Variational Monte Carlo (VMC) on a 3 dimensional model of *Hookium*, a two electron harmonium in a Hooke's potential, with parallelisation on a GPU

Blaž Stojanovič

Cavendish Laboratory, Department of Physics, J J Thomson Avenue, Cambridge. CB3 0HE

Abstract

In this written assignment we implement a fully Variational Quantum Monte Carlo (VMC) code in order to study Hooke's-law atom (Hookium), a model in which the Coulomb interaction between the nucleus and electrons is replaced with a harmonic one. Numerically obtained ground state energies and position intracules are compared to analytical solutions in the special case of $k = \frac{1}{4}$. Alongside the VMC a variational Hartree-Fock method with Cartesian Gaussian basis functions is implemented, which permits calculations of the Correlation energy as well as aids in construction of trial wave functions used in the Monte Carlo method. The software is implemented in JAX, making it possible to run it on a GPU and evaluate gradients by using automatic differentiation. Energy and variance based optimisation is compared in the case of Hookium and correlation energies are calculated for a range of k's.

Contents

1. Introduction

The Schrödinger equation underpins a large part of quantum chemistry and solid state physics. However, the quantum many-body problem, which amounts to solving the $3N$-dimensional Schrödinger equation is notoriously hard to solve. Ever since the postulation of the equation in 1925, great efforts have been made in solving the equation, both analytically and numerically. Perhaps most impactful was the development of various approximate methods to solve the many-body problem with the available computational resources. Hartree-Fock (HF) approaches solve an auxiliary system of independent electrons in a self-consistent field and assume that the wave function (for fermions) can be represented as a single Slater determinant. The HF method does not include electron correlation, which makes it a good approximation only in systems where correlation contributions are small. Post-Hartree-Fock methods, such as Coupled Cluster, Configuration interaction and Møller-Plesset theory include correlation by considering a linear combination of determinants, they can be extremely accurate but come at a high computational cost.

One of the most popular approaches used today is Density Functional Theory (DFT). It reformulates the many-body electron problem in terms of the 3-dimensional electron density $n(\mathbf{r})$, which is found by minimising the total energy functional $E[n(\mathbf{r})]$ [1]. DFT provides an alternative line of thought to the truncated Hilbert space of single particle orbitals [2] and is used extensively for simulating large systems as linear scaling variants of DFT exist [3]. While DFT is theoretically exact the true energy functional $E[n(\mathbf{r})]$ is not known and its parameterisations employ more accurate *ab initio* methods. One of which being Quantum Monte Carlo (QMC).

1.1. *Quantum Monte Carlo methods*

Quantum Monte Carlo is a class of methods that use statistical sampling to directly deal with high-dimensional integration that arises from working with the many-body wave function. QMC methods are among the most accurate achieving chemical accuracy for smaller systems [4], and can achieve any degree of statistical precision sought. Quantum Monte Carlo is also very versatile and can be applied at both zero and finite temperatures [5]. The most basic zero temperature QMC method is variational QMC (VMC). The method is composed of two parts, firstly it directly evaluates the variational energy $E_V = \langle\Psi_T|\hat{H}|\Psi_T\rangle/\langle\Psi_T|\Psi_T\rangle$ of the system using Monte Carlo integration and a trial wave function Ψ_T. Secondly the parameters of the trial wave function are optimised such as to minimise the variational energy E_V, giving the method its name. The first application of VMC was to ground state ^{4}He [6], it was later extended for studying many-body fermionic systems [7]. A way of obtaining excitation energies using VMC is to use a trial wave function that models an excited state of the system, if the trial wave function obeys a certain symmetry, the variational principle guarantees that this VMC energy calculation gives an upper bound on the lowest exact eigenstate of this symmetry. Furthermore, the method can be extended to study non-equilibrium properties of bosonic [8, 9], and fermionic [10] systems. The main advantage of VMC is its simplicity while the main drawback is that the accuracy is limited by the flexibility and form of the trial wave function [5]. As such VMC is usually employed as a first step in more advanced QMC simulations.

Projector quantum Monte Carlo (PMC), is a class of QMC methods which are in essence nothing more than stochastic implementations of the power method to obtain the dominant eigenvector of a matrix or a kernel function [11]. Their distinct advantage over VMC is that they are not constrained by our parametrisation of the trial wave function, as they can describe arbitrary probability distributions. The projector $\hat{P}$ has to be chosen in such a way, that the ground state of the system becomes the dominant eigenvector, i.e. $|\Psi_0\rangle = \lim_{n\to\infty}\hat{P}^n|\Psi_T\rangle$. Different ways of achieving this, the space (real or orbital space) in which the walk is done and choosing either first or second quantisation description, give rise to different flavours of PMC methods. Using an exponential projector $\hat{P} = e^{\tau(E_T\mathbb{1}-\hat{H})}$ can be interpreted as propagation in imaginary time $\tau \longrightarrow it$ in turn transforming the Schrödinger equation into a diffusion equation, which is a continuous limit of the random walk and lends itself to stochastic integration [12]. Directly sampling from the exact Green function is known as Projector Green Function Monte Carlo (GFMC) method [13, 14]. A convenient approximation to GFMC is its short-time approximation which leads to one of the most popular QMC methods, diffusion Monte Carlo (DMC) [4, 12]. In this regime one can exploit analytical solutions to diffusion and rate problems to write an explicit form of the Green's function. Additionally, by using the Trotter-Suzuki formula time-step bias can be expressed and accounted for [5]. DMC is statistically implemented by using a population of walkers which either branch or die, the average over all walkers is calculated. Reptation quantum Monte Carlo [12] (RMC) is an alternative formulation which only uses a single walker, and instead of branching and dying the MC moves mutate the path of that single walker. The use of a guiding wave function for importance sampling greatly improves the statistical efficiency of PMC methods, the guiding wave function is usually obtained by means of VMC or some mean field calculation.

PMC method suffer from the *sign problem*, which is present in Markov chain simulation of distributions that are not strictly positive, this is the case in fermionic and frustrated systems [11]. The problem refers to an exponential decrease in sampling efficiency with system size. The search for solutions of this problem is still an area of active research [4] but is in practice remedied by the *fixed-node* approximation [15]. In it a boundary condition is imposed into the projection, such that the projected state shares the same zero crossings (nodal surface) with a trial wave function, which is again usually obtained with VMC. The projected state is now only exact when the nodal surface is exact, nevertheless this approximation is quite accurate [4]. Fixed node is widely used, one of its first applications was to the electron gas [16], which is used in parameterisations of the exchange correlation functional in LSDA [17].

Quantum Monte Carlo methods have had a lot of success at finite temperatures. Auxiliary-field Monte Carlo, or Path Integral Monte Carlo [18], which leads to ring-polymer molecular dynamics, may be used for this purpose. Additionally QMC is not limited to continuum space applications and has been extensively used to study lattice models, notable examples being the cluster/loop algorithm and the worm algorithm [11, 19].

Quantum Monte Carlo methods are generally more computationally expensive than DFT approaches, but on the other hand QMC codes are, as a rule of thumb, simpler to implement. Furthermore, since the wave function does not need to be stored directly QMC has reasonable storage requirements. The high computational cost of the QMC methods is remedied by the fact that they are intrinsically parallelisable, the core calculation involves generating (pseudo)-random numbers, performing a simple calculation and in the end averaging over the results. Therefore, implementations of QMC algorithms that have been applied to practical problems are optimised to run on massively parallel hardware with little overhead [20]. Finally, the repetitive nature of the Monte Carlo calculation lends itself to hardware acceleration using either graphical processing units (GPUs) or field-programmable gate arrays (FPGAs) [5].

1.2. *Graphical processing units*

Building customised hardware for graphics is common practice since the 1970s, but the term *graphical processing unit* (GPU) as we know it today was popularised at the turn of the millennium by NVIDIA. GPUs were originally designed with very specialised tasks in mind, but in the last two decades there has been an in increased interest in leveraging GPU for various computationally intensive applications, usually referred to as *general-purpose computing on graphical processing units* (GPGPU). The main advantage of graphical processing units over CPUs is higher theoretical *floating point operations per second* (FLOPS) and increased theoretical bandwidth. This is achieved by dedicating a larger proportion of transistors in the processing unit to data processing as opposed to flow control and data catching, as well as not performing branch checking [21], see Figure 1. GPUs were designed to work with small fragments of data (pixels) and perform the same operation many times simultaneously on all of the fragments. In the CUDA parallel programming model the problem at hand is required to be decomposable into tasks that can be independently executed in parallel by *blocks* of *threads*. The tasks must be such that they can be executed in parallel by the threads within a block. Each thread executes functions called *kernels*. A substantial speedup of a calculation can be achieved *if* it can be made to comply with the data and thread parallelism required for the above described procedure.

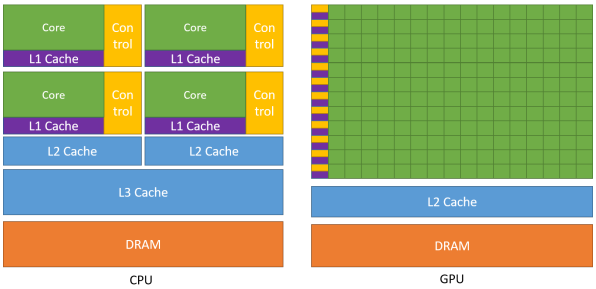


Figure 1: A comparison of a CPU and GPU, from [21].

QMC methods can be made to fit this mold and there has been extensive work done in this direction [22, 23, 24]. GPUs can be exploited by initializing multiple walkers and running them in parallel thread blocks before averaging the results over all of them. Additionally GPUs can be used to accelerate the evaluation of basis function expansions, the linear algebra associated with computing the Slater determinant and evaluation of complicated forms of the Jastrow factor in Variational Monte Carlo. Since most GPUs only support single precision arithmetic, accuracy was of concern in first applications to QMC, benchmarks have shown that chemical accuracy is still achievable on GPUs [22, 23].

1.3. *Hookium*

We can construct a toy model of two electrons that resembles the helium atom, where the electrons are bound to the nucleus with a harmonic instead of a Coulomb potential. The Hamiltonian of the model is then defined as

$$\hat{H} = -\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 + \frac{1}{2}kr_1^2 + \frac{1}{2}kr_2^2 + \frac{1}{r_{12}}. \tag{1}$$

This model is referred to as *Hookium* (sometimes Harmonium). Although it is quantitatively different from helium some meaningful qualitative comparisons can be made [25]. Most notably Hookium has an analytical solution for the case $k = \frac{1}{4}$, making it an ideal candidate to benchmark computational methods, which is

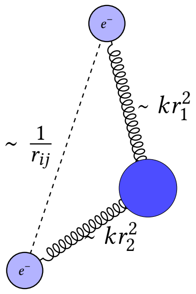


Figure 2: Hookium atom, two electrons that are bound to the nucleus with springs of coefficient k and interact via the Coulomb force.

the topic of this written assignment. The normalized closed form wave function solution of the ground state in position space is

$$\Psi(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{2\sqrt{8\pi^{5/2} + 5\pi^3}} \left(1 + \frac{r_{12}}{2}\right) \exp\left(-\frac{r_1^2 + r_2^2}{4}\right), \tag{2}$$

the energy of the state is $E = 2$. The probability of finding two electrons distance u apart; the position intractule $P(u)$, is also useful. It is defined as

$$P(u) = \iiint |\Psi(\mathbf{r}_1, \mathbf{r}_2)|^2 \, \delta(\mathbf{r}_{12} - \mathbf{u}) \, d\mathbf{r}_1 d\mathbf{r}_2 d\Omega_{\mathbf{u}}, \tag{3}$$

for Hookium it too has a closed form

$$P(u) = \frac{1}{8 + 5\pi^{1/2}} u^2 \left(1 + \frac{u}{2}\right)^2 \exp\left(-\frac{u^2}{4}\right). \tag{4}$$

Values of $h = m = c = 1$ and $1E_h = 1$ will be assumed throughout this report. Hookium will be studied with the VMC method and the above expressions will be used to compare the accuracy of the numerical method.

The rest of this written assignment is structured as follows. In section 2, the details necessary to implement the variational Monte Carlo method are presented. Additionally details of GPU acceleration and parallelisation of the method are discussed. Presentation of numerical results and benchmarks of the method follow in section 3. The final section 4 contains a short conclusion and proposals for further work.

2. Implementation

2.1. Monte Carlo Importance Sampling

The most common application of Monte Carlo methods is evaluation of integrals in high dimensional space. There MC has a distinct advantage over quadrature methods, as the statistical error decreases with the square root of samples irregardless of the dimensionality of the problem. Integrals of a function $g(\mathbf{R})$

$$I = \int g(\mathbf{R}) \mathrm{d}\mathbf{R}, \tag{5}$$

where $\mathbf{R}$ is the *configuration* of the system or simply a *walker*, can be integrated by use of an *importance function* $\mathrm{P}(\mathbf{R})$, where $\int d\mathbf{R}\mathrm{P}(\mathbf{R}) = 1$ and $\mathrm{P}(\mathbf{R}) \geq 0$. The integral can be rewritten in the form

$$\int g(\mathbf{R}) \mathrm{d}\mathbf{R} = \int \frac{g(\mathbf{R})}{\mathrm{P}(\mathbf{R})} \mathrm{P}(\mathbf{R}) \mathrm{d}\mathbf{R} = \int f(\mathbf{R}) \mathrm{P}(\mathbf{R}) \mathrm{d}\mathbf{R}, \tag{6}$$

where $f(\mathbf{R}) = g(\mathbf{R})/\mathrm{P}(\mathbf{R})$. The importance function $\mathrm{P}(\mathbf{R})$ can be interpreted as a probability density. If we now generate an infinite number of random uncorrelated configurations $\mathbf{R}_m$ from the distribution $\mathrm{P}(\mathbf{R})$, the sample average is a good estimator of the integral I

$$I = \lim_{M \to \infty} \left\{ \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{R}_m) \right\}, \tag{7}$$

and for an approximation with a finite number of samples

$$I \approx \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{R}_m). \tag{8}$$

Under conditions where the central limit theorem holds [4], the estimator is normally distributed with variance σ_f^2/M, which can also be estimated from the samples as

$$\frac{\sigma_f^2}{M} \approx \frac{1}{M(M-1)} \sum_{m=1}^{M} \left[f(\mathbf{R}_m) - \frac{1}{M} \sum_{n=1}^{M} f(\mathbf{R}_n) \right]^2. \tag{9}$$

In the case of Hookium the configurations $\mathbf{R}$ are the positions of the electrons.

2.2. Metropolis-Hastings algorithm

The integration technique from the previous section relies on our ability to obtain samples from a probability distribution $\mathrm{P}(\mathbf{R})$. In the case of QMC these distributions are high-dimensional and cannot be directly sampled from, moreover their normalisations are usually not known. The Metropolis-Hastings algorithm [26], see Algorithm 1, avoids direct sampling from the distribution $\mathrm{P}(\mathbf{R})$ and is insensitive to its normalisation. It uses a Markov process whose stationary distribution $\pi(\mathbf{R})$ is the same as $\mathrm{P}(\mathbf{R})$ to generate a sequence of configurations $\{\mathbf{R}_n\}_\mathrm{P}$ that are drawn from $\mathrm{P}(\mathbf{R})$. A Markov process is completely defined with its transition probability $\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}')$, which is the probability of transitioning from state $\mathbf{R}$ to state $\mathbf{R}'$. For the process to have a unique stationary distribution two conditions must be met, the process must be *ergodic* and it must obey *detailed balance*

$$\mathrm{P}(\mathbf{R})\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}') = \mathrm{P}(\mathbf{R}')\mathrm{P}(\mathbf{R}' \longrightarrow \mathbf{R}), \tag{10}$$

rewritten as

$$\frac{\mathrm{P}(\mathbf{R})}{\mathrm{P}(\mathbf{R}')} = \frac{\mathrm{P}(\mathbf{R}' \longrightarrow \mathbf{R})}{\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}')}. \tag{11}$$

The right transition probability $\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}')$ is not known, but we can express it with a trial move transition probability $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')$ which we sample and acceptance probability $\mathrm{A}(\mathbf{R} \longrightarrow \mathbf{R}')$ as

$$\mathrm{P}(\mathbf{R} \longrightarrow \mathbf{R}') = \mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')\mathrm{A}(\mathbf{R} \longrightarrow \mathbf{R}'). \tag{12}$$

For equation (11) to hold, the acceptance probability must be

$$A\left(\mathbf{R} \longrightarrow \mathbf{R}'\right) = \min\left(1, \frac{\mathrm{T}\left(\mathbf{R}' \longrightarrow \mathbf{R}\right)\mathrm{P}\left(\mathbf{R}'\right)}{\mathrm{T}\left(\mathbf{R} \longrightarrow \mathbf{R}'\right)\mathrm{P}(\mathbf{R})}\right). \tag{13}$$

Thus to sample from any probability distribution we need only have the ability to calculate probabilities $\mathrm{P}(\mathbf{R})$ and to sample from a trial transition probability $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')$. The efficiency of the algorithm depends on the amount of trial moves that we reject. All trial moves would be accepted if $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}') = \mathrm{P}(\mathbf{R}')$, which would just mean sampling from P directly and is the very problem we are trying to solve with Metropolis-Hastings.

Algorithm 1: Metropolis-Hastings

```
Result: A set of configurations $\{\mathbf{R}_n\}_\mathrm{P}$ sampled
        from P
Initialize walker at random configuration $\mathbf{R}$;
while no. samples less than N do
    Generate new configuration $\mathbf{R}'$ with
      transition probability $\mathrm{T}(\mathbf{R} \longrightarrow \mathbf{R}')$;
    Accept the move $(\mathbf{R} \longrightarrow \mathbf{R}')$ with probability
      $A\left(\mathbf{R} \longrightarrow \mathbf{R}'\right) = \min\left(1, \frac{\mathrm{T}(\mathbf{R}' \to \mathbf{R})\mathrm{P}(\mathbf{R}')}{\mathrm{T}(\mathbf{R} \to \mathbf{R}')\mathrm{P}(\mathbf{R})}\right)$;
    Append $\mathbf{R}$ to the set of confifigurations;
end
```

2.3. Variational Quantum Monte Carlo

Variational quantum Monte Carlo uses a trial wave function Ψ_T which is an approximation to the true ground state wave function to directly evaluate the expectation value of $\hat{H}$ which provides an upper bound on the ground state energy

$$E_V = \frac{\langle \Psi_T | H | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} = \frac{\int \Psi_T^*(\mathbf{R})\hat{H}\Psi_T(\mathbf{R})d\mathbf{R}}{\int \Psi_T^*(\mathbf{R})\Psi_T(\mathbf{R})d\mathbf{R}} \geq E_0. \tag{14}$$

The expression for the variational energy E_V can be rewritten as

$$E_V = \frac{\int |\Psi_T(\mathbf{R})|^2 \left(\Psi_T(\mathbf{R})^{-1}\hat{H}\Psi_T(\mathbf{R})\right) d\mathbf{R}}{\int |\Psi_T(\mathbf{R})|^2 \, d\mathbf{R}}. \tag{15}$$

The above integral is estimated by using Metropolis-Hastings to sample a set of configurations $\{\mathbf{R}_n\}_\mathrm{P}$ from the probability distribution given by the (normalized) trial wave function as $\mathrm{P}(\mathbf{R}) = |\Psi_T(\mathbf{R})|^2 \mathrm{d}\mathbf{R}$ and averaging these *local* E_L contributions

$$E_V \approx \frac{1}{M}\sum_{m=1}^{M} E_L\left(\mathbf{R}_m\right), \tag{16}$$

where

$$E_L(\mathbf{R}) = \Psi_T(\mathbf{R})^{-1}\hat{H}\Psi_T(\mathbf{R}). \tag{17}$$

The procedure is analogous for any other calculation of expectation value. Trial moves may be chosen in a variety of ways depending on the system studied, in the case of Hookium we will use a Gaussian distribution centered at the current position of the walker.

Estimation of the variational energy E_V is only one part of a variational Monte Carlo simulation. The second part is the variational optimisation of the trial wave function. The trial wave function Ψ_T is parameterized with a set of variational parameters $\{\alpha_k\}$, historically the number of parameters has been low due to high computational cost [4]. The optimal parameters for the system are found by minimizing the *cost function*. A straightforward choice of cost function is the variational energy E_V in eq. (15). Given that its value is bounded below due to the variational principle, eq (14), its minimisation gives parameters $\{\alpha_k\}$ that give the best energy estimate for given parameterisation. An alternative is to minimize the variance of energy

$$\sigma_E^2(\{\alpha_k\}) = \frac{\int \Psi_T^2(\{\alpha_k\})\left[E_L(\{\alpha_k\}) - E_V(\{\alpha_k\})\right]^2 d\mathbf{R}}{\int \Psi_T^2(\{\alpha_k\})d\mathbf{R}}, \tag{18}$$

this minimizes the statistical error of VMC estimation of energy. Most practical calculations are done by minimizing energy variance [4]. Minimisation of energy variance works because of the *zero-variance* property, which is exclusive for quantum expectation values. If the trial wave function Ψ_T is an exact eigenfunction of the Hamiltonian

$$\hat{H}|\Psi_T\rangle = E_V|\Psi_T\rangle, \tag{19}$$

then the local energy E_L, a random variable, does not depend on the sampled configuration $\mathbf{R}$

$$E_L(\mathbf{R}) = \Psi_T(\mathbf{R})^{-1}\hat{H}\Psi_T(\mathbf{R}) = \Psi_T(\mathbf{R})^{-1}E_V\Psi_T(\mathbf{R}) = E_V, \tag{20}$$

is constant and hence has zero variance. This equality holds only when Ψ_T is an exact eigenfunction of the Hamiltonian. However, zero-variance property has important consequences for numerical stability of optimisation, it means that energy variance minima are robust to finite sampling. Minimizing the variance of energy drives the trial wave function towards eigenstates of the Hamiltonian. Moreover, the statistical error associated with estimation of any expectation value $\langle\hat{O}\rangle$ is proportional to the variance of $\hat{O}$, one can use the zero-variance condition to define a renormalized observable $\tilde{O}$ with the same average and smaller variance [27] for more efficient sampling.

Various approaches to minimize the cost function can be taken, the simplest is trial and error using simple fitting procedures, this only works for small numbers of parameters. Alternatively a reweighting technique can be used to evaluate the energy or energy variance of a wave function with slightly different parameters $\Psi_T(\{\alpha + \delta\alpha\})$ to the one already evaluated $\Psi_T(\{\alpha\})$ [28], this increases the number of variational parameters that can be treated in small systems. Another option is to evaluate energy derivatives and use some sort of stochastic optimisation technique, *stochastic gradient descent* being the simplest and *stochastic recofiguration* [29] being a more elaborate alternative.

2.3.1. Trial wave functions Ψ_T

The choice of trial wave function Ψ_T is the limiting factor for the performance of VMC, it determines its statistical efficiency and its final accuracy. The choice of trial wave function flexible but must satisfy the following conditions [4]; both the wave function and its gradient must be finite where the potential is finite, the wave function must have the appropriate symmetry and integrals

$$\int \Psi_T^*\Psi_T, \quad \int \Psi_T^*\hat{H}\Psi_T \quad \text{and} \quad \int \Psi_T^*\hat{H}^2\Psi_T \tag{21}$$

must exist. A popular choice and the one we will use in this project is the *Slater-Jastrow* state. The trial wave function is a product of a Slater determinant $D(\mathbf{R})$ of single particle states $\{\psi_k(\mathbf{r})\}$

$$D(\mathbf{X}) = \frac{1}{\sqrt{N!}}\begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \cdots & \psi_N(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \cdots & \psi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x}_N) & \psi_2(\mathbf{x}_N) & \cdots & \psi_N(\mathbf{x}_N) \end{vmatrix} \tag{22}$$

and the *Jastrow correlation factor* $J(\mathbf{X})$

$$\Psi_{SJ}(\mathbf{X}) = e^{J(\mathbf{X})}D(\mathbf{X}), \tag{23}$$

where $\mathbf{X}$ is a configuration that contains both the spin and position degrees of freedom $\mathbf{x}_i = (\mathbf{r}_i, s_i)$. The discriminant is usually decomposed into spin up and down components as

$$D(\mathbf{X}) = D^{\uparrow}\left(\mathbf{r}_1, \ldots, \mathbf{r}_{N_\uparrow}\right) D^{\downarrow}\left(\mathbf{r}_{N_\uparrow+1}, \ldots, \mathbf{r}_N\right), \tag{24}$$

this is computationally beneficial as it results in smaller determinants and no explicit sum over spin. The Jastrow factor is heuristically constructed to incorporate electron correlation into the wave function. In most practical calculations it is limited to one- and two-body terms [4]

$$J(\mathbf{R}) = \underbrace{\sum_{i=1}^{N}\chi(\mathbf{r}_i)}_{\text{one-body}} - \underbrace{\frac{1}{2}\sum_{i=1}^{N}\sum_{j<i}^{N} u\left(\mathbf{r}_i, \mathbf{r}_j\right)}_{\text{two-body}}. \tag{25}$$

The first term describes the nuclear-electronic correlation and the second electron-electron correlation, in this project we will focus only on the second term. Various forms of $u(\mathbf{r}_i, \mathbf{r}_j)$ exist depending on application area, a common form for atomic and molecular calculations is

$$u(r_{ij}) = \frac{a_{ij}r_{ij}}{1 + \sum_{\kappa=1} b_{ij}^{(\kappa)} r_{ij}^{\kappa}} \tag{26}$$

The parameters a_{ij} are chosen to satisfy the *cusp conditions*,

$$\left.\frac{du}{dr}\right|_{r=0} = \begin{cases} -\frac{1}{2}, & \text{for opposite spins,} \\ -\frac{1}{4}, & \text{for parallel spins.} \end{cases} \tag{27}$$

and the $b_{ij}^{(\kappa)}$ parameters remain to be optimised variationally.

2.3.2. Basis sets

There are various choices of single particle states $\{\psi_k(\mathbf{r})\}$ that compose the Slater determinant. They can be obtained from less expensive electronic structure methods, popular choices being LDA or HF, they can be from a standard basis set, such as atomic orbitals, Gaussian or plane wave sets, or they can be designed for the particular case at hand. In this work, we will compare performance of HF orbitals obtained from Cartesian Gaussian-type orbitals (GTOs)

$$G_{ijk}(\mathbf{r}, \alpha, \mathbf{A}) = (x - x_A)^i (y - y_A)^j (z - z_A)^k e^{-\alpha r_A^2}, \quad (28)$$

and HF orbitals obtained from the harmonic oscillator basis set

$$\phi_k(r) = \frac{H_{2k-1}(r/\sqrt{2})}{2^k \sqrt{(2k-1)!} r/\sqrt{2}} \frac{\exp\left(-r^2/4\right)}{(2\pi)^{3/4}}. \quad (29)$$

Both basis sets and implementation of Hartree-Fock are discussed in more detail in Appendix B and Appendix A.

2.3.3. Generating trial moves

When proposing trial moves, we would like to minimize the correlation between subsequent states $\mathbf{R}$ and $\mathbf{R}'$ as well as maximize the acceptance probability $A(\mathbf{R} \longrightarrow \mathbf{R}')$. Trial move probability determines both the size and average acceptance of the moves, practice we seek a compromise between both. First proposal probabilities were symmetric, meaning that the acceptance probability factors into

$$A\left(\mathbf{R} \longrightarrow \mathbf{R}'\right) = \min\left(1, \frac{\mathrm{P(R')}}{\mathrm{P(R)}}\right) = \min\left(1, \frac{|\Psi(\mathbf{R}')|^2}{|\Psi(\mathbf{R})|^2}\right). \quad (30)$$

However it is beneficial to use information from the probability distribution to aid sampling. A common choice of non-symmetric trial move probability and the one we will use is

$$\mathrm{T}\left(\mathbf{R} \rightarrow \mathbf{R}'\right) = \frac{1}{(2\pi\tau)^{3N/2}} \exp\left(-\frac{\left(\mathbf{R}' - \mathbf{R} - \mathbf{v}\left(\mathbf{R}\right)\tau\right)^2}{2\tau}\right), \quad (31)$$

where τ is the time step and $\mathbf{v}(\mathbf{R}) = \nabla\Psi(\mathbf{R})/\Psi(\mathbf{R})$ is the drift velocity, that drives the Markov random walk in the direction of increasing $|\Psi(\mathrm{R})|$ [11].

2.4. Technical Considerations

Before implementing the variational Monte Carlo method in computer code, we must first consider what we want our software to achieve. The main developing principle during this written assignment was to implement VMC in a way that would facilitate the use of rich variational *ansatzes* without much hassle, all while maintaining enough speed to perform optimisation in a reasonable time. Moreover we would like the code to be applicable to systems beyond just the Harmonium atom, be modular so as to quickly add new features if needed and we want the most computationally demanding parts of the code to run on the GPU. This leaves us with seemingly opposing design goals. On one hand, the flexibility of the trial wave function should mean that the user only needs to supply the functional form of the wave function and parameters to be optimised, and not gradients and Laplacians of the trial wave function, they should be estimated automatically. On the other hand however, the code needs to be quick and run on a GPU, therefore compiled. While possible, perhaps with some sort of generic programming in C++ extended with CUDA, this sort of design would be painful to implement to say the least. This dichotomy is aided by use of modern frameworks that have been designed for use in Machine Learning where there is similar tension between speed and flexibility. The best choice for this particular case is *JAX*, which is a Python library that extends *numpy* with *function transformations*, which enables use of *automatic differentiation* [30] for gradient and Hessian estimation and use of the *Accelerated Linear Algebra* (XLA) compiler, all while maintaining the expressive nature of Python.

2.4.1. Software structure

The design of the VMC software is displayed on Figure 3, It is divided into two modules that can but need not be, used together. The first module is meant either for standalone Hartree-Fock calculations or for the construction of the HF basis that is to be used in the Slater determinant part of the trial wave function in VMC. The HF procedure is given two inputs. One is the System which specifies the nuclear positions and types of nucleus (i.e. Hooke or Coulomb potential) along with parameters $\{\alpha_j\}_{j=0}^{j=k-1}$ which parameterize the geometry of the problem and are to be optimised. The second input is the Basis, i.e. an expansion of Cartesian Gaussians in eq. (28), parameterized by $\{\alpha_j\}_{j=k}^{j=N}$ which are optimised, the Basis also provides HF the means of evaluating integrals V_{pq}, T_{pq}, S_{pq} and V_{pqrs}. The HF procedure itself is described in Appendix A, vanilla *JAX* has no problem differentiating through any part of the process, except through solving generalized eigenvalue problem ($F^{(i+1)} \longrightarrow D^{(i+1)}$) which occurs once each iteration of the self-consistent loop. For this purpose we used *jax.custom_jvp* and *jax.custom_vjp* implemented in [31] which use derivation from [32]. The forward mode computed derivatives w.r.t $\{\alpha_j\}_{j=0}^{j=K}$ are then used to perform gradient based optimisation of the Hartree-Fock energy.

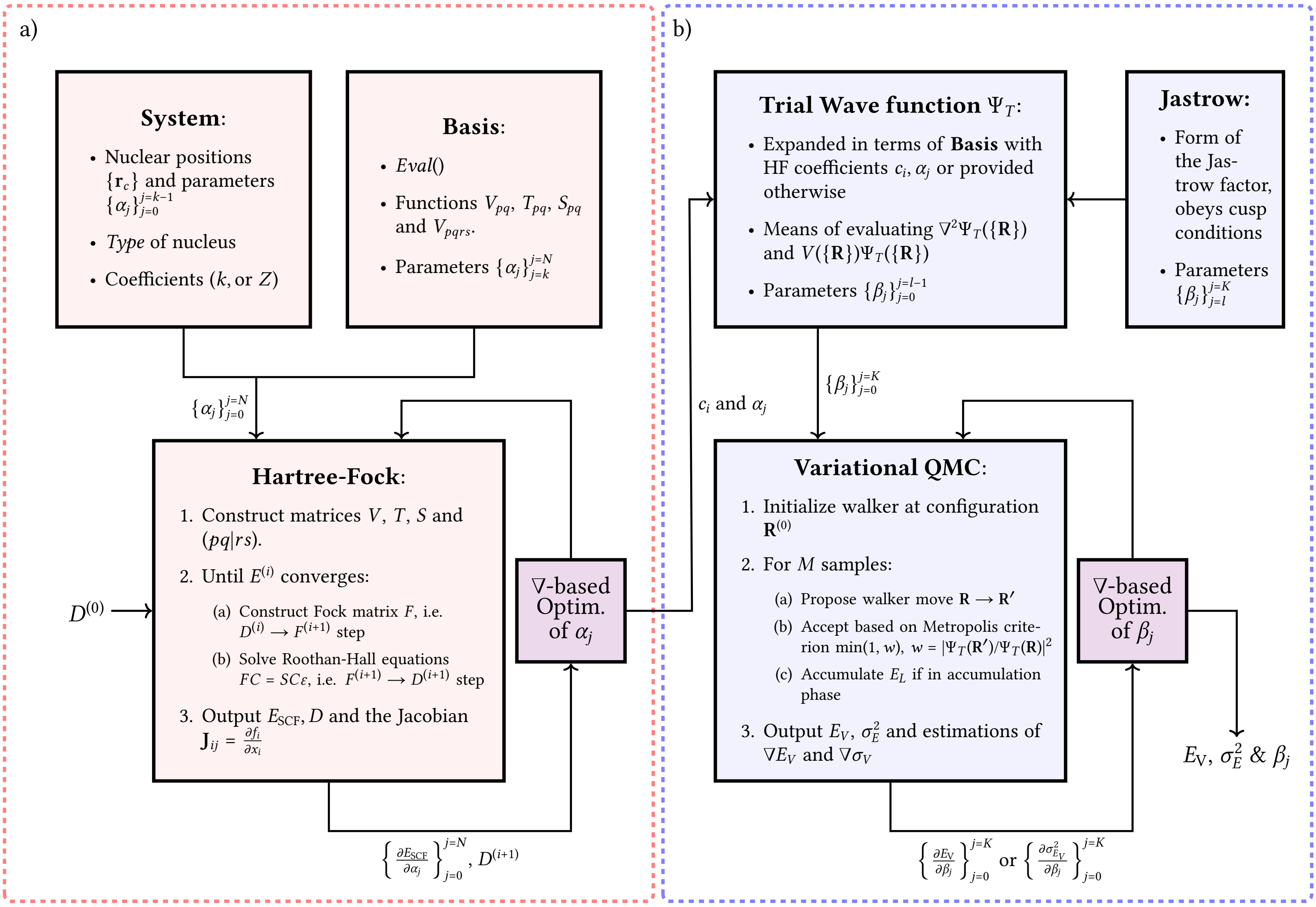

a)
System:
Nuclear positions $\{\mathbf{r}_c\}$ and parameters $\{\alpha_j\}_{j=0}^{j=k-1}$
Type of nucleus
Coefficients $(k,$ or $Z)$
Basis:
Eval()
Functions V_{pq}, T_{pq}, S_{pq} and V_{pqrs}.
Parameters $\{\alpha_j\}_{j=k}^{j=N}$
$\{\alpha_j\}_{j=0}^{j=N}$
$D^{(0)}$
Hartree-Fock:
1. Construct matrices V, T, S and $(pq|rs)$.
2. Until $E^{(i)}$ converges:
(a) Construct Fock matrix F, i.e. $D^{(i)} \longrightarrow F^{(i+1)}$ step
(b) Solve Roothan-Hall equations $FC = SC\varepsilon$, i.e. $F^{(i+1)} \longrightarrow D^{(i+1)}$ step
3. Output E_{SCF}, D and the Jacobian $\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_i}$
∇-based Optim. of α_j
$\left\{ \frac{\partial E_{\text{SCF}}}{\partial \alpha_j} \right\}_{j=0}^{j=N}, D^{(i+1)}$
b)
Trial Wave function Ψ_T:
Expanded in terms of **Basis** with HF coefficients c_i, α_j or provided otherwise
Means of evaluating $\nabla^2\Psi_T(\{\mathbf{R}\})$ and $V(\{\mathbf{R}\})\Psi_T(\{\mathbf{R}\})$
Parameters $\{\beta_j\}_{j=0}^{j=l-1}$
Jastrow:
Form of the Jastrow factor, obeys cusp conditions
Parameters $\{\beta_j\}_{j=l}^{j=K}$
c_i and α_j
$\{\beta_j\}_{j=0}^{j=K}$
Variational QMC:
1. Initialize walker at configuration $\mathbf{R}^{(0)}$
2. For M samples:
(a) Propose walker move $\mathbf{R} \longrightarrow \mathbf{R}'$
(b) Accept based on Metropolis criterion $\min(1, w)$, $w = |\Psi_T(\mathbf{R}')/\Psi_T(\mathbf{R})|^2$
(c) Accumulate E_L if in accumulation phase
3. Output E_V, σ_E^2 and estimations of ∇E_V and $\nabla\sigma_V$
∇-based Optim. of β_j
E_{V}, σ_E^2 & β_j
$\left\{ \frac{\partial E_{\text{V}}}{\partial \beta_j} \right\}_{j=0}^{j=K}$ or $\left\{ \frac{\partial \sigma_{E_V}^2}{\partial \beta_j} \right\}_{j=0}^{j=K}$

Figure 3: Diagram of the general software framework developed to solve the Hookium atom with VMC. It is comprised of two modules, that need not be used together. Both the *a*) Hartree-Fock and *b*) the VMC module are variational and allow for rich variational approaches to a particular problem.

The second module performs the Monte Carlo estimation of the variational energy and its gradients. As input it receives the trial wave function (TWF) with variational parameters $\{\beta_j\}_{j=0}^{j=K}$ which include both parameters in the Slater determinant and those in the Jastrow factor. The Laplacian of the TWF as well as its gradient, which is used for more efficient sampling, are calculated using automatic differentiation. No Monte Carlo estimators of the gradients were implemented in the main procedure, it was found that gradients of the variational energy E_V and its variance σ_E estimated via automatic differentiation by unrolling the loop over samples were sufficient for the purposes of this article. Moreover, their evaluation is very quick in XLA compiled code. The optimisation procedure can in essence be any gradient based optimisation method. Furthermore, with minor extensions to the code the procedure could output the Hessian alongside the gradients to access more powerful optimisation techniques, in this report most results are produced with plain gradient descent. The most computationally demanding part of the VMC calculation is the sampling loop, as very large numbers of samples are needed due to the stochastic nature of the process. Fortunately the loop can be sped up massively, firstly by compiling the automatic differentiation to run on a hardware accelerator (XLA allows to compile for TPUs as well), and secondly the loop can be broken down into multiple walkers propagating independently before their energies are accumulated.

3. Results and analysis

3.1. Hookium

3.1.1. HF energy

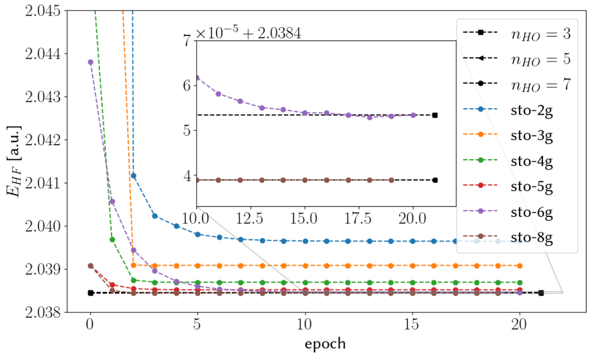


Figure 4: Optimisation of bases with different number of Gaussian basis functions for the Hartree-Fock calculation of Hookium. The exponents α_j of the Gaussians were initially set to STO-ng optimised values for the s-orbital of Hellium. HF energies calculated using different number of Harmonic Oscillator basis functions n_{HO} are given as reference, values from [25].

3.1.2. Variational Monte Carlo

3.1.3. CPU vs. GPU speedup

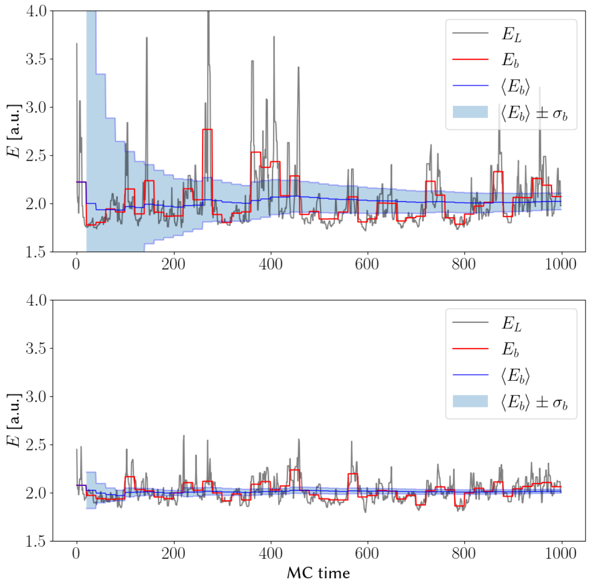


Figure 5: Comparison of Monte Carlo local energies E_L from a trial run with (top) Slater trial wave function and (bottom) unoptimised Slater-Jastrow trial wave function. The difference in variance and hence sampling efficiency is evident from the reblocked average energy $\langle E_b \rangle$ and reblocked variance σ_b.

4. Conclusion

4.1. Code expandability and complexity

Remember that HF is slow as hell!

Bibliography

[1] P. Hohenberg, W. Kohn, Inhomogeneous electron gas, Physical review 136 (3B) (1964) B864.

[2] W. Kohn, Nobel lecture: Electronic structure of matter—wave functions and density functionals, Reviews of Modern Physics 71 (5) (1999) 1253.

[3] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, M. C. Payne, Introducing onetep: Linear-scaling density functional simulations on parallel computers, The Journal of chemical physics 122 (8) (2005) 084119.

[4] W. Foulkes, L. Mitas, R. Needs, G. Rajagopal, Quantum monte carlo simulations of solids, Reviews of Modern Physics 73 (1) (2001) 33.

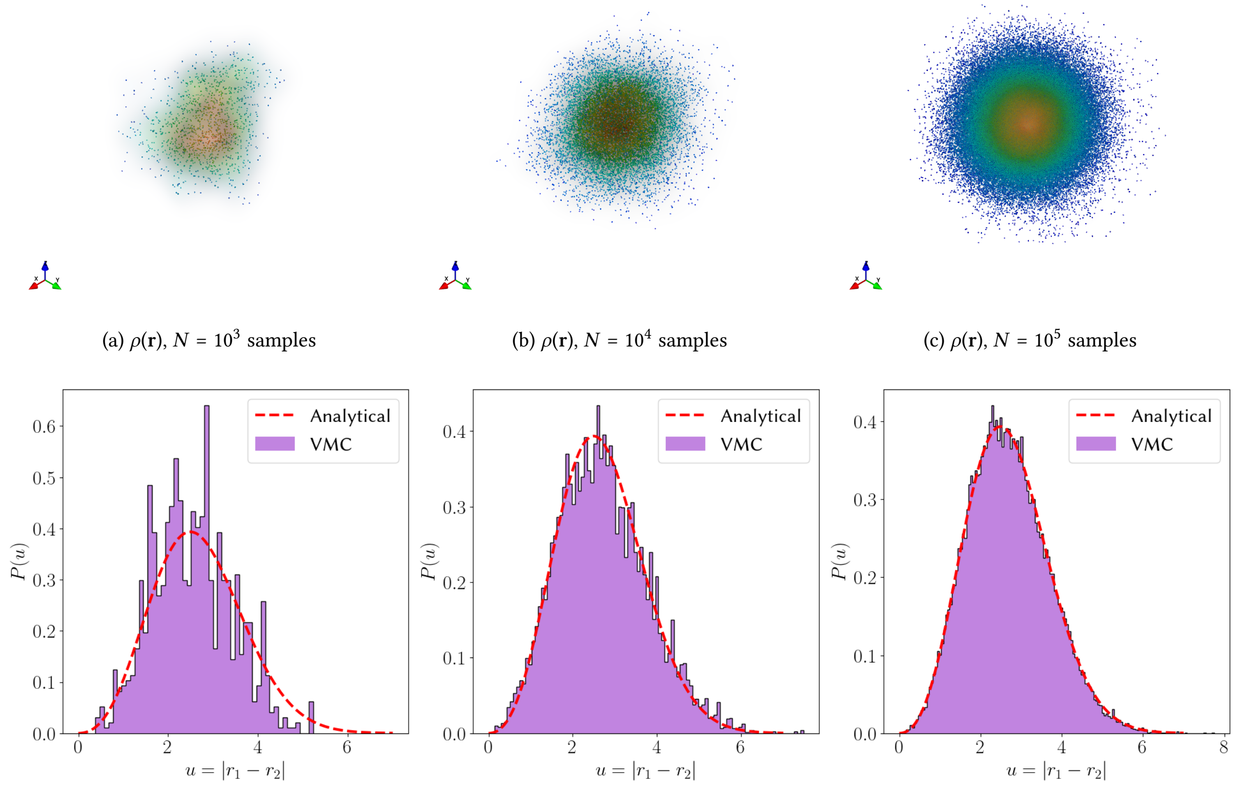
[5] B. M. Austin, D. Y. Zubarev, W. A. Lester Jr, Quantum monte carlo and related approaches, Chemical reviews 112 (1) (2012) 263–288.

[6] W. L. McMillan, Ground state of liquid he 4, Physical Review 138 (2A) (1965) A442.

[7] D. Ceperley, G. V. Chester, M. H. Kalos, Monte carlo simulation of a many-fermion study, Physical Review B 16 (7) (1977) 3081.

[8] G. Carleo, F. Becca, M. Schiró, M. Fabrizio, Localization and glassy dynamics of many-body quantum systems, Scientific reports 2 (1) (2012) 1–6.

[9] G. Carleo, F. Becca, L. Sanchez-Palencia, S. Sorella, M. Fabrizio, Light-cone effect and supersonic correlations in one-and two-dimensional bosonic superfluids, Physical Review A 89 (3) (2014) 031602.



(a) $\rho(\mathbf{r})$, $N = 10^3$ samples

(b) $\rho(\mathbf{r})$, $N = 10^4$ samples

(c) $\rho(\mathbf{r})$, $N = 10^5$ samples

(d) $P(u)$, $N = 10^3$ samples

(e) $P(u)$, $N = 10^4$ samples

(f) $P(u)$, $N = 10^5$ samples

Figure 6: Top row: Electron density of the Hookium atom, sampled from a single Gaussian trial function, with optimised exponent and two term Jastrow factor. Bottom row: The position intracule of the samples in top row, compared with the known analytical form.

[10] K. Ido, T. Ohgoe, M. Imada, Time-dependent many-variable variational monte carlo method for nonequilibrium strongly correlated electron systems, Physical Review B 92 (24) (2015) 245106.

[11] J. Gubernatis, N. Kawashima, P. Werner, Quantum Monte Carlo Methods: Algorithms for Lattice Models, Cambridge University Press, 2016. doi:10.1017/CBO9780511902581.

[12] P. J. Reynolds, J. Tobochnik, H. Gould, Diffusion quantum monte carlo, Computers in Physics 4 (6) (1990) 662–668.

[13] M. Kalos, Monte carlo calculations of the ground state of three- and four-body nuclei, Physical Review 128 (4) (1962) 1791.

[14] M. Kalos, Stochastic wave function for atomic helium, Journal of Computational Physics 1 (2) (1966) 257–276.

[15] J. B. Anderson, A random-walk simulation of the schrödinger equation: H+ 3, The Journal of Chemical Physics 63 (4) (1975) 1499–1503.

[16] D. M. Ceperley, B. J. Alder, Ground state of the electron gas by a stochastic method, Physical review letters 45 (7) (1980) 566.

[17] S. H. Vosko, L. Wilk, M. Nusair, Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis, Canadian Journal of physics 58 (8) (1980) 1200–1211.

[18] D. M. Ceperley, Path integrals in the theory of condensed helium, Reviews of Modern Physics 67 (2) (1995) 279.

[19] N. Prokof'Ev, B. Svistunov, I. Tupitsyn, Exact, complete, and universal continuous-time worldline monte carlo approach to the statistics of discrete quantum systems, Journal of Experimental and Theoretical Physics 87 (2) (1998) 310–321.

[20] R. Needs, M. Towler, N. Drummond, P. Lopez Rios, J. Trail, Variational and diffusion quantum monte carlo calculations with the casino code, The Journal of chemical physics 152 (15) (2020) 154106.

[21] Cuda c++ programming guide, https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.

[22] A. G. Anderson, W. A. Goddard III, P. Schröder, Quantum monte carlo on graphical processing units, Computer Physics Communications 177 (3) (2007) 298–306.

[23] J. S. Meredith, G. Alvarez, T. A. Maier, T. C. Schulthess, J. S. Vetter, Accuracy and performance of graphics processors: A quantum monte carlo application case study, Parallel Computing 35 (3) (2009) 151–163.

[24] K. Esler, J. Kim, D. Ceperley, L. Shulenburger, Accelerating quantum monte carlo simulations of real materials on gpu clusters, Computing in Science & Engineering 14 (1) (2010) 40–51.

[25] D. P. O'Neill, P. M. Gill, Wave functions and two-electron probability distributions of the hooke's-law atom and helium, Physical Review A 68 (2) (2003) 022505.

Basis	Optim.	$b_{ij}^{(1)}$	$b_{ij}^{(2)}$	$E_V \pm \sigma_E$
STO-2g + J1	✗	\	\	2.375010 ± 0.783208
STO-2g + J1	✓	-0.06367405	\	2.144614 ± 0.508516
STO-4g + J1	✗	\	\	2.024604 ± 0.237808
STO-4g + J1	✓	0.32431903	\	2.009259 ± 0.169574
1g + J1	✓	0.20198098	\	2.000155 ± 0.012942
1g + J2	✓	0.22442447	-0.00406822	2.000112 ± 0.006640

Table 1: Ground state energies for different trial wave functions, calculated with 10^6 samples. The second column indicates if the Jastrow factor was optimised.

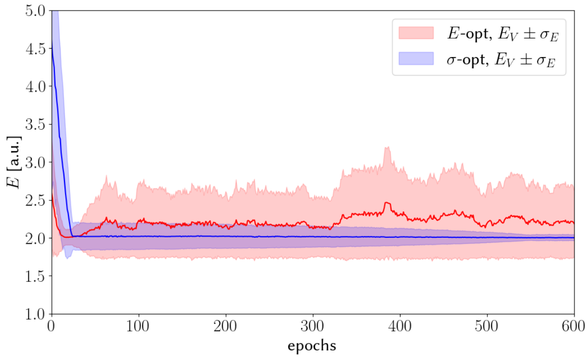


Figure 7: Comparison of energy (red) and variance (blue) optimisation of a 1g+J1 trial wave function over 600 epochs with 10^5 samples. Variance optimisation is much more robust and finds optimal parameters ($\alpha = 0.25, b_{12} = 0.202$) from a very bad initialisation.

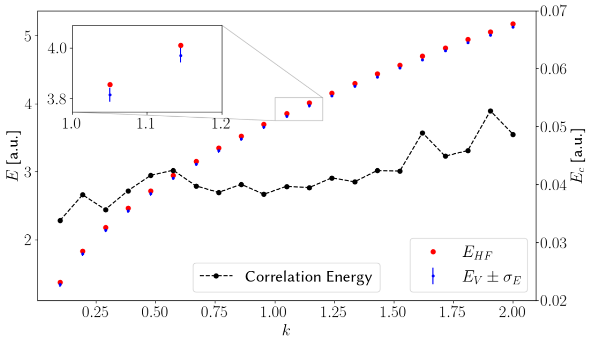


Figure 8: Correlation energy for different Hooke coefficients $k \in [0.5, 2.0]$. The Hartree-Fock energy was computed with sto-6g basis optimised over 15 epochs with learning rate $\alpha_{HF} = 0.05$ and the true ground state with 1g+J1 basis optimised over 10^3 epochs with learning rate $\alpha = 0.005$ and 10^5 samples.

[26] W. K. Hastings, Monte carlo sampling methods using markov chains and their applications (1970).

[27] R. Assaraf, M. Caffarel, Zero-variance principle for monte carlo algorithms, Physical review letters 83 (23) (1999) 4682.

[28] C. Umrigar, K. Wilson, J. Wilkins, Optimized trial wave functions for quantum monte carlo calculations, Physical Review Letters 60 (17) (1988) 1719.

[29] S. Sorella, Green function monte carlo with stochastic reconfiguration, Physical review letters 80 (20) (1998) 4558.

[30] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: a survey, Journal of machine learning research 18 (2018).

[31] J. Dominic, Generalized eigenvalue problem derivatives, https://jackd.github.io/posts/generalized-eig-jvp/.

[32] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, R. Haeb-Umbach, On the computation of complex-valued gradients with application to statistically optimum beamforming, arXiv preprint arXiv:1701.00392 (2017).

[33] L. E. McMurchie, E. R. Davidson, One-and two-electron integrals over cartesian gaussian functions, Journal of Computational Physics 26 (2) (1978) 218–231.

[34] S. Obara, A. Saika, Efficient recursive computation of molecular integrals over cartesian gaussian functions, The Journal of chemical physics 84 (7) (1986) 3963–3974.

[35] J. T. Fermann, E. F. Valeev, Fundamentals of molecular integrals evaluation, arXiv preprint arXiv:2007.12057 (2020).

[36] T. Helgaker, P. R. Taylor, Gaussian basis sets and molecular integrals, in: Modern Electronic Structure Theory: Part II, 1995, pp. 725–856.

Appendix A. Hartree-Fock

Restricted Hartree-Fock is applicable to *closed-shell* systems where there is an even number of electrons that form spin-up and spin-down pairs. The unrestricted Hartree-Fock calculation, where spin is explicitly considered, is only slightly more complicated. Both make use of a Slater determinant state and are solved using self-consistent field iteration. By far the most common approach is to expand single electron states in terms of a basis set, see Appendix B,

$$\psi_i(\mathbf{r}) = \sum_{j=1}^{n} c_{ij}\phi_i(\mathbf{r}). \quad (A.1)$$

An *exact* HF calculation gives an upper bound on the ground state energy of the system, this is because it does not contain electron correlation. In practice exact HF energy is not achieved because the basis set is finite. In the restricted case, truncated expansion in terms of a basis set leads to *Roothan-Hall equations*

$$\mathbf{FC} = \mathbf{SC}\epsilon. \tag{A.2}$$

It is a nonlinear generalized eigenvalue problem, where the *Fock* matrix **F** depends on the coefficient matrix **C**. **S** is the *overlap* matrix, essentially containing information of how close the basis set is to orthogonality

$$S_{pq} = \int \phi_p(\mathbf{r})\phi_q(\mathbf{r})d\mathbf{r}, \tag{A.3}$$

and ϵ is a diagonal matrix of orbital energies. The density matrix **D** can be defined in terms of **C** as

$$D_{pq} = 2\sum_{a=1}^{\frac{N}{2}} c_{ap}c_{aq}, \tag{A.4}$$

from which the electron density is easily expressed

$$\rho(\mathbf{r}) = \sum_{pq} D_{pq}\phi_p(\mathbf{r})\phi_q(\mathbf{r}). \tag{A.5}$$

The Fock matrix is in the literature usually decomposed into

$$\mathbf{F} = \mathbf{H}_{\text{core}} + \mathbf{G}, \tag{A.6}$$

where $\mathbf{H}_{\text{core}}$ is the sum of the kinetic and potential contributions

$$\mathbf{H}_{\text{core}} = \mathbf{T} + \mathbf{V}, \tag{A.7}$$

defined in terms of basis functions as

$$T_{pq} = -\frac{1}{2}\int \phi_p(\mathbf{r})\nabla^2\phi_q(\mathbf{r})d\mathbf{r} \tag{A.8}$$

and

$$V_{pq} = \int \phi_p(\mathbf{r})\left[\sum_{A=1}^{M} V(\mathbf{r}, \{k_A\})\right]\phi_q(\mathbf{r})d\mathbf{r}. \tag{A.9}$$

G is the two-electron contribution to the Fock operator, defined as

$$G_{pq} = \sum_{rs} D_{rs}\left[(pq \mid rs) - \frac{1}{2}(pr \mid sq)\right] \tag{A.10}$$

where $(pq \mid rs)$ is the *two-electron* integral, in literature written in *chemists'* notation with round brackets

$$(pq \mid rs) = \int \frac{\phi_p(\mathbf{r}_1)\,\phi_q(\mathbf{r}_1)\,\phi_r(\mathbf{r}_2)\,\phi_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2. \tag{A.11}$$

In terms of quantities defined the energy of HF is

$$E = \text{Tr}\left(\mathbf{DH}_{\text{core}}\right) + \frac{1}{2}\text{Tr}(\mathbf{DG}). \tag{A.12}$$

Roothan-Hall equations are solved by iterating between two steps. In the first step, the Fock matrix is constructed from the density matrix (i.e. expansion coefficients), and in the second a new set of coefficients and consequently a new density matrix is obtained by solving the generalized eigenvalue problem. Repeating the two steps converges the system towards self-consistency.

Appendix B. Basis sets

Appendix B.1. Gaussian basis set

Cartesian Gaussian-type orbitals (GTOs) are one of the most popular basis sets in computational chemistry. They were introduced as an approximation to Slater-type orbitals (STOs), as they massively simplify evaluations of integrals in from the previous section. Each Slater orbital is approximated by an expansion in Gaussian primitives centered at $\mathbf{r}_A = (x_A, y_A, z_A)$,

$$G_{ijk}(\mathbf{r}, \alpha, \mathbf{A}) = (x - x_A)^i(y - y_A)^j(z - z_A)^k e^{-\alpha r_A^2}, \tag{B.1}$$

where i, j, k combine into the total angular-momentum quantum number as $l = i+j+k \geq 0$ and α is a variational parameter, usually determined with some least-square fit to relevant Slater orbitals. Gaussians of a given total angular-momentum l form a shell, the s-shell is G_{000}, p-shell are G_{100}, G_{010} and G_{001}, and so on for d, f, etc. shells.

A large body of scientific work exists with regards to efficient evaluation of Gaussian-type orbital integrals arising from Hartree-Fock. Most approaches [33, 34] boil down to recursive relations that simplify the kinetic energy T_{pq} and two-electron $(pq \mid rs)$ integrals into overlap integrals S_{pq}. The most problematic is the nuclear-electron contribution which is not separable in Cartesian directions and requires special treatment. Even though the implementation of the general GTOs basis set is straightforward it proved to be too much work within the time constraints of this written assignment, it is thus a clear direction for expansion of the code in this work.

The integrals simplify considerably if we only consider a single site at $\mathbf{r}_0 = (0, 0, 0)$, which is enough to treat Hookium. The overlap integral $S^{\alpha\beta}_{ijklmn} := S[G_{ijk}(\mathbf{r}, \alpha, \mathbf{0}), G_{lmn}(\mathbf{r}, \beta, \mathbf{0})]$ decomposes into separate components along Cartesian axes $S^{\alpha\beta}_{ijklmn} = S^{\alpha\beta}_{il}S^{\alpha\beta}_{jm}S^{\alpha\beta}_{kn}$

each with closed form solution

$$\begin{aligned} S_{il}^{\alpha\beta} &= \int_{-\infty}^{\infty} x^{i+l} e^{-\alpha x^2 - \beta x^2} \mathrm{d}x = \\ &= \frac{1}{2}\left(1 + (-1)^{i+l}\right) \sqrt{\frac{1}{(\alpha+\beta)^{1+i+j}}}\Gamma\left(\frac{1+i+l}{2}\right), \end{aligned} \tag{B.2}$$

where Γ is the gamma function. The integral vanishes for odd sums $i+l$. As in the general case, other integrals will be expressed in terms of overlap integrals for easier evaluation, starting with the nuclear attraction contribution $V_{ijklmn}^{\alpha\beta} = V_{il}^{\alpha\beta} V_{jm}^{\alpha\beta} V_{kn}^{\alpha\beta}$, for each Cartesian axis we get

$$V_{il}^{\alpha\beta} = k \int_{-\infty}^{\infty} x^i e^{-\alpha x^2} x^2 x^l e^{-\beta x^2} \mathrm{d}x = k S_{i+2l}^{\alpha\beta} = k S_{il+2}^{\alpha\beta}. \tag{B.3}$$

The kinetic energy matrix elements are simplified with use of the derivative identity

$$\frac{\partial}{\partial x} G_{ijk} = i G_{i-1jk} - 2\alpha G_{i+1jk}, \tag{B.4}$$

consequently

$$\frac{\partial^2}{\partial x^2} G_{ijk} = 4\alpha^2 G_{i+2jk} - 2\alpha(2i+1) G_{ijk} + i(i-1) G_{i-2jk}. \tag{B.5}$$

Again, Laplacian operator allows for decomposition into (xyz) components, $T_{ijklmn}^{\alpha\beta} = T_{il}^{\alpha\beta} T_{jm}^{\alpha\beta} T_{kn}^{\alpha\beta}$ each contributing

$$T_{il}^{\alpha\beta} = -\frac{1}{2} \int_{-\infty}^{\infty} x^i e^{-\alpha x^2} \frac{\partial^2}{\partial x^2} \left(x^l e^{-\beta x^2}\right) \mathrm{d}x, \tag{B.6}$$

expressed with overlap integrals

$$T_{il}^{\alpha\beta} = -2\beta^2 S_{il+2}^{\alpha\beta} + \beta(2l+1) S_{il}^{\alpha\beta} - \frac{i}{2}(i-1) S_{il-2}^{\alpha\beta}. \tag{B.7}$$

The two-electron integral is the most difficult, it cannot be expressed as a product of integrals for each axis. Closed form solutions for arbitrary angular momenta exist but include a 15-fold summation and are impractical for anything but $(ss|ss)$ integrals. In practice the expression is evaluated by using a recursive relation to break down high angular momentum integrals into only $(ss|ss)$-type contributions [35]. We use the Obara-Saika scheme, following notation from notation from [36]. First we recursively define Hermite Gaussians E_t^{ij} as

$$\begin{aligned} E_t^{ij} &= \frac{1}{2p} E_{t-1}^{i,j-1} + \frac{q Q_x}{\beta} E_t^{i,j-1} + (t+1) E_{t+1}^{i,j-1} \\ E_t^{ij} &= \frac{1}{2p} E_{t-1}^{i-1,j} - \frac{q Q_x}{\alpha} E_t^{i-1,j} + (t+1) E_{t+1}^{i-1,j} \\ E_0^{00} &= K_{AB} \\ E_t^{ij} &= 0 \quad \text{if} \quad t < 0, \quad \text{or} \quad t > i+j, \end{aligned} \tag{B.8}$$

where the auxiliary variables K_{AB}, Q_x, q, p and P_x are defined for two Gaussians with α and β at positions $\mathbf{r}_A, \mathbf{r}_B$ as

$$\begin{aligned} K_{AB} &= \exp\left(-q Q_x^2\right) \\ Q_x &= x_A - x_B \\ q &= \frac{\alpha\beta}{\alpha+\beta} \\ p &= \alpha + \beta \\ P_x &= \frac{1}{p}\left(\alpha A_x + \beta B_x\right). \end{aligned} \tag{B.9}$$

For our simplified case of coinciding only first terms in the recurrence relation (B.8) are nonzero and $K_{AB} = 1$. Additionally we need to define the auxiliary Hermite Coulomb integral $R_{tuv}^n(p, \mathbf{r}_P, \mathbf{R}_C)$ between Gaussian at $\mathbf{r}_P$ and Coulomb charge center at $\mathbf{R}_C$ as

$$\begin{aligned} R_{t+1,u,v}^n &= t R_{t-1,u,v}^{n+1} + (x_P - x_C) R_{t,u,v}^{n+1} \\ R_{t,u+1,v}^n &= u R_{t,u-1,v}^{n+1} + (y_P - y_C) R_{t,u,v}^{n+1} \\ R_{t,u,v+1}^n &= v R_{t,u,v-1}^{n+1} + (z_P - z_C) R_{t,u,v}^{n+1} \\ R_{0,0,0}^n &= (-2p)^n F_n\left(p R_{PC}^2\right). \end{aligned} \tag{B.10}$$

Again the relations simplify for our case, F_n is the Boys function, a special case of the incomplete Gamma function

$$F_n(x) = \int_0^1 \exp\left(-x t^2\right) t^{2n} \mathrm{d}t. \tag{B.11}$$

The two-body integral $(pq|rs)$ is evaluated as [36]

$$\begin{aligned} (pq|rs) = &\frac{2\pi^{5/2}}{pp'\sqrt{p+p'}} \sum_{tuv} E_{tuv}^{ab} \sum_{t'u'v'} (-1)^{t'+u'+v'} \\ &\times E_{t'u'v'}^{cd} R_{t+t',u+u',v+v'}\left(\alpha, \mathbf{P}, \mathbf{P'}\right), \end{aligned} \tag{B.12}$$

where $p = \alpha + \beta$ and $p' = \gamma + \delta$ are sums of Gaussian exponents.

Appendix B.2. Harmonic Oscillator basis set

The harmonic Oscillator basis set is composed of eigenfunctions of the Harmonic oscillator in three dimensions

$$\phi_k(r) = \frac{H_{2k-1}(r/\sqrt{2})}{2^k\sqrt{(2k-1)!}\,r/\sqrt{2}} \frac{\exp\left(-r^2/4\right)}{(2\pi)^{3/4}}. \tag{B.13}$$

This basis set is not nearly as commonly used as the GTOs, but is a good enough basis set for Hooke's law atom. The integrals required for Hartree-Fock are straightforward to compute and cheap to evaluate, especially with no angular dependence involved, one only needs to use Hermite recurrence relations

$$H_{n+1}(x) = 2xH_n(x) - H_n'(x), \tag{B.14}$$

where the derivatives satisfy

$$H_n'(x) = 2nH_{n-1}(x). \tag{B.15}$$

The overlap integral is the most straightforward, using orthogonality between the polynomials

$$\int_{-\infty}^{\infty} H_m(x)H_n(x)e^{-x^2}\, dx = \sqrt{\pi}2^n n! \delta_{nm}, \tag{B.16}$$

we obtain

$$S_{kl} = \int_0^{\infty} \phi_k(r)\phi_l(r)4\pi r^2 \mathrm{d}r = \delta_{kl}. \tag{B.17}$$

The potential matrix element is not much more difficult to compute

$$\begin{aligned} V_{kl} = k\int_0^{\infty} \phi_k(r)r^2\phi_l(r)4\pi r^2 \mathrm{d}r &= \\ = (4k-1)\delta_{kl} + \delta_{kl+1}\sqrt{4k^2-6k+2} & \\ + \delta_{kl-1}\sqrt{4k^2+2k}, & \end{aligned} \tag{B.18}$$

because one uses the recurrence relations once V_{kl} is tridiagonal instead of diagonal. The kinetic integral is a bit more arduous to evaluate and involves several applications of aforementioned recurrence and derivative expressions

$$\begin{aligned} T_{kl} = -\frac{1}{2}\int_0^{\infty} \phi_k(r)\left(\nabla^2\phi_l(r)\right)4\pi r^2 \mathrm{d}r &= \\ = \frac{(3+4k)}{8}\delta_{kl} - \frac{1}{4}\delta_{kl+1}\sqrt{\frac{1}{2}(1-3n+2n^2)} & \\ - \frac{1}{4}\delta_{kl-1}\sqrt{\frac{1}{2}(n+2n^2)}. & \end{aligned} \tag{B.19}$$

The two-electon integral is the toughest to perform, depending on the type of basis function it is solved by using either a multipole expansion for the $\propto \frac{1}{r_{12}}$ term, representing the term using a Laplace transform, or performing the integral in Fourier space. The latter approach works in the case of harmonic-oscillator basis, because the Fourier transform of a product of two basis functions ϕ_k, ϕ_j is an even polynomial multiplied by an exponential factor.