6568400E-04 -3.2082164E-02 -6.1426897E-01 -5.0407795E-01 -2.3595179E-01
155   3.1079528E+01  3.3486473E-04 -2.6807434E-01  3.5412768E-02 -1.9851858E-02 -1.1221864E-02
156   5.8358006E+01  6.0909364E-03 -5.3331952E-01  4.5960941E-01  2.4801552E-01  1.0435215E-01
157   1.1000966E+02  8.1427606E-02 -2.9005669E-01  2.4006230E-01  1.2924157E-01  5.4795638E-02
158   2.0848751E+02  2.4884351E-01  6.9516352E-02 -2.1149330E-02 -8.3830283E-03 -3.3844887E-03
159   3.9793692E+02  3.2142412E-01  1.8355468E-01 -9.7197522E-02 -4.6461116E-02 -1.9084088E-02
160   7.6660359E+02  2.4991232E-01  1.4391088E-01 -7.6545227E-02 -3.6856495E-02 -1.5215454E-02
161   1.4944323E+03  1.4966669E-01  8.3245213E-02 -4.3851270E-02 -2.0984900E-02 -8.6331107E-03
162   2.9570622E+03  7.9577568E-02  4.3190468E-02 -2.2617427E-02 -1.0837131E-02 -4.4669823E-03
163   5.9602824E+03  4.1165686E-02  2.1906788E-02 -1.1412507E-02 -5.4487875E-03 -2.2411963E-03
164   1.2287274E+04  2.1764771E-02  1.1463227E-02 -5.9575944E-03 -2.8476002E-03 -1.1727872E-03
165   2.6025475E+04  1.1962129E-02  6.2485980E-03 -3.2404186E-03 -1.5460261E-03 -6.3599606E-04
166   5.6920168E+04  6.8009560E-03  3.5390295E-03 -1.8337079E-03 -8.7543173E-04 -3.6036597E-04
167   1.2924147E+05  3.9458576E-03  2.0472703E-03 -1.0599356E-03 -5.0564610E-04 -2.0804375E-04
168   3.0640048E+05  2.2955149E-03  1.1896197E-03 -6.1574652E-04 -2.9382355E-04 -1.2092274E-04
169   7.6297421E+05  1.3185566E-03  6.8270858E-04 -3.5328364E-04 -1.6853589E-04 -6.9348191E-05
170   2.0076699E+06  7.3321294E-04  3.7951322E-04 -1.9637484E-04 -9.3691920E-05 -3.8555676E-05
171   5.6164426E+06  3.8556276E-04  1.9951148E-04 -1.0322691E-04 -4.9245293E-05 -2.0263741E-05
172   1.6802886E+07  1.7968610E-04  9.2969270E-05 -4.8101095E-05 -2.2948263E-05 -9.4433293E-06
173   5.4065393E+07  5.8601596E-05  3.0318645E-05 -1.5686187E-05 -7.4832266E-06 -3.0792590E-06
174
175
176   3.2777475E-02 -9.3316637E-06  7.0000665E-08  1.7578676E-06  2.5432750E-07 -2.2427559E-07
177   6.1052208E-02  4.5611594E-06 -1.0155629E-07 -1.4049180E-06 -4.7480480E-07 -9.5509282E-06
178   1.1371780E-01 -1.5158182E-05  3.0331219E-07  4.4901492E-06  1.2252677E-06 -6.9349211E-05
179   2.1181678E-01 -4.2451680E-06 -4.7050070E-07 -2.5070434E-06 -7.4556346E-04
180   3.9455421E-01 -2.2395358E-05  1.1587154E-06  1.3135576E-05  1.2921192E-05 -2.5959546E-03
181   7.3499994E-01 -2.4101698E-05 -1.9000805E-06 -1.3264295E-05  1.3219801E-04 -3.3854786E-03
182   1.3694267E+00 -4.7031223E-05  4.3178993E-06  4.0361802E-05  2.1838455E-03  5.1556783E-04
183   2.5522453E+00 -6.8311062E-05 -6.2725455E-06 -8.6157072E-05  8.5954352E-03  6.2321848E-03
184   4.7592004E+00 -1.1523798E-04  1.5580022E-05 -1.3497218E-03  7.4633643E-03  4.9748472E-03
185   8.8821010E+00 -1.7775408E-04 -5.3566907E-05 -1.1851927E-02 -1.0070720E-02 -4.6165537E-03
186   1.6598538E+01 -2.9859355E-04 -1.0682392E-03 -2.0366327E-02 -1.6714489E-02 -7.8238944E-03
187   3.1079528E+01 -4.4960812E-04 -1.2235754E-02  1.7031350E-03 -8.3772517E-04 -4.8139265E-04
188   5.8358006E+01 -3.6322307E-04 -3.3281971E-02  2.8614634E-02  1.5437728E-02  6.4952597E-03
189   1.1000966E+02  5.8633180E-03 -2.4755158E-02  2.0357568E-02  1.0947083E-02  4.6403333E-03
190   2.0848751E+02  2.7663532E-02  7.9050943E-03 -2.4590451E-03 -9.8714652E-04 -3.9984555E-04
191   3.9793692E+02  4.9278632E-02  2.8466864E-02 -1.5119849E-02 -7.2324632E-03 -2.9709995E-03
192   7.6660359E+02  5.1169865E-02  2.9666664E-02 -1.5802650E-02 -7.6127992E-03 -3.1434241E-03
193   1.4944323E+03  3.9687448E-02  2.2187704E-02 -1.1701297E-02 -5.5999814E-03 -2.3035144E-03
194   2.9570622E+03  2.6194603E-02  1.4300717E-02 -7.4988679E-03 -3.5955499E-03 -1.4826020E-03
195   5.9602824E+03  1.5931698E-02  8.5328085E-03 -4.4517913E-03 -2.1251481E-03 -8.7380651E-04
196   1.2287274E+04  9.3288421E-03  4.9546810E-03 -2.5802520E-03 -1.2347919E-03 -5.0889786E-04
197   2.6025475E+04  5.3842381E-03  2.8338848E-03 -1.4721886E-03 -7.0212833E-04 -2.8866928E-04
```

112

```
198    5.6920168E+04   3.0926864E-03   1.6225415E-03  -8.4241915E-04  -4.0273993E-04  -1.6592330E-04
199    1.2924147E+05   1.7734094E-03   9.2533153E-04  -4.7969863E-04  -2.2872350E-04  -9.4046964E-05
200    3.0640048E+05   1.0104040E-03   5.2642764E-04  -2.7285039E-04  -1.3034549E-04  -5.3681772E-05
201    7.6297421E+05   5.6708693E-04   2.9446980E-04  -1.5247942E-04  -7.2704287E-05  -2.9900411E-05
202    2.0076699E+06   3.0829826E-04   1.5999996E-04  -8.2847907E-05  -3.9556961E-05  -1.6286587E-05
203    5.6164426E+06   1.5771287E-04   8.1695277E-05  -4.2278166E-05  -2.0160475E-05  -8.2925202E-06
204    1.6802886E+07   7.0609820E-05   3.6575817E-05  -1.8929933E-05  -9.0358362E-06  -3.7196921E-06
205    5.4065393E+07   2.1734831E-05   1.1245924E-05  -5.8182964E-06  -2.7744208E-06  -1.1412212E-06
206
207
208                                 2P+             3P+             4P+             5P+             6P+
209                       -6.1939589E+02  -1.5549183E+02  -3.8038164E+01  -7.7221173E+00  -9.5408788E-01
210
211
212    3.2777475E-02  -2.4336490E-06   1.6754306E-05  -5.5892525E-05  -6.0387643E-05   1.2386475E-03
213    6.1052208E-02   9.3576855E-06  -6.4207914E-05   2.1493368E-04   3.0280989E-04   1.1874241E-02
214    1.1371780E-01  -2.0842321E-05   1.4278822E-04  -4.6618926E-04  -4.7530162E-04   7.7427164E-02
215    2.1181678E-01   3.6697396E-05  -2.4977203E-04   8.0604051E-04   9.0129034E-04   3.2545555E-01
216    3.9455421E-01  -5.7637303E-05   3.8792609E-04  -1.2273094E-03  -4.6882426E-03   5.3826085E-01
217    7.3499994E-01   8.5352209E-05  -5.6655569E-04   1.5800164E-03  -5.2087664E-02   3.1320799E-01
218    1.3694267E+00  -1.2053399E-04   8.0999454E-04  -1.6846906E-03  -3.6568845E-01  -2.3816734E-01
219    2.5522453E+00   1.7779718E-04  -9.4218169E-04   1.2903151E-02  -6.9361708E-01  -4.5706220E-01
220    4.7592004E+00  -2.2790063E-04   1.3715452E-03   1.6619670E-01  -1.6575956E-01  -1.1313649E-01
221    8.8821010E+00   2.7867735E-04  -4.9912777E-03   6.1780391E-01   5.6618084E-01   2.5261361E-01
222    1.6598538E+01  -5.3063638E-04  -7.6655861E-02   4.7545603E-01   3.7594386E-01   1.6531536E-01
223    3.1079528E+01   8.1926120E-04  -3.9443725E-01  -2.6325439E-01  -1.2640907E-01  -5.0282867E-02
224    5.8358006E+01   1.9516933E-02  -5.3111228E-01  -4.6566357E-01  -2.5019762E-01  -1.0025868E-01
225    1.1000966E+02   1.5033101E-01  -1.3594093E-01  -1.2662521E-01  -6.8504877E-02  -2.7729878E-02
226    2.0848751E+02   3.2133747E-01   1.7641008E-01   9.0446983E-02   4.2739129E-02   1.6796378E-02
227    3.9793692E+02   3.2072249E-01   2.0124520E-01   1.0945733E-01   5.2312345E-02   2.0490470E-02
228    7.6660359E+02   2.0050509E-01   1.2068503E-01   6.4829782E-02   3.0961770E-02   1.2158730E-02
229    1.4944323E+03   9.6326899E-02   5.5757202E-02   2.9651607E-02   1.4096148E-02   5.5177320E-03
230    2.9570622E+03   3.9654466E-02   2.2415263E-02   1.1842862E-02   5.6316919E-03   2.2095082E-03
231    5.9602824E+03   1.4971482E-02   8.3261254E-03   4.3824962E-03   2.0776849E-03   8.1267403E-04
232    1.2287274E+04   5.3674900E-03   2.9641098E-03   1.5566798E-03   7.3927073E-04   2.9002714E-04
233    2.6025475E+04   1.8661840E-03   1.0233010E-03   5.3671985E-04   2.5416668E-04   9.9365917E-05
234    5.6920168E+04   6.3461305E-04   3.4729890E-04   1.8196989E-04   8.6386092E-05   3.3896608E-05
235    1.2924147E+05   2.1249834E-04   1.1571279E-04   6.0588224E-05   2.8673602E-05   1.1205637E-05
236    3.0640048E+05   6.9752539E-05   3.7968687E-05   1.9866465E-05   9.4297765E-06   3.7008695E-06
237    7.6297421E+05   2.2397871E-05   1.2139640E-05   6.3493838E-06   3.0032946E-06   1.1732181E-06
238    2.0076699E+06   6.9131086E-06   3.7504542E-06   1.9605527E-06   9.3071395E-07   3.6543963E-07
239    5.6164426E+06   2.0594701E-06   1.1127777E-06   5.8164270E-07   2.7491428E-07   1.0730215E-07
240    1.6802886E+07   5.5135401E-07   2.9875011E-07   1.5608727E-07   7.4164034E-08   2.9160511E-08
241    5.4065393E+07   1.4879784E-07   8.0244195E-08   4.1931736E-08   1.9799755E-08   7.7181920E-09
242
243
244    3.2777475E-02  -1.3207878E-09   3.4965780E-08  -7.5446197E-08  -8.3100511E-08   1.8324502E-06
245    6.1052208E-02   1.0946486E-08  -1.6565764E-07   4.0833249E-07   5.8897280E-07   2.3927737E-05
246    1.1371780E-01  -4.0803138E-08   4.7032525E-07  -1.2299178E-06  -1.2621203E-06   2.1304902E-04
247    2.1181678E-01   1.0985682E-07  -1.0728611E-06   2.9362395E-06   3.3058064E-06   1.2220435E-03
248    3.9455421E-01  -2.5276752E-07   2.2024214E-06  -6.1486831E-06  -2.3905280E-05   2.7585314E-03
249    7.3499994E-01   5.3549677E-07  -4.2892502E-06   1.0840453E-05  -3.6457970E-04   2.1905696E-03
250    1.3694267E+00  -1.0650882E-06   8.2210039E-06  -1.5772498E-05  -3.4916695E-03  -2.2739744E-03
251    2.5522453E+00   2.2017354E-06  -1.3001460E-05   1.6782632E-04  -9.0413702E-03  -5.9569861E-03
252    4.7592004E+00  -3.9021382E-06   2.5481600E-05   2.9612675E-03  -2.9460697E-03  -2.0109624E-03
253    8.8821010E+00   6.5774673E-06  -1.2317379E-04   1.5028180E-02   1.3765672E-02   6.1415481E-03
254    1.6598538E+01  -1.7395834E-05  -2.5543939E-03   1.5787682E-02   1.2475252E-02   5.4850895E-03
255    3.1079528E+01   3.7061187E-05  -1.7988429E-02  -1.2006348E-02  -5.7712025E-03  -2.2961420E-03
256    5.8358006E+01   1.2348390E-03  -3.3073785E-02  -2.8932698E-02  -1.5538574E-02  -6.2260449E-03
257    1.1000966E+02   1.3011629E-02  -1.1426030E-02  -1.0640642E-02  -5.7553138E-03  -2.3297340E-03
258    2.0848751E+02   3.8062766E-02   2.0811109E-02   1.0682682E-02   5.0499788E-03   1.9847382E-03
259    3.9793692E+02   5.1851468E-02   3.3230331E-02   1.7496779E-02   8.3583775E-03   3.2735798E-03
260    7.6660359E+02   4.4072294E-02   2.6230738E-02   1.4053605E-02   6.7075174E-03   2.6337808E-03
261    1.4944323E+03   2.8624950E-02   1.6364740E-02   8.6772510E-03   4.1218999E-03   1.6130929E-03
262    2.9570622E+03   1.5770993E-02   8.8005603E-03   4.6349164E-03   2.2024914E-03   8.6408133E-04
263    5.9602824E+03   7.8639115E-03   4.3171518E-03   2.2652863E-03   1.0729240E-03   4.1949616E-04
264    1.2287274E+04   3.6559185E-03   1.9947426E-03   1.0443832E-03   4.9574149E-04   1.9453650E-04
265    2.6025475E+04   1.6113476E-03   8.7355432E-04   4.5694978E-04   2.1613580E-04   8.4425446E-05
266    5.6920168E+04   6.7497388E-04   3.6607705E-04   1.9133996E-04   9.0852090E-05   3.5684824E-05
267    1.2924147E+05   2.6989658E-04   1.4571912E-04   7.6161518E-05   3.5980911E-05   1.4034697E-05
268    3.0640048E+05   1.0215312E-04   5.5349847E-05   2.8914350E-05   1.3745415E-05   5.4085524E-06
269    7.6297421E+05   3.6629771E-05   1.9732887E-05   1.0312021E-05   4.8624567E-06   1.8917339E-06
270    2.0076699E+06   1.2254330E-05   6.6477539E-06   3.4717529E-06   1.6546375E-06   6.5339148E-07
271    5.6164426E+06   3.8316748E-06   2.0582388E-06   1.0757435E-06   5.0527204E-07   1.9549628E-07
272    1.6802886E+07   1.0835857E-06   5.8992873E-07   3.0797781E-07   1.4759391E-07   5.8723790E-08
273    5.4065393E+07   2.3568386E-07   1.2586494E-07   6.5813885E-08   3.0642650E-08   1.1707548E-08
274
275
276                                 3D-             4D-             5D-             6D-
277                       -1.3454927E+02  -2.8439539E+01  -4.0951554E+00  -1.9053474E-01
278
279
280    3.2777475E-02   3.2651074E-07  -2.6571957E-05  -7.1019724E-05   8.9491015E-02
281    6.1052208E-02  -1.1894050E-06   9.1843839E-05   1.7552722E-04   1.8654044E-01
282    1.1371780E-01   2.4625831E-06  -1.9297146E-04  -5.2234218E-04   3.0652101E-01
283    2.1181678E-01  -4.0986129E-06   3.2129889E-04  -2.7417100E-04   3.2033054E-01
284    3.9455421E-01   6.1043128E-06  -5.0589439E-04  -1.4563385E-02   2.5061410E-01
285    7.3499994E-01  -8.9341665E-06   6.0227820E-04  -1.0696676E-01   8.9685454E-02
286    1.3694267E+00   9.7845165E-06  -7.2930002E-04  -3.4928272E-01  -9.3657675E-02
287    2.5522453E+00   6.1678263E-06   1.2297876E-02  -4.9298781E-01  -1.6224526E-01
288    4.7592004E+00  -9.5122511E-05   1.2306050E-01  -2.2981896E-01  -7.5638494E-02
289    8.8821010E+00  -1.7044696E-03   4.0358527E-01   1.9020216E-01   4.6878442E-02
```

```
290    1.6598538E+01 -3.3490722E-02  4.5955502E-01  2.6898485E-01  7.0909914E-02
291    3.1079528E+01 -1.7976659E-01  1.4192470E-01  9.4582711E-02  2.5010988E-02
292    5.8358006E+01 -3.5445132E-01 -1.5547080E-01 -6.1021210E-02 -1.5051975E-02
293    1.1000966E+02 -3.3068223E-01 -1.8988192E-01 -8.3626235E-02 -2.1126171E-02
294    2.0848751E+02 -1.9048277E-01 -1.1074452E-01 -4.8575649E-02 -1.2239840E-02
295    3.9793692E+02 -8.1562455E-02 -4.7076496E-02 -2.0731398E-02 -5.2359922E-03
296    7.6660359E+02 -2.9130071E-02 -1.6474611E-02 -7.1776740E-03 -1.8063717E-03
297    1.4944323E+03 -9.6393649E-03 -5.4487794E-03 -2.3916031E-03 -6.0404305E-04
298    2.9570622E+03 -3.0667014E-03 -1.7080815E-03 -7.4001605E-04 -1.8594553E-04
299    5.9602824E+03 -1.0114887E-03 -5.6807403E-04 -2.4954728E-04 -6.3083197E-05
300    1.2287274E+04 -3.5232458E-04 -1.9484679E-04 -8.4131236E-05 -2.1116516E-05
301    2.6025475E+04 -1.2123834E-04 -6.7931142E-05 -2.9858569E-05 -7.5513692E-06
302    5.6920168E+04 -5.1206264E-05 -2.8281789E-05 -1.2218172E-05 -3.0678055E-06
303    1.2924147E+05 -1.2423186E-05 -6.9580184E-06 -3.0604736E-06 -7.7428947E-07
304    3.0640048E+05 -1.0166505E-05 -5.6313512E-06 -2.4435258E-06 -6.1470251E-07
305
306
307    3.2777475E-02  5.1272654E-08 -7.1337975E-08 -9.8695082E-08  1.5640040E-04
308    6.1052208E-02  9.4368734E-09  1.9756252E-07  4.1408779E-07  4.4494135E-04
309    1.1371780E-01  1.5149148E-07 -6.8163196E-07 -1.6165852E-06  9.9782587E-04
310    2.1181678E-01  1.7093311E-07  1.2680016E-06 -1.1699505E-06  1.4231482E-03
311    3.9455421E-01  5.8354045E-07 -3.3044203E-06 -8.8017777E-05  1.5196087E-03
312    7.3499994E-01  9.5286576E-07  4.2711904E-06 -8.8497625E-04  7.4213549E-04
313    1.3694267E+00  2.5613533E-06 -9.4734702E-06 -3.9447915E-03 -1.0579134E-03
314    2.5522453E+00  5.1356596E-06  1.8658384E-04 -7.6009953E-03 -2.5015174E-03
315    4.7592004E+00  9.4183882E-06  2.5869202E-03 -4.8349100E-03 -1.5912555E-03
316    8.8821010E+00 -2.4846844E-05  1.1600744E-02  5.4711831E-03  1.3487348E-03
317    1.6598538E+01 -1.2668094E-03  1.8043647E-02  1.0560150E-02  2.7838624E-03
318    3.1079528E+01 -9.5791864E-03  7.5931119E-03  5.0588871E-03  1.3376943E-03
319    5.8358006E+01 -2.5906001E-02 -1.1405232E-02 -4.4834319E-03 -1.1062167E-03
320    1.1000966E+02 -3.2939316E-02 -1.8940614E-02 -8.3445574E-03 -2.1081864E-03
321    2.0848751E+02 -2.5687730E-02 -1.4944831E-02 -6.5558307E-03 -1.6518885E-03
322    3.9793692E+02 -1.4730133E-02 -8.5051338E-03 -3.7462797E-03 -9.4625040E-04
323    7.6660359E+02 -6.8893687E-03 -3.8966271E-03 -1.6970918E-03 -4.2702250E-04
324    1.4944323E+03 -2.8874401E-03 -1.6343513E-03 -7.1820842E-04 -1.8148297E-04
325    2.9570622E+03 -1.0971088E-03 -6.1041945E-04 -2.6378718E-04 -6.6204885E-05
326    5.9602824E+03 -4.1131546E-04 -2.3233303E-04 -1.0267240E-04 -2.6017839E-05
327    1.2287274E+04 -1.4526502E-04 -7.9726162E-05 -3.4004690E-05 -8.4885749E-06
328    2.6025475E+04 -5.3097540E-05 -3.0262169E-05 -1.3565060E-05 -3.4583031E-06
329    5.6920168E+04 -1.8273467E-05 -9.8440828E-06 -4.1025343E-06 -1.0137506E-06
330    1.2924147E+05 -6.0357247E-06 -3.5020021E-06 -1.6064965E-06 -4.1341700E-07
331    3.0640048E+05 -2.7989375E-06 -1.5096828E-06 -6.3118668E-07 -1.5620812E-07
332
333
334                        3D+            4D+            5D+            6D+
335            -1.2827939E+02 -2.6928655E+01 -3.8101969E+00 -1.8169381E-01
336
337
338    3.2777475E-02 -4.4802587E-07  2.4275314E-05 -7.5317133E-05 -1.0707794E-01
339    6.1052208E-02  1.6224646E-06 -8.3565772E-05  1.6617400E-04 -1.9937753E-01
340    1.1371780E-01 -3.3669023E-06  1.7587412E-04 -5.8806415E-04 -3.1991976E-01
341    2.1181678E-01  5.6351388E-06 -2.9003014E-04 -7.3087509E-04 -3.1570572E-01
342    3.9455421E-01 -8.3286912E-06  4.6510025E-04 -1.8676569E-02 -2.3125690E-01
343    7.3499994E-01  1.1432832E-05 -5.5581886E-04 -1.2503947E-01 -6.3047095E-02
344    1.3694267E+00 -1.5216574E-05  4.6523033E-04 -3.7416157E-01  1.0510482E-01
345    2.5522453E+00 -6.0920422E-06 -1.5488351E-02 -4.8552690E-01  1.5427095E-01
346    4.7592004E+00  1.3841937E-04 -1.4245950E-01 -1.8761814E-01  6.0618295E-02
347    8.8821010E+00  2.2622365E-03 -4.2633114E-01  2.1406810E-01 -5.2331763E-02
348    1.6598538E+01  4.0492708E-02 -4.4457077E-01  2.5889309E-01 -6.6387969E-02
349    3.1079528E+01  1.9887657E-01 -1.0644157E-01  7.5275550E-02 -1.9454885E-02
350    5.8358006E+01  3.6589928E-01  1.7260955E-01 -6.9603820E-02  1.6825684E-02
351    1.1000966E+02  3.2180831E-01  1.8693843E-01 -8.1980897E-02  2.0168569E-02
352    2.0848751E+02  1.7529852E-01  1.0234078E-01 -4.4647524E-02  1.0953251E-02
353    3.9793692E+02  7.0461699E-02  4.0733158E-02 -1.7825555E-02  4.3835269E-03
354    7.6660359E+02  2.2927985E-02  1.2979953E-02 -5.6181141E-03  1.3760728E-03
355    1.4944323E+03  6.6296509E-03  3.7587050E-03 -1.6416421E-03  4.0398079E-04
356    2.9570622E+03  1.7162680E-03  9.5520955E-04 -4.0992766E-04  1.0009302E-04
357    5.9602824E+03  4.3584798E-04  2.4710517E-04 -1.0871783E-04  2.6857714E-05
358    1.2287274E+04  1.0119312E-04  5.5350430E-05 -2.3344938E-05  5.6550027E-06
359    2.6025475E+04  2.4512440E-05  1.4120973E-05 -6.3442033E-06  1.5822968E-06
360    5.6920168E+04  5.1949414E-06  2.7463397E-06 -1.1095423E-06  2.6310949E-07
361    1.2924147E+05  1.1521615E-06  6.8995617E-07 -3.2368921E-07  8.2241922E-08
362    3.0640048E+05  3.0400933E-07  1.5808078E-07 -6.2498362E-08  1.4655020E-08
363
364
365    3.2777475E-02 -2.5112890E-09  4.1934348E-08 -1.3098950E-07 -1.8714305E-04
366    6.1052208E-02  9.5532465E-09 -1.9775127E-07  3.9426617E-07 -4.7556643E-04
367    1.1371780E-01 -2.2638410E-08  5.6940672E-07 -1.9101004E-06 -1.0414533E-03
368    2.1181678E-01  4.5049675E-08 -1.2834203E-06 -3.2547256E-06 -1.4026322E-03
369    3.9455421E-01 -8.2213089E-08  2.8124617E-06 -1.1324607E-04 -1.4022465E-03
370    7.3499994E-01  1.4317164E-07 -4.5883166E-06 -1.0349402E-03 -5.2172390E-04
371    1.3694267E+00 -2.4505475E-07  5.2366233E-06 -4.2269885E-03  1.1873505E-03
372    2.5522453E+00  1.5406900E-08 -2.3897014E-04 -7.4874326E-03  2.3788950E-03
373    4.7592004E+00  2.7528844E-06 -3.0018144E-03 -3.9485289E-03  1.2757083E-03
374    8.8821010E+00  6.5489371E-05 -1.2268290E-02  6.1592757E-03 -1.5057738E-03
375    1.6598538E+01  1.5963932E-03 -1.7471504E-02  1.0170578E-02 -2.6079155E-03
376    3.1079528E+01  1.0724050E-02 -5.6943677E-03  4.0295907E-03 -1.0414543E-03
377    5.8358006E+01  2.6975934E-02  1.2723409E-02 -5.1328540E-03  1.2409059E-03
378    1.1000966E+02  3.2419769E-02  1.8801788E-02 -8.2428979E-03  2.0278041E-03
379    2.0848751E+02  2.4114276E-02  1.4046415E-02 -6.1248998E-03  1.5024841E-03
380    3.9793692E+02  1.3197018E-02  7.6097968E-03 -3.3285423E-03  8.1849208E-04
381    7.6660359E+02  5.8221774E-03  3.2862557E-03 -1.4212514E-03  3.4805303E-04
```

```
382   1.4944323E+03  2.2707250E-03  1.2839997E-03 -5.6067292E-04  1.3798793E-04
383   2.9570622E+03  7.8370384E-04  4.3440158E-04 -1.8606588E-04  4.5400882E-05
384   5.9602824E+03  2.6332078E-04  1.4918807E-04 -6.5774960E-05  1.6268067E-05
385   1.2287274E+04  7.8488463E-05  4.2535433E-05 -1.7781357E-05  4.2895664E-06
386   2.6025475E+04  2.4632134E-05  1.4332594E-05 -6.5269824E-06  1.6378398E-06
387   5.6920168E+04  6.0733526E-06  3.0796907E-06 -1.1747621E-06  2.7012610E-07
388   1.2924147E+05  1.8431209E-06  1.1573922E-06 -5.7003982E-07  1.4770355E-07
389   3.0640048E+05  4.3591834E-07  2.0024127E-07 -6.4421032E-08  1.3266370E-08


                             4F-            5F-
                        -1.4283216E+01 -2.9142420E-01


396   3.2777475E-02 -1.0725471E-05  1.8335167E-03
397   6.1052208E-02  3.7371756E-05  1.4299458E-02
398   1.1371780E-01 -6.5117205E-05  5.1474113E-02
399   2.1181678E-01  1.3410944E-04  1.1826153E-01
400   3.9455421E-01 -1.5304599E-04  2.0834794E-01
401   7.3499994E-01  3.9034287E-04  2.8704100E-01
402   1.3694267E+00  1.8711942E-03  3.1098500E-01
403   2.5522453E+00  2.9485902E-02  2.3916164E-01
404   4.7592004E+00  1.3401088E-01  9.8299863E-02
405   8.8821010E+00  2.9018994E-01 -3.8916120E-02
406   1.6598538E+01  3.4836741E-01 -9.0866778E-02
407   3.1079528E+01  2.6972459E-01 -7.6747815E-02
408   5.8358006E+01  1.4163449E-01 -4.1716302E-02
409   1.1000966E+02  5.3462215E-02 -1.5589750E-02
410   2.0848751E+02  1.6463172E-02 -4.8330990E-03
411   3.9793692E+02  4.0383670E-03 -1.1670891E-03
412   7.6660359E+02  9.6404242E-04 -2.8455384E-04
413   1.4944323E+03  2.0990334E-04 -5.9438911E-05
414   2.9570622E+03  4.2739189E-05 -1.2950119E-05
415   5.9602824E+03  1.5565905E-05 -4.3943503E-06


418   3.2777475E-02 -2.0783188E-08  3.6300411E-06
419   6.1052208E-02  9.3085995E-08  3.8674853E-05
420   1.1371780E-01 -2.3659580E-07  1.8998293E-04
421   2.1181678E-01  6.3262257E-07  5.9574736E-04
422   3.9455421E-01 -1.0880802E-06  1.4323838E-03
423   7.3499994E-01  3.4104055E-06  2.6933700E-03
424   1.3694267E+00  2.3452890E-05  3.9826033E-03
425   2.5522453E+00  5.1386501E-04  4.1805895E-03
426   4.7592004E+00  3.1956728E-03  2.3449523E-03
427   8.8821010E+00  9.4515168E-03 -1.2688828E-03
428   1.6598538E+01  1.5493324E-02 -4.0424851E-03
429   3.1079528E+01  1.6367583E-02 -4.6582595E-03
430   5.8358006E+01  1.1712012E-02 -3.4499746E-03
431   1.1000966E+02  6.0012126E-03 -1.7499308E-03
432   2.0848751E+02  2.4902691E-03 -7.3123932E-04
433   3.9793692E+02  8.0917742E-04 -2.3369431E-04
434   7.6660359E+02  2.5236369E-04 -7.4645254E-05
435   1.4944323E+03  6.5321088E-05 -1.8336818E-05
436   2.9570622E+03  1.7865583E-05 -5.4973542E-06
437   5.9602824E+03  5.2672434E-06 -1.4302192E-06


                             4F+            5F+
                        -1.3898700E+01 -2.6933365E-01


444   3.2777475E-02  1.1675980E-05 -2.6792931E-03
445   6.1052208E-02 -4.1377000E-05 -1.7785577E-02
446   1.1371780E-01  7.0857124E-05 -5.8567071E-02
447   2.1181678E-01 -1.4708650E-04 -1.2737380E-01
448   3.9455421E-01  1.6844262E-04 -2.1546329E-01
449   7.3499994E-01 -4.2699706E-04 -2.8794124E-01
450   1.3694267E+00 -2.1767351E-03 -3.0639932E-01
451   2.5522453E+00 -3.2039098E-02 -2.3140727E-01
452   4.7592004E+00 -1.4005668E-01 -8.9645492E-02
453   8.8821010E+00 -2.9608946E-01  4.2772328E-02
454   1.6598538E+01 -3.4809129E-01  9.0168542E-02
455   3.1079528E+01 -2.6411894E-01  7.4324210E-02
456   5.8358006E+01 -1.3656877E-01  3.9659289E-02
457   1.1000966E+02 -5.0325596E-02  1.4467214E-02
458   2.0848751E+02 -1.4919582E-02  4.3176435E-03
459   3.9793692E+02 -3.4681534E-03  9.8742386E-04
460   7.6660359E+02 -7.6780228E-04  2.2383383E-04
461   1.4944323E+03 -1.3866054E-04  3.8326704E-05
462   2.9570622E+03 -2.6734026E-05  8.1305216E-06
463   5.9602824E+03 -5.2648888E-06  1.3970692E-06


466   3.2777475E-02  2.4030104E-08 -5.3099163E-06
467   6.1052208E-02 -1.1468771E-07 -4.8102747E-05
468   1.1371780E-01  2.6715090E-07 -2.1618653E-04
469   2.1181678E-01 -7.5061919E-07 -6.4167480E-04
470   3.9455421E-01  1.1736847E-06 -1.4814308E-03
471   7.3499994E-01 -4.0316229E-06 -2.7020456E-03
472   1.3694267E+00 -2.7854269E-05 -3.9245322E-03
473   2.5522453E+00 -5.6048759E-04 -4.0459078E-03
```

```
474    4.7592004E+00 -3.3449889E-03 -2.1394404E-03
475    8.8821010E+00 -9.6582637E-03  1.3958154E-03
476    1.6598538E+01 -1.5510672E-02  4.0174278E-03
477    3.1079528E+01 -1.6082860E-02  4.5249404E-03
478    5.8358006E+01 -1.1361507E-02  3.2985776E-03
479    1.1000966E+02 -5.7154982E-03  1.6425083E-03
480    2.0848751E+02 -2.3118320E-03  6.6885753E-04
481    3.9793692E+02 -7.2978324E-04  2.0764163E-04
482    7.6660359E+02 -2.1946679E-04  6.4005928E-05
483    1.4944323E+03 -5.2827097E-05  1.4546466E-05
484    2.9570622E+03 -1.4013535E-05  4.2861177E-06
485    5.9602824E+03 -3.3747959E-06  8.7670867E-07
486
487
488  Calculated energy =   -2.7274377546109416E+04
489  Literature energy =   -2.7274378906250000E+04
490  Difference (cal - lit) = 1.36E-03
491  Relative error (abs((cal - lit)/lit)) = 4.99E-08
492  Basis set for original program
493 35 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
494    3.2777474514072306E-02 1.0d0 1.0d0
495    6.1052208024089318E-02 1.0d0 1.0d0
496    1.1371779544172864E-01 1.0d0 1.0d0
497    2.1181678224305334E-01 1.0d0 1.0d0
498    3.9455421330447510E-01 1.0d0 1.0d0
499    7.3499993917294237E-01 1.0d0 1.0d0
500    1.3694267119018748E+00 1.0d0 1.0d0
501    2.5522453323068639E+00 1.0d0 1.0d0
502    4.7592004160127050E+00 1.0d0 1.0d0
503    8.8821010163462422E+00 1.0d0 1.0d0
504    1.6598538397889143E+01 1.0d0 1.0d0
505    3.1079527707765703E+01 1.0d0 1.0d0
506    5.8358006063602978E+01 1.0d0 1.0d0
507    1.1000966104096939E+02 1.0d0 1.0d0
508    2.0848750508658674E+02 1.0d0 1.0d0
509    3.9793692474094433E+02 1.0d0 1.0d0
510    7.6660358942891128E+02 1.0d0 1.0d0
511    1.4944323346681333E+03 1.0d0 1.0d0
512    2.9570621576747585E+03 1.0d0 1.0d0
513    5.9602824302208819E+03 1.0d0 1.0d0
514    1.2287273922270530E+04 1.0d0 1.0d0
515    2.6025474503929181E+04 1.0d0 1.0d0
516    5.6920167949008617E+04 1.0d0 1.0d0
517    1.2924146561822579E+05 1.0d0 1.0d0
518    3.0640047899595002E+05 1.0d0 1.0d0
519    7.6297421342751326E+05 1.0d0 1.0d0
520    2.0076698631188064E+06 1.0d0 1.0d0
521    5.6164425941082174E+06 1.0d0 1.0d0
522    1.6802885560418982E+07 1.0d0 1.0d0
523    5.4065392869313173E+07 1.0d0 1.0d0
524    1.8809445122304142E+08 1.0d0 1.0d0
525    7.1100930633447683E+08 1.0d0 1.0d0
526    2.9330970963636436E+09 1.0d0 1.0d0
527    1.3255935848588299E+10 1.0d0 1.0d0
528    6.5852664686153687E+10 1.0d0 1.0d0
529 30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
530    3.2777474514072306E-02 1.0d0 1.0d0
531    6.1052208024089318E-02 1.0d0 1.0d0
532    1.1371779544172864E-01 1.0d0 1.0d0
533    2.1181678224305334E-01 1.0d0 1.0d0
534    3.9455421330447510E-01 1.0d0 1.0d0
535    7.3499993917294237E-01 1.0d0 1.0d0
536    1.3694267119018748E+00 1.0d0 1.0d0
537    2.5522453323068639E+00 1.0d0 1.0d0
538    4.7592004160127050E+00 1.0d0 1.0d0
539    8.8821010163462422E+00 1.0d0 1.0d0
540    1.6598538397889143E+01 1.0d0 1.0d0
541    3.1079527707765703E+01 1.0d0 1.0d0
542    5.8358006063602978E+01 1.0d0 1.0d0
543    1.1000966104096939E+02 1.0d0 1.0d0
544    2.0848750508658674E+02 1.0d0 1.0d0
545    3.9793692474094433E+02 1.0d0 1.0d0
546    7.6660358942891128E+02 1.0d0 1.0d0
547    1.4944323346681333E+03 1.0d0 1.0d0
548    2.9570621576747585E+03 1.0d0 1.0d0
549    5.9602824302208819E+03 1.0d0 1.0d0
550    1.2287273922270530E+04 1.0d0 1.0d0
551    2.6025474503929181E+04 1.0d0 1.0d0
552    5.6920167949008617E+04 1.0d0 1.0d0
553    1.2924146561822579E+05 1.0d0 1.0d0
554    3.0640047899595002E+05 1.0d0 1.0d0
555    7.6297421342751326E+05 1.0d0 1.0d0
556    2.0076698631188064E+06 1.0d0 1.0d0
557    5.6164425941082174E+06 1.0d0 1.0d0
558    1.6802885560418982E+07 1.0d0 1.0d0
559    5.4065392869313173E+07 1.0d0 1.0d0
560 30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
561    3.2777474514072306E-02 1.0d0 1.0d0
562    6.1052208024089318E-02 1.0d0 1.0d0
563    1.1371779544172864E-01 1.0d0 1.0d0
564    2.1181678224305334E-01 1.0d0 1.0d0
565    3.9455421330447510E-01 1.0d0 1.0d0
```

```
566    7.3499993917294237E-01 1.0d0 1.0d0
567    1.3694267119018748E+00 1.0d0 1.0d0
568    2.5522453323068639E+00 1.0d0 1.0d0
569    4.7592004160127050E+00 1.0d0 1.0d0
570    8.8821010163462422E+00 1.0d0 1.0d0
571    1.6598538397889143E+01 1.0d0 1.0d0
572    3.1079527707765703E+01 1.0d0 1.0d0
573    5.8358006063602978E+01 1.0d0 1.0d0
574    1.1000966104096939E+02 1.0d0 1.0d0
575    2.0848750508658674E+02 1.0d0 1.0d0
576    3.9793692474094433E+02 1.0d0 1.0d0
577    7.6660358942891128E+02 1.0d0 1.0d0
578    1.4944323346681333E+03 1.0d0 1.0d0
579    2.9570621576747585E+03 1.0d0 1.0d0
580    5.9602824302208819E+03 1.0d0 1.0d0
581    1.2287273922270530E+04 1.0d0 1.0d0
582    2.6025474503929181E+04 1.0d0 1.0d0
583    5.6920167949008617E+04 1.0d0 1.0d0
584    1.2924146561822579E+05 1.0d0 1.0d0
585    3.0640047899595002E+05 1.0d0 1.0d0
586    7.6297421342751326E+05 1.0d0 1.0d0
587    2.0076698631188064E+06 1.0d0 1.0d0
588    5.6164425941082174E+06 1.0d0 1.0d0
589    1.6802885560418982E+07 1.0d0 1.0d0
590    5.4065392869313173E+07 1.0d0 1.0d0
591  25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
592    3.2777474514072306E-02 1.0d0 1.0d0
593    6.1052208024089318E-02 1.0d0 1.0d0
594    1.1371779544172864E-01 1.0d0 1.0d0
595    2.1181678224305334E-01 1.0d0 1.0d0
596    3.9455421330447510E-01 1.0d0 1.0d0
597    7.3499993917294237E-01 1.0d0 1.0d0
598    1.3694267119018748E+00 1.0d0 1.0d0
599    2.5522453323068639E+00 1.0d0 1.0d0
600    4.7592004160127050E+00 1.0d0 1.0d0
601    8.8821010163462422E+00 1.0d0 1.0d0
602    1.6598538397889143E+01 1.0d0 1.0d0
603    3.1079527707765703E+01 1.0d0 1.0d0
604    5.8358006063602978E+01 1.0d0 1.0d0
605    1.1000966104096939E+02 1.0d0 1.0d0
606    2.0848750508658674E+02 1.0d0 1.0d0
607    3.9793692474094433E+02 1.0d0 1.0d0
608    7.6660358942891128E+02 1.0d0 1.0d0
609    1.4944323346681333E+03 1.0d0 1.0d0
610    2.9570621576747585E+03 1.0d0 1.0d0
611    5.9602824302208819E+03 1.0d0 1.0d0
612    1.2287273922270530E+04 1.0d0 1.0d0
613    2.6025474503929181E+04 1.0d0 1.0d0
614    5.6920167949008617E+04 1.0d0 1.0d0
615    1.2924146561822579E+05 1.0d0 1.0d0
616    3.0640047899595002E+05 1.0d0 1.0d0
617  25 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
618    3.2777474514072306E-02 1.0d0 1.0d0
619    6.1052208024089318E-02 1.0d0 1.0d0
620    1.1371779544172864E-01 1.0d0 1.0d0
621    2.1181678224305334E-01 1.0d0 1.0d0
622    3.9455421330447510E-01 1.0d0 1.0d0
623    7.3499993917294237E-01 1.0d0 1.0d0
624    1.3694267119018748E+00 1.0d0 1.0d0
625    2.5522453323068639E+00 1.0d0 1.0d0
626    4.7592004160127050E+00 1.0d0 1.0d0
627    8.8821010163462422E+00 1.0d0 1.0d0
628    1.6598538397889143E+01 1.0d0 1.0d0
629    3.1079527707765703E+01 1.0d0 1.0d0
630    5.8358006063602978E+01 1.0d0 1.0d0
631    1.1000966104096939E+02 1.0d0 1.0d0
632    2.0848750508658674E+02 1.0d0 1.0d0
633    3.9793692474094433E+02 1.0d0 1.0d0
634    7.6660358942891128E+02 1.0d0 1.0d0
635    1.4944323346681333E+03 1.0d0 1.0d0
636    2.9570621576747585E+03 1.0d0 1.0d0
637    5.9602824302208819E+03 1.0d0 1.0d0
638    1.2287273922270530E+04 1.0d0 1.0d0
639    2.6025474503929181E+04 1.0d0 1.0d0
640    5.6920167949008617E+04 1.0d0 1.0d0
641    1.2924146561822579E+05 1.0d0 1.0d0
642    3.0640047899595002E+05 1.0d0 1.0d0
643  20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
644    3.2777474514072306E-02 1.0d0 1.0d0
645    6.1052208024089318E-02 1.0d0 1.0d0
646    1.1371779544172864E-01 1.0d0 1.0d0
647    2.1181678224305334E-01 1.0d0 1.0d0
648    3.9455421330447510E-01 1.0d0 1.0d0
649    7.3499993917294237E-01 1.0d0 1.0d0
650    1.3694267119018748E+00 1.0d0 1.0d0
651    2.5522453323068639E+00 1.0d0 1.0d0
652    4.7592004160127050E+00 1.0d0 1.0d0
653    8.8821010163462422E+00 1.0d0 1.0d0
654    1.6598538397889143E+01 1.0d0 1.0d0
655    3.1079527707765703E+01 1.0d0 1.0d0
656    5.8358006063602978E+01 1.0d0 1.0d0
657    1.1000966104096939E+02 1.0d0 1.0d0
```

```
658    2.0848750508658674E+02 1.0d0 1.0d0
659    3.9793692474094433E+02 1.0d0 1.0d0
660    7.6660358942891128E+02 1.0d0 1.0d0
661    1.4944323346681333E+03 1.0d0 1.0d0
662    2.9570621576747585E+03 1.0d0 1.0d0
663    5.9602824302208819E+03 1.0d0 1.0d0
664 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
665    3.2777474514072306E-02 1.0d0 1.0d0
666    6.1052208024089318E-02 1.0d0 1.0d0
667    1.1371779544172864E-01 1.0d0 1.0d0
668    2.1181678224305334E-01 1.0d0 1.0d0
669    3.9455421330447510E-01 1.0d0 1.0d0
670    7.3499993917294237E-01 1.0d0 1.0d0
671    1.3694267119018748E+00 1.0d0 1.0d0
672    2.5522453323068639E+00 1.0d0 1.0d0
673    4.7592004160127050E+00 1.0d0 1.0d0
674    8.8821010163462422E+00 1.0d0 1.0d0
675    1.6598538397889143E+01 1.0d0 1.0d0
676    3.1079527707765703E+01 1.0d0 1.0d0
677    5.8358006063602978E+01 1.0d0 1.0d0
678    1.1000966104096939E+02 1.0d0 1.0d0
679    2.0848750508658674E+02 1.0d0 1.0d0
680    3.9793692474094433E+02 1.0d0 1.0d0
681    7.6660358942891128E+02 1.0d0 1.0d0
682    1.4944323346681333E+03 1.0d0 1.0d0
683    2.9570621576747585E+03 1.0d0 1.0d0
684    5.9602824302208819E+03 1.0d0 1.0d0
```

Listing A.6: *Output file for calculating the single point energy of Pa using a predefined basis set.*

Appendix B

Scripts for cudaDFRATOM

These are the scripts for automating the optimization of basis sets using cudaDFRATOM.

```bash
#!/bin/bash

sym=''
nuc=''
exepath=/home/dhenness/prophet/exe  # path to the executable
ver=dfratomgpu_binarysearch2.x      # the cudaDFRATOM executable file
alljobs=false
bottom=false
wtime=01:00:00

if [[ -z $exepath ]]; then
  echo "You must set the path to the executable"
  exit
fi

while [ $# -gt 0 ]
do
  val=$1
  shift
  case $val in
    "-a")
        alljobs=true
      ;;
    "-b")
      bottom=true
    ;;
    "-gauss")
        nuc='gauss_'
      ;;
    "-point")
        nuc='point_'
      ;;
    "-S")
        sym='S'
        low=$1
        high=$2
        shift
        shift
      ;;
    "-P")
        sym='P'
        low=$1
        high=$2
        shift
        shift
      ;;
    "-D")
        sym='D'
        low=$1
        high=$2
        shift
        shift
      ;;
    "-F")
        sym='F'
        low=$1
        high=$2
        shift
        shift
      ;;
    "-t")
        wtime=$1
        shift
```

```bash
64        ;;
65      *)
66        job=$val
67        ;;
68    esac
69  done
70
71  boilerplate(){
72    cat << EOF
73  #!/bin/bash
74  #SBATCH --account=def-mariusz
75  #SBATCH --gres=gpu:1
76  #SBATCH --mem=4GB
77  #SBATCH --time=$1
78  #SBATCH --job-name=$2
79  #SBATCH --output=%x-%j.out
80
81  echo --------------------------------------
82  echo --------------------------------------
83  echo began running on \`date\`
84  echo --------------------------------------
85  echo --------------------------------------
86  $3 $4 > $5
87  echo --------------------------------------
88  echo --------------------------------------
89  echo stopped running on \`date\`
90  echo --------------------------------------
91  echo --------------------------------------
92  EOF
93  }
94
95  if [[ -z $nuc ]]; then
96    echo No nuc entered
97    exit
98  fi
99
100 if [ "$alljobs" = true ]; then
101   job=`ls *inp`
102 elif [ -z "$job" ]; then
103   echo no job given
104   exit
105 fi
106
107 for file in $job; do
108   dir=`echo $file | sed 's/_.*$//'`
109   mkdir $dir
110   cp $file $dir/$file-orig
111   cd $dir
112
113   # keep old out files and delete others
114   rename "out" "out-old" *.out
115   rm *inp
116   rm *sh
117
118   file_base=$(echo $file | sed -e "s/$nuc.*/$nuc/")
119   read -r -a old_bs <<< $(sed -n -e "/nbs/{s/nbs=//; p}" "$file-orig")
120   read -r -a old_start <<< $(sed -n -e "/start/{s/start=//; s/\$end//; p}" "$file-orig")
121
122   # keep track of how many files are made
123   total=0
124   for i in $(seq $low $high); do
125     for j in $(seq 0 6); do
126       new_bs[j]=${old_bs[j]}
127       new_start[j]=${old_start[j]}
128     done
129
130     # make alterations based on symmetry
131
132     case $sym in
133       "F")
134         ((new_bs[5]-=$i))
135         ((new_bs[6]-=$i))
136         # make sure that the number of basis functions is not zero
137         if [[ "${new_bs[5]}" == "0" ]]; then
138           break
139         fi
140         ((new_start[5]+=$i))
141         ((new_start[6]+=$i))
142         ;;
143       "D")
144         ((new_bs[3]-=$i))
145         ((new_bs[4]-=$i))
146         if [[ "${new_bs[3]}" == "0" ]]; then
147           break
148         fi
149         ((new_start[3]+=$i))
150         ((new_start[4]+=$i))
151         ;;
152       "P")
153         ((new_bs[1]-=$i))
154         ((new_bs[2]-=$i))
155         if [[ "${new_bs[1]}" == "0" ]]; then
```

```
156          break
157        fi
158        ((new_start[1]+=$i))
159        ((new_start[2]+=$i))
160        ;;
161      "S")
162        ((new_bs[0]-=$i))
163        if [[ "${new_bs[0]}" == "0" ]]; then
164          break
165        fi
166        ((new_start[0]+=$i))
167        ;;
168    esac
169    new_file=$file_base${new_bs[0]}"s"${new_bs[1]}"p-"${new_bs[2]}"p+"${new_bs[3]}"d-"${new_bs[4]}"d+"${new_bs[5]}"f-"${new_bs[6]}"f+.inp"
170    cp $file-orig $new_file
171    sed -i -e "s/${old_bs[0]} ${old_bs[1]} ${old_bs[2]} ${old_bs[3]} ${old_bs[4]} ${old_bs[5]} ${old_bs[6]}/\
172 ${new_bs[0]} ${new_bs[1]} ${new_bs[2]} ${new_bs[3]} ${new_bs[4]} ${new_bs[5]} ${new_bs[6]}/" $new_file # continuing from last line
173    if [ "$bottom" == true ]; then
174      sed -i -e "s/${old_start[0]} ${old_start[1]} ${old_start[2]} ${old_start[3]} ${old_start[4]} ${old_start[5]} ${old_start[6]}/\
175 ${new_start[0]} ${new_start[1]} ${new_start[2]} ${new_start[3]} ${new_start[4]} ${new_start[5]} ${new_start[6]}/" $new_file # continuing from last
         line
176    fi
177
178    boilerplate $wtime ${new_file%????} $exepath/$ver $new_file ${new_file%???}log > ${new_file%???}sh
179    ((total++))
180  done
181
182
183  echo "Submitting $total input files"
184  for sub in *sh; do
185    sbatch $sub
186  done
187 done
```

Listing B.1: *BS_sub_auto.sh: Generate and submit input files.*

```
1  #!/bin/bash
2
3  tol=$1
4
5  if [ -z $tol ]; then
6    tol='5*10^-8'
7  fi
8
9  echo The tolerance in relative error is $tol
10
11 mkdir best
12 for dir in `ls -d */`; do
13   if [ "$dir" != 'old/' ]; then
14     if [ "$dir" != 'best/' ]; then
15       if [ "$dir" != 'tests/' ]; then
16       echo entering $dir
17       cd $dir
18       for file in *.inp; do
19         error=`grep -i 'rela' ${file%????}*log | tail -1 | awk '{print $7}'`
20         comperror=`echo $error | sed -e 's/E/\*10\^/'`
21         testthress=`echo "scale=20; ($tol - $comperror)*10^10 >= 1" | bc -l`
22         if [ $testthress -ge 1 ]; then
23           cp $file ../best
24           cp ${file%????}*out ../best
25           echo Found best for $file
26           break
27         fi
28       done
29       fi
30     fi
31   fi
32   cd ..
33 done
```

Listing B.2: *BS_get_best.sh: Find the best basis set.*