Exercise Lecture XI

1) Variational Monte Carlo (VMC)

2) Simulated annealing

1. **Quantum oscillator in 1D: variational approach** *(see Fig. 1)*
Consider the hamiltonian of the quantum harmonic oscillator $\mathcal{H} = p^2/2 + x^2/2$ (having choosen $m = k = \hbar = 1$). We want to solve it numerically with VMC. To this purpose consider two different choices for the trial wavefunction:

$$\psi(x) = \begin{cases} B(a^2 - x^2), & \text{for } |x| < a; \\ 0, & \text{for } |x| > a. \end{cases} \qquad (1.a)$$

or:

$$\psi(x) = Ae^{-\beta x^2} \qquad (1.b)$$

with a and β variational parameters, A and B proper normalization constants.

(a) Calculate

$$< E >=< E_L >= \frac{1}{M}\sum_{i=1}^{M} E_L(x_i),$$

with

$$E_L(x) = \frac{\mathcal{H}\psi(x)}{\psi(x)} \quad (local \quad energy)$$

and with the x_i distributed according $\psi(x)^2$. Using the trial wavefunction $(1.b)$, remind the exercise concerning the Metropolis sampling of a gaussian function, Lecture VII; see the code `metropolis_gaussian.f90`. See `metropolis_parabola.f90` for $(1.a)$.

(b) Verify numerically and analytically that the variational solution gives in the two cases:

$(1.a)$: $a = (35/2)^{1/4} \approx 2.0453$; $< E >= 0.6$

$(1.b)$: $\beta = \frac{1}{2}$; $< E >= 0.5$

(*In this case the exact analytic solution is known, $E_0 = 0.5$; however, in case (1.a) the minimum of $< E >$ as a function of a is rather flat: use steps of 0.01 for a and $n = 100000$ Metropolis accumulation steps to appreciate the minimum)*

(c) Instead of solving the problem by minimizing $< E_L >$, solve it minimizing the variance:

$$\sigma^2 =< E_L^2 > - < E_L >^2;$$

you should observe that in the case $(1.a)$ the minimum of the variance is positive, whereas in the case $(1.b)$ is zero.
(Important: note the usefulness of the "zero variance property"; the EXACT minimum of σ^2 is 0 is the class of the trial wavefunctions contains the exact result, whereas in general the minimium of $< E >$ is not known!)

2. **Anharmonic quantum oscillator 1D** *(see Fig. 2)*

   (a) Consider also an anharmonic part in the potential, i.e. $V(x) = x^2/2 + bx^4$. Set for instance $b = 1/8$, make a plot of $V(x)$. Use the first order perturbation theory to calculate the changes at the lowest order in energy of the ground state due to the anharmonic term. Choose a reasonable form for the trial wavefunction, e.g. (1.b), and calculate $< E >$ with VMC. Compare the result obtained with the one obtained using the perturbation theory.

3. **Hydrogen atom.** *(Optional)*

   (a) We want to find numerically the ground state of the H atom. We want to find numerically the ground state. The hamiltonian is: $\mathcal{H} = p^2/(2m) - e^2/r$; It is convenient to use atomic units ($h = 1$, $m_e = 1/2$, e^2=2, and therefore to measure all length in terms of Bohr radius, $a_0 = \hbar^2/(me^2)$, the energies in Rydberg, $Ry = me^4/(2\hbar^2)$. Consider the trial wavefunction $\psi(\mathbf{r}) = \psi(r) = e^{-r/a}$, where a is a variational parameter. calculate the optimal value of ak and the corresponding energy.
   Hints:
   1) you may reduce the problem (spherical part) to a 1D problem. Pay attention that

$$< E_L >= \frac{\int \psi^2(r) E_L(r) d\mathbf{r}}{\psi^2(r) d\mathbf{r}} = \frac{\int \psi^2(r) E_L(r) 4\pi r^2 dr}{\int \psi^2(r) 4\pi r^2 dr}$$

   and therefore the sampling probability is $r^2\psi^2(r)$ and not just $\psi^2(r)$.
   2) Pay attention close to the origin!

   (b) Approach the problem keeping a 3D formulation, considering the isotropy of the trial move (The displacement is a vector!)

4. **Simulated annealing**
   Simulated annealing is a stochastic method for global energy minimization, considering the system starting from a sufficiently high temperature; at each temperature it goes towards equilibrium according to the Boltzmann factor (see the application of the Metropolis algorithm in the canonical ensemble); then the temperature is slightly reduced and the equilibration procedure is repeated, and so on, until a global equilibrium state is reached at T=0. The method can be efficiently used for function minimization, even if the function is not representing an energy. In program `simulated_annealing.f90` it is implemented for the minimization of $f(x) = (x + 0.2) * x + cos(14.5 * x - 0.3)$. Initial temperature, initial position and scaling factor for the temperature are input quantities. Test the program by choosing different initial parameters and scaling factor for the temperature.

```
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! metropolis_gaussian.f90
!
! METROPOLIS sampling of several physical observables for the
! hamiltonian:         h = -1/2 \nabla^2 + 1/2 x^2),
! comparison exact expected results with numerical results
! on psi^2(x), with  psi(x) = exp(-x^2/(4\sigma^2))
! \sigma=1 => psi^2(x) = costant * standard gaussian
!  P(x) =  exp(-x**2/(2*sigma**2))/sqrt(2*pi*sigma**2)
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

program metropolis_gaussian
  implicit none
  integer, parameter :: dp=selected_real_kind(13)
  integer :: i,n
  real(kind=dp):: sigma,etot,ekin,epot,rnd,ekinL,epotL,etot2
  real (kind=dp) :: pigr,pi2b,var,beta,var_th
  real(kind=dp):: x,x1,x2,xp,delta,expx,expxp,p,acc
  character(len=13), save :: format1 = "(a7,2x,2f9.5)"
  open(unit=7,file='e_var_gauss.dat',position='append')

  pigr = 2*asin(1.0_dp)
  acc = 0.0_dp
  x1 = 0.0_dp
  x2 = 0.0_dp
  ekin = 0.0_dp
  epot = 0.0_dp
  etot2= 0.0_dp
  print*, "n, sigma  (remember: beta = 1 / (4*sigma**2)), x0, delta"
  read*,   n,sigma,x,delta
  beta = 1 / (4*sigma**2)
!  call random_seed(put=seed)

  do i=1,n
     ekinL = - 0.5_dp * ((x/(2*sigma**2))**2 - 1/(2*sigma**2))
     epotL = 0.5_dp * x**2
     ekin = ekin + ekinL
     epot = epot + epotL
     etot = ekin + epot
     etot2 = etot2 + (ekinL + epotL)**2
     x1 = x1 + x
     x2 = x2 + x**2
     !ccccccccccccccccccccccccccc
     expx = - x**2 /(2*sigma**2)    !
     call random_number(rnd)        !
     xp = x + delta * (rnd-0.5_dp)  !
```

```
      expxp = - xp**2 /(2*sigma**2)  !   metropolis
      p = exp (expxp-expx)           !   algorithm
      call random_number(rnd)        !
      if (p > rnd) then              !
         x = xp                      !
      !ccccccccccccccccccccccccccccc
         acc=acc+1.0_dp
      endif
   enddo

   var_th =  1._dp/(32*beta**2)+beta**2/2-1._dp/4

   write(unit=*,fmt=*)"acceptance ratio = ",acc/n
   write(unit=*,fmt=*)"# Results (simulation vs. exact results):"
   write(unit=*,fmt=format1)"etot = ",etot/n,1.0_dp/(8.0_dp*sigma**2)&
        +0.5_dp*sigma**2
   write(unit=*,fmt=format1)"ekin = ",ekin/n,1.0_dp/(8.0_dp*sigma**2)
   write(unit=*,fmt=format1)"epot = ",epot/n,0.5_dp*sigma**2
   write(unit=*,fmt=format1)"evar = ",etot2/n-(etot/n)**2,var_th
   write(unit=*,fmt=format1)"<x>  = ",x1/n,0.0_dp
   write(unit=*,fmt=format1)"<x^2>= ",x2/n,sigma**2

   write(7,*)sigma,etot/n,sqrt(abs(etot2/n-(etot/n)**2)),etot2/n-(etot/n)**2

  close(7)

end program metropolis_gaussian
```

```
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! simulated_annealing.f90
! for function minimization; adapted from U. Schmitt, 2003-01-15
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
PROGRAM anneal
IMPLICIT NONE
INTEGER :: istep, nsteps
REAL, PARAMETER :: scale=0.5 ! should be chosen for specific function
REAL :: func, fx, fx_min, fx_new, temp, tfactor, x, x_min, x_new
REAL, DIMENSION(2) :: rand ! random numbers
x = 1.0; fx = func(x); fx_min = fx  ! starting point for search
PRINT *, 'Starting from x = ', x, ', f(x) = ', fx

PRINT *, 'initial (high) temperature (e.g., 10)?'  ! annealing schedule
READ *, temp
PRINT *, 'annealing temperature reduction factor (e.g., 0.9)?'
READ *, tfactor
PRINT *, 'number of steps per block (equilibration, e.g., 1000)?'
READ *, nsteps

Do WHILE (temp > 1E-5) ! anneal cycle
  DO istep = 1, nsteps
    CALL RANDOM_NUMBER(rand) ! 2 random numbers
    x_new  = x + scale*SQRT(temp)*(rand(1) - 0.5) ! stochastic move
    fx_new = func(x_new) ! new object function value
    IF (EXP(-(fx_new - fx)/temp) > rand(2)) THEN ! success, save
      fx = fx_new
      x = x_new
    END IF
      write(1,fmt=*)temp,x,fx
    IF (fx < fx_min) THEN
      fx_min = fx
      x_min = x
      PRINT '(3ES13.5)', temp, x_min, fx_min
    END IF
  END DO
  temp = temp * tfactor ! decrease temperature
END DO

End PROGRAM anneal

REAL FUNCTION func(x) ! Function to minimize
Implicit NONE
REAL :: x
func = (x + 0.2)*x + cos(14.5*x - 0.3)
END FUNCTION
```

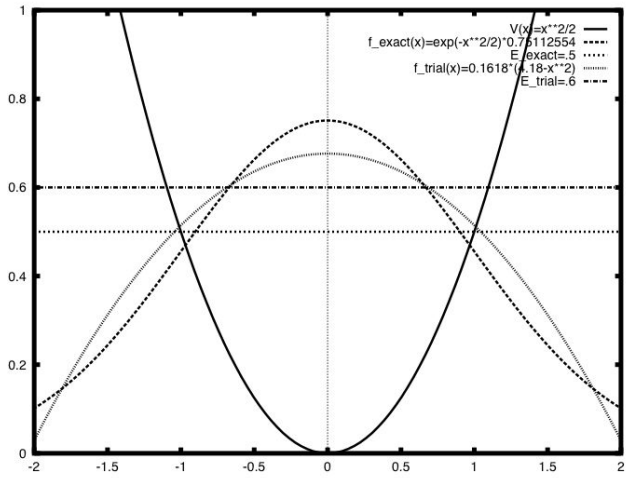

Figure 1: **Harmonic oscillator in 1D with VMC (ex. 1)**: Ground state Eigenvalue and Eigenstate with two different trial wavefunctions.

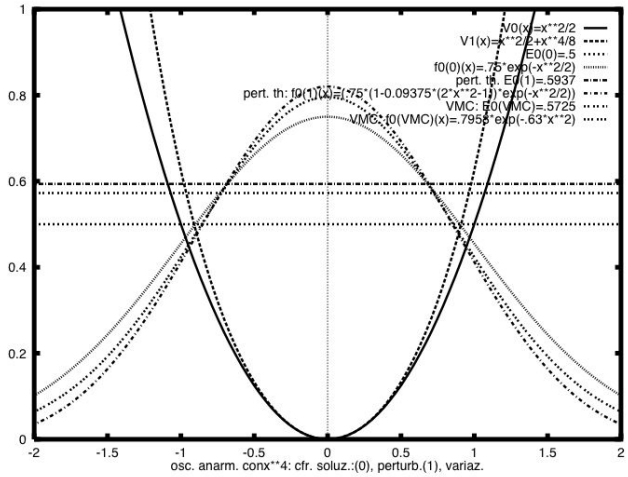

Figure 2: **Anharmonic oscillator in 1D with VMC (ex. 2)**: Ground state Eigenstate and eigenvalue with two different trial wavefunctions, and comparison with the analytic solution obtained within the first order perturbation theory.