FYS4411 - Computational quantum mechanics

Spring 2016

Project 1;
Variational Monte Carlo Studies of Bosonic systems

temporary report

Sean Bruce Sangolt Miller
s.b.s.miller@fys.uio.no

Filip Henrik Lasren
filiphenriklarsen@gmail.com

Date: March 8, 2016

Abstract

Some text that is abstact

Contents

1 Introduction **1**

2 Theory and methods **1**

2.1 Preliminary derivations . . . 1

2.1.1 Simplified problem . . . 1

2.1.2 Full problem . . . 2

2.2 The method of steepest descent . . . 4

3 Results **5**

3.1 Benchmarks . . . 5

3.1.1 Standard Metropolis sampling . . . 5

3.1.2 Importance sampling . . . 5

4 Conclusions **6**

5 Appendix **6**

1 Introduction

2 Theory and methods

2.1 Preliminary derivations

2.1.1 Simplified problem

The local energy is defined as:

$$E_L(\mathbf{R}) = \frac{1}{\Psi_T(\mathbf{R})} H\Psi_T(\mathbf{R}), \tag{1}$$

As a first approximation, it is assumed there is no interaction term in the Hamiltonian, which means the hard sphere bosons have no physical size (the hard-core diameter is zero). It is also assumed that no magnetic field is applied to the bosonic gas, leaving a perfectly spherically symmetrical harmonic trap. Inserting this new Hamiltonian into the local energy gives:

$$E_L(\mathbf{R}) = \frac{1}{\Psi_T(\mathbf{R})} \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\omega_{ho}^2 r_i^2 \right) \Psi_T(\mathbf{R}) \tag{2}$$

The potential term is trivial since this is a scalar, i.e. the denominator will cancel the wavefunction. A more challenging problem is to find an expression for $\nabla_i^2\Psi_T(\mathbf{R})$. The trial wavefunction shown in equation (...), with the aforementioned approximations, is:

$$\Psi_T(\mathbf{R}) = \prod_i e^{-\alpha r_i^2} \tag{3}$$

where α is the variational parameter for VCM. The first derivative is:

$$\nabla_j \prod_i e^{-\alpha r_i^2} = -2\alpha\mathbf{r}_j e^{-\alpha r_j^2} \prod_{i\neq j} e^{-\alpha r_i^2} \tag{4}$$

$$= -2\alpha\mathbf{r}_j \prod_i e^{-\alpha r_i^2}. \tag{5}$$

The second derivative then follows:

$$\nabla_j^2 \prod_i e^{-\alpha r_i^2} = \nabla_j \left(-2\alpha\mathbf{r}_j \prod_i e^{-\alpha r_i^2} \right) \tag{6}$$

$$= \left(4\alpha^2 r_j^2 - 2d\alpha\right) \prod_i e^{-\alpha r_i^2}. \tag{7}$$

where d is the number of dimensions. Inserting this into back into the local energy (equation (2)), the final expression can be derived:

$$\begin{aligned} E_L(\mathbf{R}) &= \frac{1}{\Psi_T(\mathbf{R})} \sum_i^N \left(\frac{-\hbar^2}{2m}\nabla_i^2 + \frac{1}{2}m\omega_{ho}^2 r^2 \right) \Psi_T(\mathbf{R}) \\ &= \sum_{i=1}^N \left[\frac{-\hbar^2}{2m}\left(4\alpha^2 r_i^2 - 2d\alpha\right) + \frac{1}{2}m\omega_{ho}^2 r_i^2 \right] \end{aligned}$$

The drift force (quantum force), still with the approximations above, is defined by:

$$F = \frac{2\nabla\Psi_T}{\Psi_T} \tag{8}$$

The gradient here is defined as

$$\nabla \equiv (\nabla_1, \nabla_2, \ldots, \nabla_N)$$

i.e. a vector of dimension Nd. The gradient with respect to a single particle's position is already given in equation 5, so it's not too hard to realise:

$$
\begin{aligned}
F &= \frac{-4\alpha}{\Psi_T}\left(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\right)\Psi_T \\
&= -4\alpha\left(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\right)
\end{aligned}
$$

2.1.2 Full problem

The full local energy[1] is a bit more tedious to derive. The first step is to rewrite the trial wavefunction to the following form:

$$
\Psi_T(\mathbf{R}) = \prod_i \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})} \tag{9}
$$

where, in order for this to fit with the previous wavefunction, $u(r_{ij}) \equiv \ln[f(r_{ij})]$ and $\phi(\mathbf{r}_i) \equiv g(\alpha, \beta, \mathbf{r}_i)$. The gradient with respect to the k-th coordinate set is:

$$
\begin{aligned}
\nabla_k \Psi_T &= \nabla_k \prod_i \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})} \qquad (10) \\
&= \nabla_k \phi_k \left[\prod_{i \neq k} \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})}\right] + \left[\prod_i \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})} \nabla_k \left(\sum_{i''<j''} u_{i''j''}\right)\right] \qquad (11) \\
&= \nabla_k \phi_k \left[\prod_{i \neq k} \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})}\right] + \left[\prod_i \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})} \left(\sum_{i''<j''} \nabla_k u_{i''j''}\right)\right] \qquad (12)
\end{aligned}
$$

The function u_{ij} is symmetric under permutation $i \leftrightarrow j$, as one can see from the definitions of u_{ij} and $f(r_{ij})$. Therefore, the last sum above can have a different indexing: All terms without an index k, will give zero when taking the derivative ∇_k, so only $i = k$ or $j = k$ remains (remember $i \neq j$). Due to the symmetry of u_{ij}, one can simply always say $i = k$ and let j be the summation index:

$$
\nabla_k \Psi_T = \nabla_k \phi_k \left[\prod_{i \neq k} \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})}\right] + \left[\prod_i \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})} \left(\sum_{j'' \neq k} \nabla_k u_{kj''}\right)\right] \tag{13}
$$

The second derivative now becomes (where ∇_k only acts on the first parenthesis to its right):

$$
\begin{aligned}
\nabla_k^2 \Psi_T = (\nabla_k^2 \phi_k) &\left[\prod_{i \neq k} \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})}\right] + (\nabla_k \phi_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \nabla_k e^{\sum_{i'<j'} u(r_{i'j'})}\right] \\
&+ \left[\nabla_k \left(\prod_i \left(\phi(\mathbf{r}_i)\right) e^{\sum_{i'<j'} u(r_{i'j'})}\right) \left(\sum_{j'' \neq k} \nabla_k u_{kj''}\right)\right] \\
&+ \left[\prod_i \phi(\mathbf{r}_i) e^{\sum_{i'<j'} u(r_{i'j'})} \left(\sum_{j'' \neq k} \nabla_k^2 u_{kj''}\right)\right]
\end{aligned}
$$

While a bit of a nuisance to read, the expression above is simply the product rule for $\nabla_k(\nabla_k \Psi_T)$. Written in terms of Ψ_T, the above can be a bit simplified[2]:

$$
\begin{aligned}
\nabla_k^2 \Psi_T = (\nabla_k^2 \phi_k) \frac{\Psi_T}{\phi(\mathbf{r}_k)} &+ (\nabla_k \phi_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \nabla_k e^{\sum_{i'<j'} u(r_{i'j'})}\right] \\
&+ \left[(\nabla_k \Psi_T) \left(\sum_{j'' \neq k} \nabla_k u_{kj''}\right)\right] \\
&+ \left[\Psi_T \left(\sum_{j'' \neq k} \nabla_k^2 u_{kj''}\right)\right]
\end{aligned}
$$

[1] The "full local energy" means not making any assumptions on the particle interactions or the potential.

[2] For the more mathematically concerned nitpicker, the product symbol only runs over the functions $\phi(\mathbf{r}_i)$), not the following exponential. This is easy to realise by recalling how Ψ_T was defined, and is important to know when inserting Ψ_T as done now.

The second term above is equal to the second term in equation 13, divided by $\phi(\mathbf{r}_k)$. Furthermore, the gradient $\nabla_k \Psi_T$ is already calculated above, and in terms of Ψ_T is:

$$\nabla_k \Psi_T = \frac{\Psi_T}{\phi(\mathbf{r}_k)} \nabla_k \phi_k + \Psi_T \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \tag{14}$$

Inserting all this back into the second derivative yields:

$$\begin{aligned}
\nabla_k^2 \Psi_T = {} & (\nabla_k^2 \phi_k) \frac{\Psi_T}{\phi(\mathbf{r}_k)} + (\nabla_k \phi_k) \left[\frac{\Psi_T}{\phi(\mathbf{r}_k)} \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \right] \\
& + \left[\frac{\Psi_T}{\phi(\mathbf{r}_k)} \nabla_k \phi_k + \Psi_T \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \right] \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \\
& + \left[\Psi_T \left(\sum_{j'' \neq k} \nabla_k^2 u_{kj''} \right) \right]
\end{aligned}$$

Giving:

$$\begin{aligned}
\frac{1}{\Psi_T} \nabla_k^2 \Psi_T = {} & (\nabla_k^2 \phi_k) \frac{1}{\phi(\mathbf{r}_k)} + (\nabla_k \phi_k) \left[\frac{1}{\phi(\mathbf{r}_k)} \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \right] \\
& + \left[\frac{1}{\phi(\mathbf{r}_k)} \nabla_k \phi_k + \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \right] \left(\sum_{j'' \neq k} \nabla_k u_{kj''} \right) \\
& + \sum_{j'' \neq k} \nabla_k^2 u_{kj''}
\end{aligned}$$

Which can be rewritten to:

$$\frac{1}{\Psi_T} \nabla_k^2 \Psi_T = \frac{\nabla_k^2 \phi_k}{\phi(\mathbf{r}_k)} + \frac{2 \nabla_k \phi_k}{\phi(\mathbf{r}_k)} \left(\sum_{i \neq k} \nabla_k u_{ki} \right) + \left(\sum_{j \neq k} \nabla_k u_{kj} \right)^2 + \sum_{l \neq k} \nabla_k^2 u_{kl} \tag{15}$$

The gradients $\nabla_k u_{ki}$ can be rewritten using partial differentiation (where $r_{k,i}$ is coordinate i of $\mathbf{r}_k$):

$$\begin{aligned}
\nabla_k u_{ki} &= \left(\frac{\partial}{\partial r_{k,1}}, \frac{\partial}{\partial r_{k,2}}, \ldots \right) u_{ki} \\
&= \frac{\partial u_{ki}}{\partial r_{ki}} \left(\frac{\partial r_{ki}}{\partial r_{k,1}}, \frac{\partial r_{ki}}{\partial r_{k,2}}, \ldots \right) \\
&= \frac{\partial u_{ki}}{\partial r_{ki}} \left(\frac{\partial}{\partial r_{k,1}} \left(\sqrt{[(r_{k,1} - r_{i,1}) \hat{e}_1 + \ldots]^2} \right), \ldots \right) \\
&= \frac{\partial u_{ki}}{\partial r_{ki}} \left(\frac{r_{k,1} - r_{i,1}}{r_{ki}}, \ldots \right) \\
&= \frac{\partial u_{ki}}{\partial r_{ki}} \frac{\mathbf{r}_k - \mathbf{r}_i}{r_{ki}} \\
&= \frac{\mathbf{r}_k - \mathbf{r}_i}{r_{ki}} u'_{ki} \ , \ \ u'_{ki} \equiv \frac{\partial u_{ki}}{\partial r_{ki}}
\end{aligned}$$

While the second derivative of u_{ki} is:

$$
\begin{aligned}
\nabla_k^2 u_{ki} &= \nabla_k(\nabla_k u_{ki}) \\
&= u'_{ki}\nabla_k\left(\frac{\mathbf{r}_k-\mathbf{r}_i}{r_{ki}}\right) + \frac{\mathbf{r}_k-\mathbf{r}_i}{r_{ki}}\nabla_k u'_{ki} \\
&= u'_{ki}\left(\frac{d}{r_{ki}} - \frac{r_{k,1}-r_{i,1}}{r_{ki}^3} - \frac{r_{k,2}-r_{i,2}}{r_{ki}^3} - \ldots\right) + \frac{\mathbf{r}_k-\mathbf{r}_i}{r_{ki}}\cdot\frac{\mathbf{r}_k-\mathbf{r}_i}{r_{ki}}u''_{ki} \\
&= \frac{u'_{ki}}{r_{ki}}\left(d - \frac{r_{k,1}-r_{i,1}}{r_{ki}^2} - \frac{r_{k,2}-r_{i,2}}{r_{ki}^2} - \ldots\right) + u_{ki} \\
&= \frac{u'_{ki}}{r_{ki}}(d-1) + u_{ki}
\end{aligned}
$$

where d, as earlier, is the number of dimensions present, which in our world is usually $d = 3$. Finally, this gives:

$$
\frac{1}{\Psi_T}\nabla_k^2\Psi_T = \frac{\nabla_k^2\phi_k}{\phi(\mathbf{r}_k)} + \frac{2\nabla_k\phi_k}{\phi(\mathbf{r}_k)}\left(\sum_{i\neq k}\frac{\mathbf{r}_k-\mathbf{r}_i}{r_{ki}}u'_{ki}\right) + \sum_{i,j\neq k}\frac{(\mathbf{r}_k-\mathbf{r}_i)(\mathbf{r}_k-\mathbf{r}_j)}{r_{ki}r_{kj}}u'_{ki}u'_{kj} + \sum_{l\neq k}u_{ki} + \frac{u'_{ki}}{r_{ki}}(d-1) \tag{16}
$$

As the exact forms of ϕ_k and u_{ki} are known, this can be written to a more recognisable, and calculable, expression. However, this expression will be quite long, so only the necessary variables will be derived. Inserting them into equation 16 is trivial. The u_{ki} derivatives are:

$$
\frac{\partial u_{ki}}{\partial r_{ki}} = \frac{\partial}{\partial r_{ki}}\left(\ln\left[1-\frac{a}{r_{ki}}\right]\right) = \frac{a}{r_{ij}^2 - ar_{ij}} \tag{17}
$$

$$
\frac{\partial^2 u_{ki}}{\partial r_{ki}^2} = \frac{\partial}{\partial r_{ki}}u'_{ki} = -a\frac{2r_{ki}-a}{r_{ki}^2 - ar_{ki}} \tag{18}
$$

and the ϕ_k derivatives are:

$$
\nabla_k\phi_k = \nabla_k e^{-\alpha(x_k^2+y_k^2+\beta z_k^2)} = -2\alpha(x_k, y_k, \beta z_k)\phi_k \tag{19}
$$

$$
\nabla_k^2\phi_k = \nabla_k(\nabla_k\phi_k) = \left[-2\alpha(2+\beta) + 4\alpha^2(x_k^2+y_k^2+\beta^2 z_k^2)\right]\phi_k \tag{20}
$$

2.2 The method of steepest descent

In order to find the value for α that minimizes $\langle E_L\rangle$, the method of steepest descent (SD) is applied. The SD method, in algorithm form, is:

$$
x_{n+1} = x_n - \gamma_n\nabla f \tag{21}
$$

where x is the variable with which one wishes to find the minimum of f. In application to the current problem, $f = \langle E_L\rangle$. However, since $\langle E_L\rangle$ is an expensive quantity to find numerically, and its derivative ($\bar{E}_\alpha \equiv \frac{d\langle E_L\rangle}{d\alpha}$) even more so, an analytical expression is desirable. This can be found as follows:

$$
\begin{aligned}
\bar{E}_\alpha &= \frac{d}{d\alpha}\int dx P(x)E_L \\
&= \frac{d}{d\alpha}\int dx\frac{|\psi|^2}{\int dx'|\psi|^2}\frac{1}{\psi}H\psi \\
&= \frac{d}{d\alpha}\int dx\frac{\psi^* H\psi}{\int dx'|\psi|^2}
\end{aligned} \tag{22}
$$

Since the Hamiltonian is hermitian, one has $\int dx\psi^* H\psi = \int dx H\psi^*\psi$, giving:

$$
\begin{aligned}
&= \frac{d}{d\alpha}\int dx\frac{H\psi^*\psi}{\int dx'|\psi|^2} \\
&= \left[\int dx\frac{H\left(\psi^*\left(\frac{d\psi}{d\alpha}\right) + \left(\frac{d\psi^*}{d\alpha}\right)\psi\right)}{\int dx'|\psi|^2}\right] - \left[\int dx\frac{H\psi^*\psi}{\left(\int dx'|\psi|^2\right)^2}\int dx'\left(\psi^*\left(\frac{d\psi}{d\alpha}\right) + \left(\frac{d\psi^*}{d\alpha}\psi\right)\right)\right]
\end{aligned} \tag{23}
$$

Again one may use the hermiticity of the Hamiltonian to get $\int dx H\psi^* \left(\frac{d\psi}{d\alpha}\right) = \int dx H \left(\frac{d\psi^*}{d\alpha}\right)\psi$. So:

$$\begin{aligned}
&= 2\left[\int dx \frac{H\psi^* \frac{d\psi}{d\alpha}}{\int dx'|\psi|^2}\right] - 2\left[\int dx \frac{H\psi^*\psi}{\left(\int dx'|\psi|^2\right)^2}\int dx'\psi^*\frac{d\psi}{d\alpha}\right] \\
&= 2\left[\int dx \frac{H\psi^* \frac{d\psi}{d\alpha}}{\int dx'|\psi|^2} - \int dx \frac{H\psi^*\psi}{\int dx'|\psi|^2}\int dx' \frac{1}{\int dx'|\psi|^2}\psi^*\frac{d\psi}{d\alpha}\right] \\
&= 2\left[\int dx \frac{\psi^*\left(\frac{E_L}{\psi}\frac{d\psi}{d\alpha}\right)\psi}{\int dx'|\psi|^2} - \int dx \frac{\psi^* E_L \psi}{\int dx'|\psi|^2}\int dx' \frac{\psi^*\left(\frac{1}{\psi}\frac{d\psi}{d\alpha}\right)\psi}{\int dx'|\psi|^2}\right] \\
&= 2\left(\langle\frac{\bar{\psi}_\alpha}{\psi}E_L\rangle - \langle\frac{\bar{\psi}_\alpha}{\psi}\rangle\langle E_L\rangle\right)
\end{aligned} \tag{24}$$

where $\bar{\psi}_\alpha \equiv \frac{d\psi}{d\alpha}$. This is a much better expression to use since one only need one Monte Carlo cycle to find $\bar{E}_\alpha$. Therefore, one Monte Carlo cycle will give one value for $\bar{E}_L$.
Since α will only have to be determined once, it is permissible to do so with greater accuracy. This means ... (explain about higher accuracy and divisions by 2)

3 Results

3.1 Benchmarks

As in most computational work, we need to make certain our program produce reliable values. A very useful way to do this is to compare our numerical results with a known analytical solution to the problem. Since we do know the analytical solution to the spherical harmonic oscillator, these may be used as benchmarks that we hope to reproduce. We have already derived an analytical expression in (**??**).

3.1.1 Standard Metropolis sampling

Tests were done with several choices of number of particles, and number of dimensions. The outputs were the expectation value of the energy, the standard deviation of this value and the time elapsed.

At first we chose to calculate the laplacian in the kinetic term numerically. The results are shown in table(blah). As can be seen the analytical results are well reproduce. The standard deviation is not zero, but this is due to numerical error in the calculation of the kinetic energy. This error increases with the number of particles and number of dimensions.

Once convinced the local energies were reproduced correctly, we implemented an analytical calculation of the laplacian as well. A user may choose to use this option by setting

```
system->setAnalyticalLaplacian (true);
```

in `main.cpp`. The tests using analytical calculation of the laplacian resulted as shown in table(blah2).

3.1.2 Importance sampling

Replacing the standard metropolis sampling with importance sampling, we allow a so called quantum force to affect the probability of accepting a move. This force works as a drift toward the most probable state. This should effectively, as the name suggests, allow us to sample more important areas and thereby not discard as many moves. In the same fashion as the toggle for use of analytical calculation of the laplacian, one can choose to use importance sampling by setting

```
system->setImportanceSampling (true);
```

in `main.cpp`. Results using importance sampling are shown in table(blah3).

Table 1: Numerically calculated laplacian.

N	D	Analytical	Numerical	Variance	Time [s]
1	1	0.5	0.5	3.6e-16	0.109
1	2	1	1	2.6e-15	0.120
1	3	1.5	1.5	1.6e-14	0.135
10	1	5	5	1.3e-13	0.498
10	2	10	10	1.7e-13	0.706
10	3	15	15	4.3e-13	1.07
100	1	50	50	1.1e-10	8.55
100	2	100	100	2.4e-09	20.9
100	3	150	150	6.8e-09	39.5
500	1	250	250	3.4e-08	118
500	2	500	500	3.9e-07	400
500	3	750	750	3.0e-06	834

Table 2: Analytically calculated laplacian.

N	D	Analytical	Numerical	Variance	Time [s]
1	1	0.5	0.5	0	0.09
1	2	1	1	0	0.10
1	3	1.5	1.5	0	0.13
10	1	5	5	0	0.33
10	2	10	10	0	0.37
10	3	15	15	0	0.38
100	1	50	50	0	2.92
100	2	100	100	0	2.90
100	3	150	150	0	2.95
500	1	250	250	0	14.2
500	2	500	500	0	14.5
500	3	750	750	0	15.2

Table 3: some caption

N	D	Analytical	Numerical	Variance	Time
1	1				
1	2				
1	3				
10	1				
10	2				
10	3				
100	1				
100	2				
100	3				
500	1				
500	2				
500	3				

4 Conclusions

5 Appendix