P3800 Project 4: Monte Carlo Simulations: Integration and 2D Ising Model.

NOTE: This is your last project for P3800. It is hoped that your solution and write-up will reflect the learning you have achieved in this course. You are encouraged to be creative in your coding (if so, make sure you explain what you did) as well as in the application of your code to solve the specific problems given below and in the presentation of your results.

1. (20 pts.) Monte Carlo Integration in 3D and 10D.

   For background information see
   http://math.fullerton.edu/mathews/n2003/MonteCarloMod.html.

   (a)Write a MC integration program in Fortran or C to estimate the value of the following 3D integral:

   $$I_1 = \int_0^{9/10} \int_0^1 \int_0^{11/10} dxdydz(4 - x^2 - y^2 - z^2)$$

   where the exact result is $14817/5000 = 2.9634$.

   (b)Some physical problems require integration in many dimensions. For example, properties of a small atom like magnesium with 12 electrons requires integration over all three dimensions for all 12 particles giving a 36 dimensional integral. If we use 64 points for each integration, this requires $64^{36} \simeq 10^{65}$ evaluations. Even with a fast computer that can evaluate the integrand one million times per second, this would take 10^{50} seconds, which is significantly longer than the age of the universe (about 10^{17} seconds). The results of your task show how such problems can be done using MC techniques in the time it takes to eat lunch.
   Write a MC integration program in Fortran or C to estimate the value of the following 10D integral:

   $$I_2 = \int_0^1 dx_1 \int_0^1 dx_2 \cdots \int_0^1 dx_{10}(x_1 + x_2 + \cdots + x_{10})^2$$

   where the exact result is $155/66 = 25.8333$
   Be sure to write your code so there is a new random number seed used each time it is run. Explore and report the impact of both re-running the code several times (say 3 times) at the same number of MC steps (MCS) as well as running the code with larger MCS (say 1000, 10000 and 100000). You may decide some sort of graph would be useful. Be sure to attach a copy of your codes to your write-up (pdf).

2. (80 pts.) Ising model in 2D.
   For background information, see MacKinnon's notes (pdf on course web page) and almost any book on statistical mechanics or statistical physics.

Your task is to write computer code in Fortran or C based on the Metropolis Monte Carlo method which simulates thermal averages of the nearest-neighbor Ising model on a square lattice having a spin-dependent energy of the form:

$$E = -J \sum_{i,j}^{NN} S_i S_j$$

where i, j is a sum over nearest neighbors (NN) only and $S_i = \pm 1$. You will use periodic boundary conditions and set $J = 1$ and $k_B = 1$ for convenience. This model can be solved exactly and shows a transition to the paramagnetic state where $M = 0$ to a state where $M \neq 0$ at a transition temperature $T_c \simeq 2.269$. Here, $M(T)$ is the magnetization $M = (1/N) < \sum_i S_i >$ where N is the number of spins S_i and the brackets mean thermal average over Monte Carlo Steps (MCS) as described in class and in MacKinnon's notes: see his Project - Ising Model description starting on page 43 (I do not recommend the mapping to a 1D problem). In addition to $M(T)$, you should also calculate the heat capacity $C_V(T)$

$$C_V = \frac{1}{N} \frac{1}{(k_B T^2)} [\langle E^2 \rangle - \langle E \rangle^2]$$

and magnetic susceptibility $\chi(T)$

$$\chi = \frac{1}{N} \frac{1}{(k_B T)} [\langle S^2 \rangle - \langle S \rangle^2]$$

You should get simulation results for these quantities for a range of temperatures around T_c, something like $1.5 \leq T \leq 3.0$ in steps of 0.05. Recall from class that for an infinite lattice, $C_V(T)$ and $\chi(T)$ should diverge at T_c. It is useful to start at the higher T (using either a ferromagnetic or a random configuration for the initial S_i) and cool the system down, using the final configuration from the previous T as the initial configuration of the new T. It is also best to discard the first 10% of the runs (MCS0) for thermal equilibration before using the results to get thermal averages. You should consider the effect of different lattice sizes $L \times L$ with L ranging from 12 to 28 (in steps of 8), as well as the impact of using different number of MCS (say, 5000, 15000 and 25000). NOTE that the number MCS *per spin* will decrease as L increases (for constant MCS), thus impacting the quality of your answer at larger lattices (unless you use a larger number for MCS).

Below is an example of some very old (and crude) Fortran code for the 3D Ising model which contains parts you may find inspiring. It calculates M at one particular value of T. Your code can be structured in a much better manner. For example, you can:
- have a separate part for the periodic BC's and store them.
- have a separate part that stores the five possible values of ΔE (see MacKinnon).
- calculate the energy for a given configuration of spins using a subroutine.
- consider choosing spins to flip (or not) S_i randomly instead of systematically.
- flip a spin or not based on the two 'if' statements mentioned in class.

In your write-up you should discuss any particular coding issues you think are relevant. There should be lots of graphs of the quantities vs T for different lattices sizes and different MCS. Maybe even a plot of your estimates for T_c vs L would be useful (try

various log-log or semi-log plots). Does it extrapolate to the correct value ? How do divergences scale with N ? Also consider including illustrative results for S_i at each lattice point for the case of $T << T_C$ as well as $T >> T_C$ which should show nearly all up-spins in the former case and nearly equal numbers of up-spins and down-spins in the latter case. Submit your completed project electronically (as with previous projects) and also a hard copy of your write-up (pdf). Be sure to attach a copy of your codes to your write-up.

Application of Monte Carlo Method in Stat. Phys. Ed. K. Binder (1987)

Table 1.1. Example of a program for a 37*37*37 Ising model of a simple cubic Ising lattice at $T/T_c = 1.4$, calculated for 25 Monte Carlo steps per spin. *Ranset* initializes the random number generator RANF. The energy change is calculated as in (1.5), the flipping probability is taken as $\exp(-\Delta E/k_B T)$. A CDC Cyber 76 took about 3.75 s for this program

```
      DIMENSION IS(37,37,37),EX(13)
      DATA IS/50653*1/
      T=1.40/0.221655
      L=37
      CALL RANSET(1)
      M=L*L*L
      DØ 3 I=1,13,2
    3 EX(I)=EXP(-2*(I-7.)/T)
      DØ 2  ITIME=1,25
      DØ 1 K1=1,L
      K1P1=K1+1
      K1M1=K1-1
      IF(K1.EQ.1)K1M1=L
      IF(K1.EQ.L)K1P1=1
      DØ 1 K2=1,L
      K2P1=K2+1
      K2M1=K2-1
      IF(K2.EQ.1)K2M1=L
      IF(K2.EQ.L)K2P1=1
      DØ 1 K3=1,L
      K3P1=K3+1
      K3M1=K3-1
      IF(K3.EQ.1)K3M1=L
      IF(K3.EQ.L)K3P1=1
      IEN=7+IS(K3,K2,K1)*(IS(K3M1,K2,K1)+IS(K3P1,K2,K1)+IS(K3,K2M1,K1)
     1                   +IS(K3,K2P1,K1)+IS(K3,K2,K1M1)+IS(K3,K2,K1P1))
      IF(EX(IEN).LT.RANF(I)) GØTØ 1
      IS(K3,K2,K1)=-IS(K3,K2,K1)
      M=M+2*IS(K3,K2,K1)
    1 CONTINUE
    2 WRITE (6,4)M,ITIME
    4 FORMAT(2I9)
      STOP
      END
```

$e^{-\Delta E/T}$

Boundary Conditions