11. b. Quantum Monte Carlo

[Gould-Tobochnik ch. 18; for basic quantum mechanics see e.g. Eisberg-Resnick p. 210 - 220 and appendix H, or Gasiorowicz. Also David Ceperleys QMC notes, material/ceperley.ps]

11.1. Introduction

Quantum MC means methods where Monte Carlo approaches are used to study quantum mechanical systems. This is a large area of research, and on this course I will only touch the surface by first going through a very simple example in detail, then mentioning the basic principles of a few other methods.

11.1.1. Reminder of the basics of quantum physics

In quantum physics particles are not considered point-like, but instead have wave-like properties. In 1+1 dimensions (x, t) a particle is described by a wave function

$$\Psi(x,t) \tag{1}$$

from which the probability that the particle is in a given region of space at a given time $P(x, t)$ is obtained as

$$P(x,t) = |\Psi(x,t)|^2 dx \tag{2}$$

which requires that Ψ is normalized as

$$\int_{-\infty}^{\infty} \Psi^*(x,t)\Psi(x,t)dx = 1. \tag{3}$$

In the non-relativistic limit the development of the system is determined by the time-dependent Schrödinger equation

$$-\frac{\hbar^2}{2m}\frac{\partial^2\Psi(x,t)}{\partial x^2}+V(x,t)\Psi(x,t)=i\hbar\frac{\partial\Psi(x,t)}{\partial t} \tag{4}$$

where V is some potential energy function. If V is independent of time, one can separate the position and time dependence as

$$\Psi(x,t)=\psi(x)e^{-iEt/\hbar} \tag{5}$$

and by insertion into the time-dependent equation 4 obtain the *time-independent* Schrödinger equation

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x)}{\partial x^2}+V(x)\psi(x)=E\psi(x) \tag{6}$$

By defining the Hamiltonian operator

$$H_{\mathrm{op}}=-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}+V(x,t) \tag{7}$$

this can also be written as

$$H_{\mathrm{op}}\psi(x)=E\psi(x) \tag{8}$$

$\psi(x)$ is an *eigenfunction* and E an *eigenvalue* of H_{op}. Often this equation has several different solutions with energies $E_n, n=1,\ldots$

11.2. Variational Quantum MC (VMC)

One of the most important uses of the time-independent Schrödinger equation in atomic, molecular and materials physics is in determining the ground state energy solution E_0 of the Schrödinger equation.

- This is because the Born-Oppenheimer approximation usually holds well in these systems, i.e. the electrons have time to relax to their ground state for every given position of the nuclei.
- This means that the E_0 solution will give the normal energy state for e.g. a single molecule.

The **variational principle** is very useful for determining E_0. The principle states that for any Hamiltonian $H_{\rm op}$ and any arbitrary trial wave function ψ

$$E[\psi] = \frac{\int \psi^*(x) H_{\rm op} \psi(x) dx}{\int \psi^*(x)\psi(x) dx} \geq E_0 \tag{9}$$

(for a proof, see e.g. Gasiorowicz p. 292-293).

This means that if the potential V in the system is known, then one can in principle start with any arbitrarily bad guess of $\psi^{\rm trial}$ which depends on a set of variable parameters v_i. If the functional form is flexible enough, then by minimizing $E[\psi]$ by optimizing the v_i one should eventually arrive at the correct $\psi(x)$ and E_0.

Doing such optimization is a task well suited for computers.

- As an ultimate case one can think of making the trial function $\psi^{\rm trial}$ fully numerical, i.e. letting the parameters v_i just be a set of points which describe the function by numerical interpolation.

- Then the function indeed is fully flexible, and by using enough number crunching one should in principle be able to solve the Schrödinger equation for any V.

  - This approach actually can be used for linear molecules [Leif Laaksonen, PhD thesis].

The optimization of the v_i can be carried out in many different ways, and in simple cases obviously ordinary numerical optimization methods are likely to be the most efficient ones. But on this MC course we will not deal with these.

The subject of this course, Monte Carlo methods, can actually be useful in at least two respects for solving eq. 9:

1. Monte Carlo methods such as simulated annealing may be used to carry out the minimization of $E[\psi]$.

2. Monte Carlo integration can be used to carry out the integrals in eq. 9.

Although we have now written everything in one dimension, in practice most systems studied are highly multidimensional.

- E.g. if we study just a 10-atom molecule with 6 electrons per atom, there are 70 particles of interest.
- Even if every particle could be described with a single wave function with only three coordinate parameters (x, y, z) we would have a 210-dimensional system at hand.
- As we know, for such high dimensions MC integration is the most efficient way to carry out numerical integration.

Moreover, by a slight rewriting of eq. 9 one can also use an MCMC method to carry out the integral. We consider bound states, for which $\psi^* = \psi$, and write

$$E[\psi] = \int \psi^2(x) E_L(x) dx \tag{10}$$

where

$$E_L = \frac{H_{\rm op}\psi(x)}{\psi(x)}. \tag{11}$$

Now the upper equation is a weighted average with the weight being

$$\frac{\psi^2(x)}{\int \psi^2(x) dx} \tag{12}$$

One can use an MCMC method to generate points x distributed as $\psi^2(x)$ so that the MC estimate of E is

$$E[\psi] = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E_L(x_i) \tag{13}$$

The algorithm for this is

```
0°  Select a trial change size $\delta$
1°  Choose a trial position $x_{\mathrm{trial}} = x_i + (1 - 2u)\delta$ where $u$ is a random number between 0 and 1
2°  Calculate $w = p(x_{\mathrm{trial}})/p(x_i)$, where in this case $p(x) = \psi^2(x)$
3°  If $w \geq 1$ accept the change
4°  If $w < 1$ :
    4 a°  Generate a random number $u$ between 0 and 1
    4 b°  Accept the state only if $u \leq w$
5°  If the trial was accepted let $x_{n+1} = x_{\mathrm{trial}}$. Else let $x_{n+1} = x_i$
6°  Return to step 1
```

This gives a sequence of points x_i which can be used to calculate the sum in equation 13.

- Remember that, as usual in MCMC, it takes a while for the system to equilibrate and reach the correct distribution of states. Hence one needs to follow the equilibration before beginning to collect statistics.

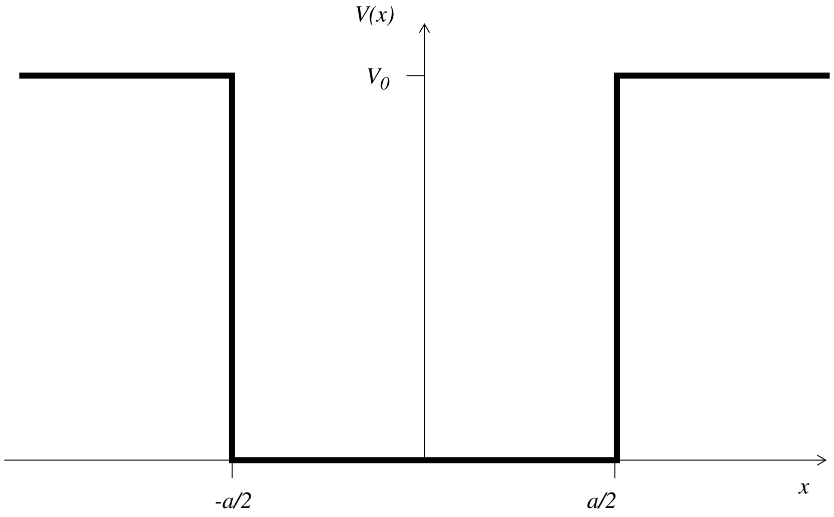
It is perfectly possible to combine steps **1.** and **2,** i.e. both use MC methods for optimizing the wave function, and then MC integration to determine the $E[\psi]$ for each candidate wave function.

11.2.1. The finite and infinite square well potentials

To give some concrete idea of how quantum MC may work, I'll use a very simple quantum mechanical model with a known analytical solution.

11.2.1.1. Analytical solution [Eisberg-Resnick sec. 6-7]

Consider the finite square well potential:



This potential is the simplest possible potential which gives bound state solutions to the Schrödinger equation.

- The surprising thing is it is not entirely preposterous physically.
  - E.g. a neutron bound inside a nucleus, or an free electron in a thin conducting film surrounded by insulator, will experience a potential energy function $V(r)$ which is actually quite close to a sphere/square well, respectively.

This potential can mathematically be written as

$$V(x) = \begin{cases} V_0 & \text{when } |x| > a/2 \\ 0 & \text{when } |x| \leq a/2 \end{cases} \tag{14}$$

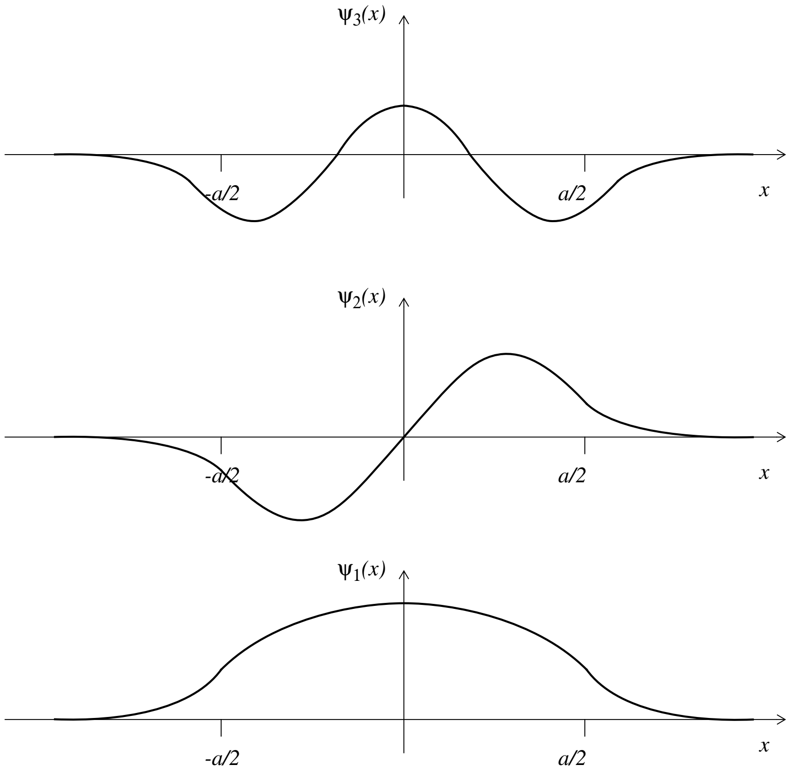
If the energy of the particle $E < V_0$ this gives bound state solutions with quantized energy levels. In the well region the solutions are of the form for standing waves and can in general be written as

$$\psi(x) = A\sin(kx) + B\cos(kx) \quad \text{when } |x| < a/2 \tag{15}$$

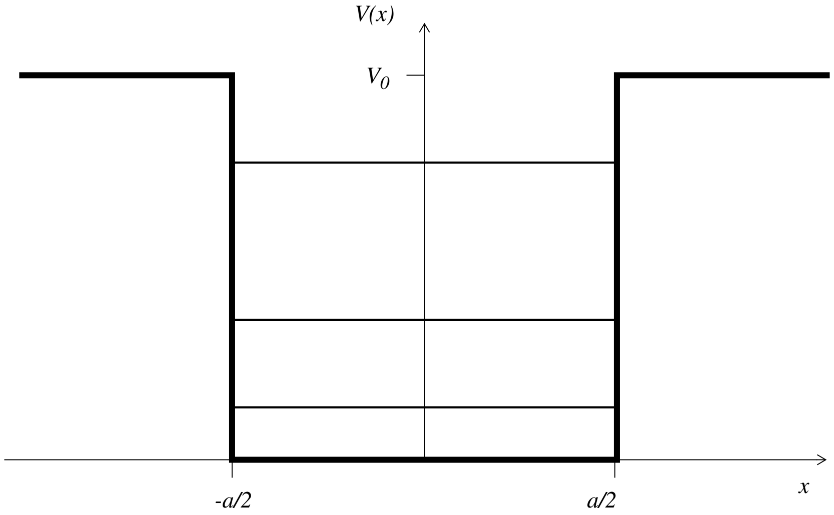
Outside the well region the solutions must vanish with increasing $|x|$ and are hence of the form

$$\psi(x) = \begin{cases} Ce^{kx} & \text{when } x < -a/2 \\ Ge^{-kx} & \text{when } x > a/2 \end{cases} \tag{16}$$

The values of the 4 constants A, B, C and G can be determined by requiring that both $\psi(x)$ and its first derivative are continuous at $\pm a/2$. One then obtains solutions of the type



with a few quantized energy levels like



Unfortunately the solution leads to a transcendental equation which is not analytically solvable. Here I will quote a sample numerical solution which will be used later. If m, V_0 and a are related such that

$$\sqrt{\frac{mV_0a^2}{2\hbar^2}} = 4 \tag{17}$$

then the ground state solution will be approximately

$$E_0 = 0.0980 V_0. \tag{18}$$

Now consider the infinite square well potential, i.e. the limit $V_0 \to \infty$. Then states outside the box will be completely forbidden, and we are left with the solution inside the box

$$\psi(x) = A\sin(kx) + B\cos(kx) \tag{19}$$

By requiring the physically reasonable boundary condition

$$\psi(x) = 0 \quad \text{when } x = \pm a/2 \tag{20}$$

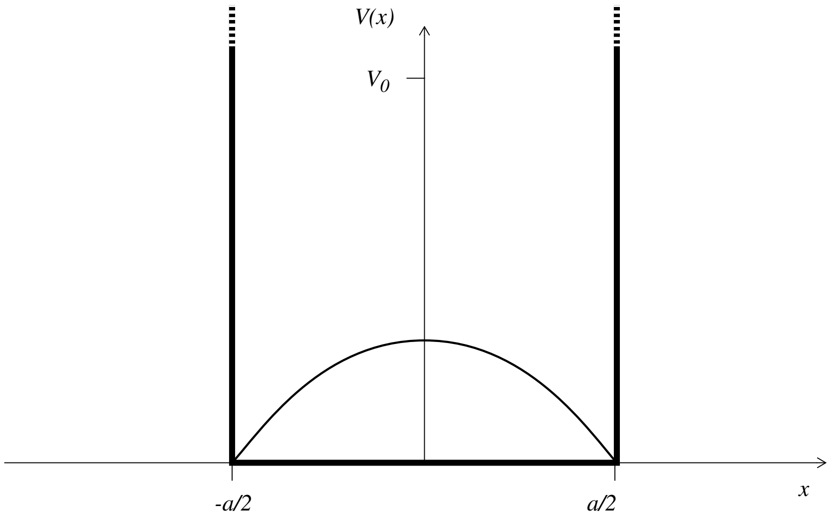
one can easily show that the exact solutions of the Schrödinger equation are simple sine and cosine functions, and the lowest of these gives

$$\psi_0(x) = \sqrt{\frac{2}{a}}\cos\frac{\pi x}{a} \tag{21}$$

which has the ground state energy

$$E_0 = \frac{\pi^2\hbar^2}{2ma^2} \tag{22}$$

This solution is illustrated here:



11.2.1.2. MC simulation solution

To get a concrete idea of how variational MC may work, I will use the simplest possible bound state solution of the Schrödinger equation, the infinite square well, and show how simulated annealing can be used to carry out the optimization of

$$E[\psi] = \frac{\int \psi^*(x) H_{\rm op} \psi(x) dx}{\int \psi^*(x)\psi(x) dx} \geq E_0 \tag{23}$$

For bound states this can be simplified using $\psi^* = \psi$.

For simplicity, I will just use direct numerical integration to carry out the integrals, and focus on testing whether the variational optimization can be used to find the correct ground state energy. The optimization scheme is ordinary simulated annealing.

The basic idea of the code is (u is always a new random number in the interval $[0, 1[$):

0 a° Select a trial wave function ψ with $n_{\rm param}$ parameters to be varied $\psi(x,(v_i, i = 1 \ldots n_{\rm param}))$.

0 b° Select a variation interval Δv

0 c° Select an initial temperature T

1° Select one of the parameters v_i using $i = {\rm int}(u * n_{\rm param})$

2° Calculate $E_b = \dfrac{\int \psi(x,\bar{v}) H_{\rm op} \psi(x,\bar{v}) dx}{\int \psi(x,\bar{v}) \psi(x,\bar{v}) dx}$

3° Get a trial $v_i^{\rm trial} = v_i(1 - \Delta v/2 + u \Delta v)$

4° Calculate $E_a = \dfrac{\int \psi(x,\bar{v}) H_{\rm op} \psi(x,\bar{v}) dx}{\int \psi(x,\bar{v}) \psi(x,\bar{v}) dx}$

5° Obtain $\Delta E = E_a - E_b$

- **6°** If $\Delta E \leq 0$ accept the state
- **7°** If $\Delta E > 0$ accept the state only if $u < e^{-\Delta E/T}$

8° If the state is rejected, return to the previous state: $v_i = v_i^{\rm before}$

9° Lower the temperature T a bit

10° Return to step 1

A code implementation of this looks like follows (main routine). Since a computer can not handle infinities directly, this is actually for a finite square well, and the idea is that the user keeps increasing V_0 towards infinity.

```
!
! Program to solve ground state energy E of Schrödinger equation in
! 1D using variational principle
!
! Potential is the square well potential, of height V0 and interval -a/2 to a/2
! Working interval is -5a/2 to 5a/2.
! See any quantum physics textbook on this, e.g. Eisberg-Resnick ch. 6.7
!
!

module constants

  double precision, parameter :: hbar = 1.055d-34
  double precision, parameter :: eV = 1.602d-19
  double precision, parameter :: me = 9.109d-31
  double precision, parameter :: pi = 3.141592653589793238


end module constants

program boxvariational
```

```
use constants
implicit none

integer, external :: iargc
character*80 :: buf
double precision, external :: sgrnd,grnd

double precision :: V0
integer :: nparams,nsteps
integer :: seed

integer :: i,j,nrand
double precision :: x,psi,d2psi,a,Eb,Ea,Emin,dE,Te,Te0,vb,Eref
double precision, allocatable :: v(:)

logical :: accept

if (iargc() < 4) then
   print *,'Usage: boxvariational V0[eV] nparams nsteps seed '
   STOP ' '
endif

call getarg(1,buf)
read(unit=buf,fmt=*) V0; V0=V0*eV
call getarg(2,buf)
read(unit=buf,fmt=*) nparams
```

```
call getarg(3,buf)
read(unit=buf,fmt=*) nsteps
call getarg(4,buf)
read(unit=buf,fmt=*) seed

print *,’Program started with V0’,V0/eV,’ nparams’,nparams,’ seed’,seed

allocate(v(nparams))

x=sgrnd(seed)

a=1.0d-10

Eref=pi**2*hbar**2/(2*me*a**2)
print *,’Infinite square well E0’,Eref/eV

! Generate random parameters as starting point
! NOTE: depends on choice of getpsi !
v(1)=-a/2+grnd()*a
v(2)=grnd()*2.0/sqrt(a)

do i=1,nparams
   print *,’rand param’,i,v(i)
enddo

call output(nparams,v,a)
```

```
! Evaluate E(psi) at beginning
call calcpsiEpsi(nparams,v,a,V0,Eb)
print *,'E0',Eb/eV

Te0=Eb*2;
Emin=Eb

! Then for variational optimization with simulated annealing
do i=1,nsteps
   ! Metropolis optimization temperature
   Te=Te0*(1.0-1.1*i/nsteps); if (Te < 0.0d0) Te=0.0
   ! Get Ebefore
   call calcpsiEpsi(nparams,v,a,V0,Eb)
   ! Do trial: change one v by random amount
   nrand=int(grnd()*nparams)+1; if (nrand > nparams) nrand=nparams
   vb=v(nrand);
   v(nrand)=v(nrand)*(0.8+0.4*grnd())
   ! Get Eafter
   call calcpsiEpsi(nparams,v,a,V0,Ea)
   ! Metropolis selection step
   dE=Ea-Eb
   accept=.false.
   if (dE < 0) then
      accept=.true.
   else
```

```
        if (Te > 0.0d0) then
           if (grnd() < exp(-dE/Te)) accept=.true.
        endif
     endif
     if (Ea < Emin) Emin=Ea
     if (.not. accept) then
        v(nrand)=vb
        Ea=Eb
     else
        print '(A,I6,10(G12.4))','i T Ea Emin',i,Te/eV,Ea/eV,Emin/eV,(v(j),j=1,nparams)
     endif
  enddo

  call output(nparams,v,a)

end program boxvariational
```

The subroutine actually carrying out the integration is as follows:

```
subroutine calcpsiEpsi(n,v,a,V0,E)
  use constants
  implicit none

  double precision, intent(IN) :: a,V0
  double precision, intent(INOUT) :: v(*)
  integer, intent(IN) :: n
```

```
double precision, intent(OUT) :: E

integer :: i,nint
double precision :: x,dx,sum1,sum2,psi,d2psi,sumnorm

nint=500
dx=5*a/2/nint

sum1=0.0d0
sum2=0.0d0
sumnorm=0.0d0
do i=-nint,nint-1
   x=(i+0.5)*dx
   call getpsi(x,n,v,a,psi,d2psi)
   ! Nabla squared i.e. kinetic energy term
   sum1=sum1+ psi*(-hbar**2/(2*me)*d2psi*dx)
   ! Potential energy term
   if (abs(x) > a/2) sum2=sum2+psi*V0*psi*dx ! Else it is zero
   !print *,’sum2’,x,V0,psi,dx
   sumnorm=sumnorm+psi*psi*dx
enddo

E=(sum1+sum2)/sumnorm

!print *,’psiEpsi Integral terms’,sum1/eV,sum2/eV
```

```
end subroutine calcpsiEpsi
```

As is evident from the code, this is just an ordinary direct summation numerical integration with no finesse. As explained above, for a multidimensional wavefunction $\psi(\mathbf{x})$ it would be worthwhile to use MC integration to carry out the integral.

The crucial remaining part is of course the subroutine `getpsi` which returns both ψ and $\frac{d^2\psi}{dx^2}$. In the following I will describe three different choices of the functional form.

---

1. Test function to check that method works at all.

As a first case, I constructed a function which should definitely eventually return the exact correct solution.

- I thus chose a cosine function which is cutoff to zero at $a/2$, but made its midpoint random (parameter $v(1)$).

- I also added a quadratic term of strength $v(2)$.

  - This was just to confuse the solver

The subroutine calculating the function is

```
subroutine getpsi(x,n,v,a,psi,d2psi)
  use constants
  implicit none
  !
  ! Subroutine returns value of wave function psi(x) and the second
  ! x derivative of it d$^2$ psi(x)/dx$^2$
  !
  ! The wave function itself is parametrized by the n elements in array v
  !

  double precision, intent(IN) :: x,v(*),a
  integer, intent(IN) :: n
  double precision, intent(OUT) :: psi,d2psi

  integer :: i
  double precision :: AA,BB
  ! Cosine term plus disturbance
  if (n /= 2) then
     print *,'Number of varied parameters wrong'
     STOP
  endif

  ! Correct solution but modified midpoint
  psi=sqrt(1/(a/2))*cos(pi*(x-v(1))/a)
  d2psi=-sqrt(1/(a/2))*pi**2/a**2*cos(pi*(x-v(1))/a)
  if (abs(x) > a/2) then
     psi=0.0d0; d2psi=0.0d0;
  endif
  ! Modify it a bit with a quadratic v(2)
  psi=psi+v(2)*(x/a)**2
  d2psi=d2psi+2*v(2)/a**2
end subroutine getpsi
```

Since we know from the analytical solution that the correct solution is a cosine function centered at $x = 0$, both $v(1)$ and $v(2)$ should go to zero, and the code should return the exact energy solution. The code has $a = 1$ Å and m is the electron mass, hence the ground state is

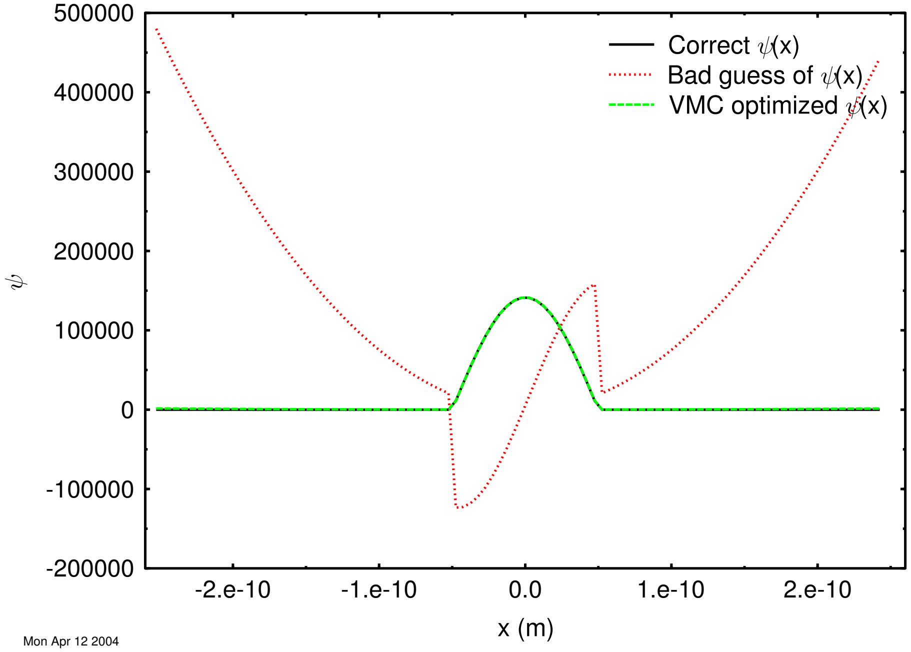
$$E_0 = \frac{\pi^2 \hbar^2}{2ma^2} = 37.64 \text{ eV}. \tag{24}$$

This is actually obtained, below is parts of the code output when ran:

```
 a.out 150 2 1000 1367
 Program started with V0   150.000000000000       nparams           2  seed
        1367
 Infinite square well E0   37.6392928209403
 rand param           1  4.881833728841002E-011
 rand param           2   75290.3363842727
 E0   143.131860975966
i T Ea Emin     1   285.9       141.9       141.9      0.4882E-10  0.6695E+05
i T Ea Emin   100   254.8       146.8       140.4      0.1874E-10  0.1795E+06
i T Ea Emin   200   223.3       143.2       129.5      0.8428E-10  0.7185E+05
i T Ea Emin   498   129.4       48.20       45.17      0.5853E-11   5325.
i T Ea Emin   750   50.10       38.13       37.64      0.1842E-11   1176.
i T Ea Emin   998   0.000       37.64       37.64      0.5950E-13   231.8
```

Note that the parameters $v(1)$ and $v(2)$ both have decreased by about three orders of magnitude, and that the optimized energy is 37.64 eV. So everything behaves as it should.

Here is a plot of the correct wave function, the initial very bad guess and the final optimized wave function:



2. Gaussian wavefunction

In real-life solution of the Schrödinger equation, the correct solution is seldom known. Hence one typically starts from some common functional form and forms the total wavefunction as a sum or product of these. If there are enough terms in the sum, one can then by the variational principle argue that the sum of the individual functions should then eventually approach the correct solution.

One of the more common functions used is the ordinary Gaussian function. For the infinite square well, we know that a Gaussian will never give the correct answer, since it has infinite range whereas the correct solution goes to zero at $\pm a/2$. Nevertheless, I tested it out. The code is almost identical to the one given above, except that the function part itself is now:

```
! Series of Gaussians
if (mod(n,2) == 1) then
   print *,'Number of varied parameters must be even'
   STOP
endif
psi=0.0d0
d2psi=0.0d0
do i=1,n,2
   AA=v(i)
   BB=v(i+1)
   psi=psi+AA*exp(-x*x/(BB**2))
   d2psi=d2psi+2*AA/BB**2*exp(-x*x/BB**2)*(2*x**2/BB**2-1)
```

```
  enddo
```

which gives a sum of any number of Gaussian functions. When ran, this gave e.g.

```
a.out 10000 2 1000 1367
i T Ea Emin   984   0.000        50.83        50.81       0.1003E+05  0.2970E-10
```

i.e. 50.81 eV as the final answer. But this answer is now a function of V_0, since the Gaussian can not go to zero outside the box. Increasing the number of Gaussian wave functions did not help...

In fact, we can now also compare the results to the solution of the finite square well. As explained above, for that it is known that if

$$\sqrt{\frac{mV_0a^2}{2\hbar^2}} = 4 \tag{25}$$

then the ground state solution will be approximately

$$E_0 = 0.0980V_0. \tag{26}$$

Now using our $a = 1$ Å we can solve the above equation to obtain $V_0 = 243.84$ eV, which implies the correct solution $E_0 = 23.89$ eV. Running the code using this value of V_0 gives $E_0 = 25.23$ eV, which is pretty close to the correct solution, but still above it, as it should be by the variational principle.

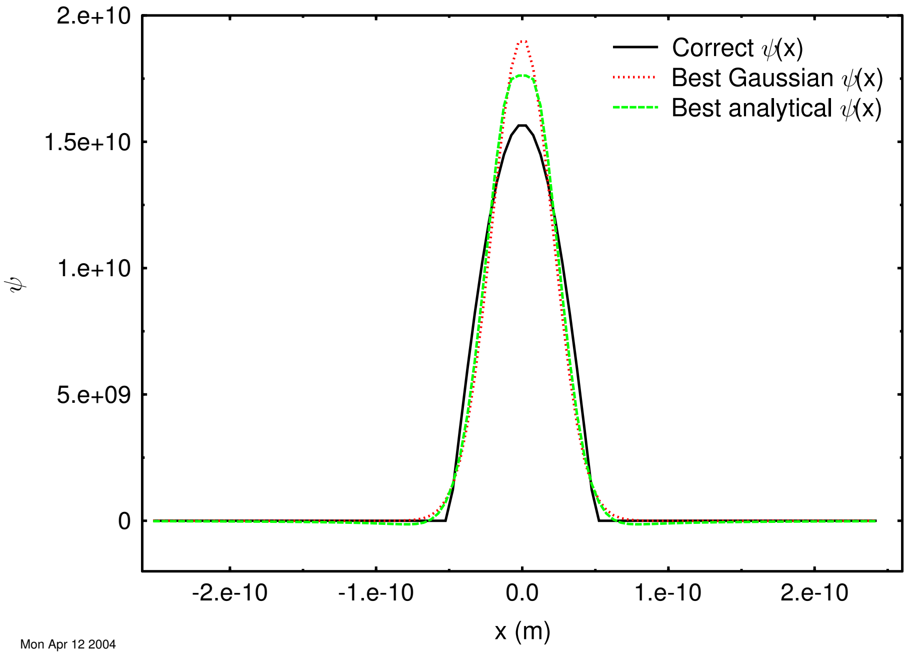
3. We saw above that the Gaussian only does not give a very good solution. In an attempt to obtain a better solution, I still subtracted another peaked function

$$\frac{A}{1+(x/a)^2} \tag{27}$$

from the Gaussian one, to enable partial compensation of the long tail:

```
! Gaussian
psi=psi+AA*exp(-x*x/(BB**2))
d2psi=d2psi+2*AA/BB**2*exp(-x*x/BB**2)*(2*x**2/BB**2-1)
! Inverse polynomial
help1=1+x**2/w**2
psi=psi-AAA/help1;
d2psi=d2psi-AAA*(-2/w**2)*help1*( help1 - 4*x**2/w**2 )/help1**4
```

This gave after a few tests with different seeds and running times a minimum of 46.57 eV, clearly less than that of the Gaussian only. Here is a comparing the best Gaussian and best Gaussian+Inverse polynomial with the exact answer:



By increasing the number of parts of the trial wave function one could almost certainly decrease the energy further, and eventually reach something very close to the exact solution 37.64.

The above example was clearly a toy example, in that the method used is not the most efficient possible one here, and may not be so in any other case either. Nevertheless, it did illustrate a few basic principles which are valid in modern electronic structure calculations:

1. The variational principle plus an optimization scheme can be used to solve the Schrödinger equation

2. By increasing the number and type of trial wave functions (called “basis” functions in molecular and condensed matter physics) one can obtain increasingly accurate solutions of the Schrödinger equation

11.3. Random walk Quantum MC

[Gould-Tobochnik 18.6]

The time-dependent Schrödinger equation was

$$-\frac{\hbar^2}{2m}\frac{\partial^2\Psi(x,t)}{\partial x^2}+V(x,t)\Psi(x,t)=i\hbar\frac{\partial\Psi(x,t)}{\partial t} \tag{28}$$

This can be rewritten in *imaginary* time $\tau = it/\hbar$ as

$$\frac{\hbar^2}{2m}\frac{\partial^2\Psi(x,\tau)}{\partial x^2}-V(x,t)\Psi(x,t)=\frac{\partial\Psi(x,\tau)}{\partial\tau} \tag{29}$$

Now if we rewrite this with the identity $D = \hbar^2/2m$ we get

$$\frac{\partial\Psi(x,\tau)}{\partial\tau}=D\frac{\partial^2\Psi(x,\tau)}{\partial x^2}-V(x,t)\Psi(x,t) \tag{30}$$

Now we see that the first 2 terms are just the diffusion equation. The last term adds a growth or decay term to this equation.

Earlier on the course we saw that a random walk directly simulates diffusion, and can be used to solve the diffusion equation numerically. Since the Schrödinger equation can be written in the form of a diffusion equation for imaginary time, this means that a random walk in imaginary time can be used to solve the Schrödinger equation!.

The ground state energy E_0 can be obtained as

$$E_0 =< V >= \frac{\sum n_i V(x_i)}{\sum n_i} \tag{31}$$

where n_i is the number of walkers at x_i at time τ. The time average of E_0 calculated as above gives the final answer.

The walkers move quite randomly, but the number of walkers is varied according to the potential energy $V(x)$ in a manner which leads to the average energy E_0 obtaining a value which corresponds to the solution of the Schrödinger equation.

For additional details and an algorithm see Gould-Tobochnik.

11.4. Diffusion Quantum MC (DMC)

Diffusion quantum MC is an improvement of the random walk QMC. It also involves random walkers, but also a function which gives the weight of each walker depending on what the potential energy $V(x)$ is at the position of the walker.

11.5. Path integral quantum MC (PIMC)

All of the methods describe above are essentially limited to determining the ground state or lowest-lying excited states of the system. Path integral MC can actually also simulate the state of the system at finite temperatures. It relies on the Feynman path integral formulation of quantum mechanics, and uses a Metropolis MC method to sample the system. The kinetic energy (i.e. temperature) of the system is obtained from the virial theorem.