Physics 212 – Problem Set #9

Solutions

1.) a.) $\mu(\rho) = -k_BT + P/\rho - a\rho + k_BT\log(\lambda^3\rho)$

$- k_BT\log(1-b\rho)$

minimize:

$$0 = \frac{\partial\mu}{\partial\rho} = -\frac{P}{\rho^2} - a + \frac{k_BT}{\rho} + \frac{b\,k_BT}{1-b\rho}$$

$$\left(\frac{P}{\rho^2} + a\right) = k_BT\,\frac{1}{\rho(1-b\rho)}$$

$$(P + a\rho^2)(1-b\rho) = k_BT\rho$$

$$\left(P + a\left(N/V\right)^2\right)\left(1 - b\frac{N}{V}\right) = k_BT\frac{N}{V}$$

$$\left(P + a\left(\frac{N}{V}\right)^2\right)(V - bN) = Nk_BT$$

this is the van der Waals equation of state.

b.) According to the figure caption, the van der Waals coexistence pressure at $T = 373^\circ K$ is $P = 1.5\times10^7\ \text{dyne/cm}^2$

Since $100^\circ C = 373^\circ K$ is the boiling point of water, the coexistence pressure of water at this temperature should be

$$1\,\text{atm} = 10^6\ \text{dyne/cm}^2$$

The parameters a and b in Fig. 11.13 are apparently fit to very high temperature data. Set

$$\rho_0 = 1\,g/cm^3 \div m_{H_2O}$$
$$= 1\,g/cm^3 \,/\, (18 \times 1.66 \times 10^{-24}\,g)$$
$$= 3.3 \times 10^{22}\ \text{molecule}/cm^3$$

with $b = 5.05 \times 10^{-23}\ cm^3/\text{molecule}$ $\quad b\rho_0 = 1.7$

But, the van der Waals equation makes sense only for $b\rho < 1$.

c.) $$\Delta G = \int \rho(x)\,[\mu[\rho(x)] - \mu_0]\, d^3x$$

$$\sim (10^{-2}\,\text{mole}/cm^3) \cdot N_A \cdot 0.4 \times 10^{-13}\ \text{erg/molecule} \cdot 3\text{Å} \cdot \text{Area}$$

then

$$\frac{\Delta G}{\text{Area}} \sim 7\ \text{erg}/cm^2 \quad \text{or} \quad 7\ \text{dyne}/cm$$

2.) Since this problem is mainly running the simulation and watching it, I'll give an abbreviated answer.

c.) On the next pages, I give my Java code for the Ising model in a magnetic field.

for $H = -0.2$ $T = 1.5$ I found that the bubble grows from initial size

$R =$	4	5	6
for	1 in 10	6 in 10	10 in 10

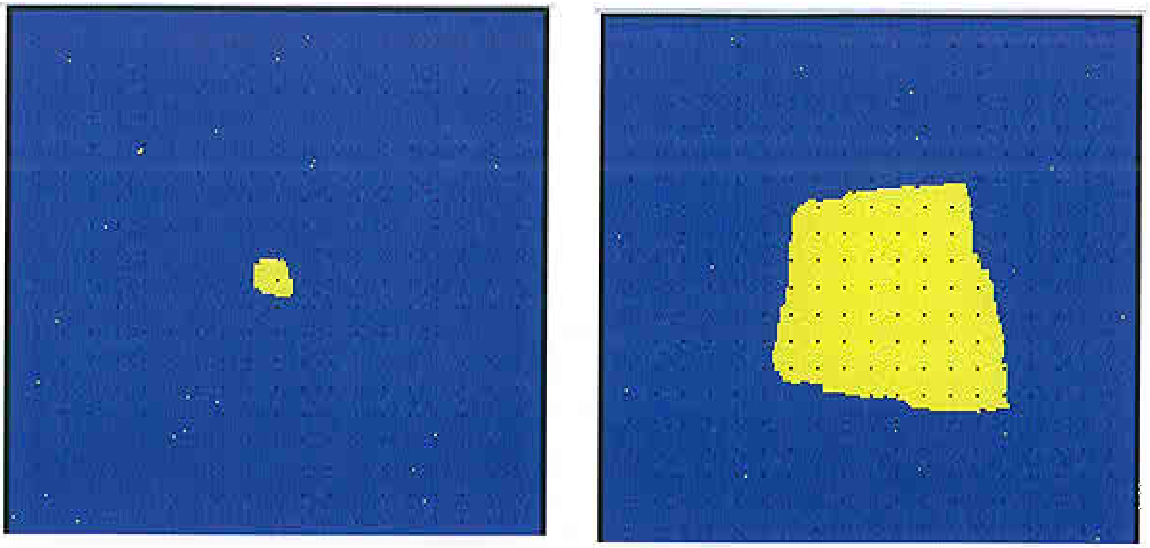
runs.

so $R = 5$ is approximately the critical bubble size under these conditions

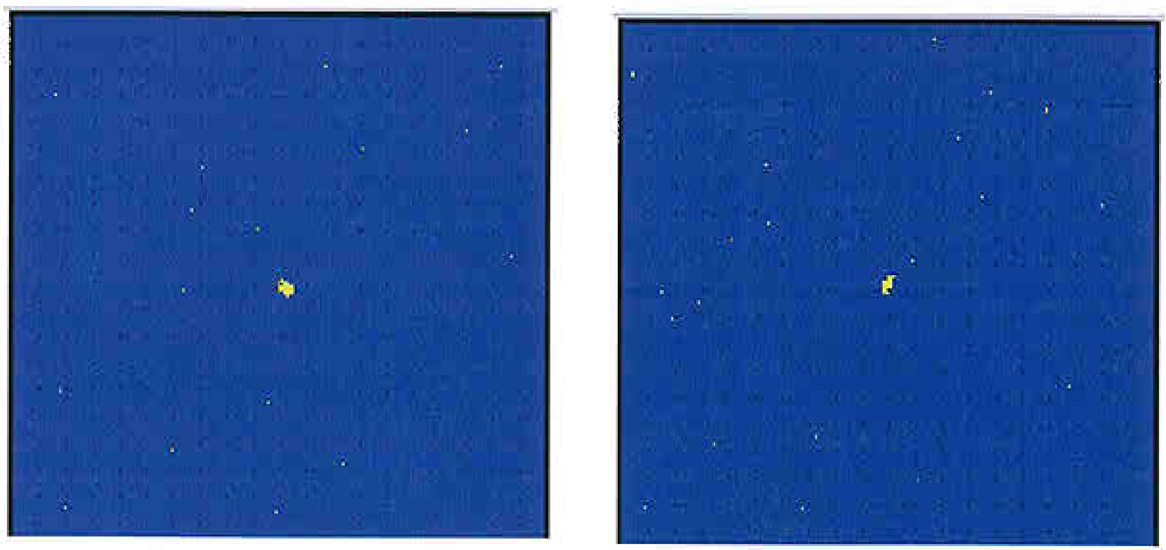
```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;

public class myIsing2 extends IsingGUI {

    double h;

 /*     solve() carries out the Metropolis method for updating a
          2-dimensional lattice of spins.   The updating runs
           continuously until you push the Stop button of the
              applet.

         The spin values are given as the elements of an array of
            integers called

                  State[m][n]

         The entries of State should take the values 1 or -1.

         n and m range over the values  0, 1, 2, ...   Nx or Ny
           (If you call State[m][n] for m or n outside this
            range, bad things will happen.)

          Nx and Ny are set equal to 200; please do not change this.
             The program then simulates a 200 x 200 spin model.

         To implement fixed-spin boundary conditions, do Metropolis
              updates for   m, n = 1 .. Nx-1  only
         To implement periodic boundary conditions, do Metropolis
              updates for   m, n = 1 .. Nx-1  only
           and,
              when the 1 element is updated, copy it to the Nx element
              when the Nx-1 element is updated, copy it to the 0 element


          The temperature is available as a global variable called T.

          The method below is set up to simulate the thermodynamics of
              the Hamiltonian

                   H  =  -    sum_{nm}   S_{nm}/T

              corresponding to an external magnetic field  h = 1
               and no interaction.   You should change this Hamiltonian
                  appropriately to simulate the Ising model with J = 1 .

            The value of the energy/site will be reported in the applet
              if you implement the function   computeEnergy()  below

                                                                          */


    void solve(){
        /* intialization */
      h = -0.2;
      int i,j;
      int Ns = 4;
      for (i = 1; i <= Nx; i++){
          for (j = 1; j <= Ny; j++){
              State[i][j] = 1;
      /*  add a reversed-spin droplet of size Ns   */
```

```
            if ( (i - Nx/2)*(i-Nx/2) + (j - Ny/2)*(j-Ny/2) < Ns*Ns )
                State[i][j] = -1;
        }
    }
      /*   sweep through the lattice and do Metropolis updating */
    int N = 0;
    while (true){
        N++;
            /* impose periodic boundary conditions */
       for (i = 2; i <= Nx-1; i++){
          State[i][0] = State[i][Ny-1];
          State[i][Ny] = State[i][1];
       }
       for (j = 2; j <= Ny-1; j++){
          State[0][j] = State[Nx-1][j];
           State[Nx][j] = State[1][j];
       }
               /* now sweep */
       for (i = 1; i < Nx; i++){
        for (j = 1; j < Ny; j++){
            int S = State[i][j];
            double Hnoflip = H(i,j,S);
            double Hflip = H(i,j,-S);
            if (Hflip < Hnoflip){
                State[i][j] *= -1;
            }  else {
                double r = ran();
                      /* gets a random number between 0 and 1 */
                double ratio = Math.exp( - (Hflip - Hnoflip)/T);
                if (r < ratio)  State[i][j] *= -1;
            }
        }
      }
        /* put a large number here to make the updating run slower */
      if (timetostop) break;
    if (N%2 == 0) refreshPicture();
    }
  }

  double H(int m, int n, int S){
      /*  value of H including only terms involving
                S = State[m][n]
            S is kept as an argument so we can flip it
            With more complicated Hamiltonian, the
                return value would depend on State[m+1][n]
                      and other neighboring spins       */
      return   - ( S * (State[m+1][n] + State[m-1][n] + State[m][n+1]
                                   + State[m][n-1])
                       + h * S) ;
  }

  double computeEnergy(){
      int i,j;
         /* should return  E/N   */
      double EE = 0.0;
      for (i = 1; i < Nx; i++){
        for (j = 1; j < Ny; j++){
              EE +=  H(i,j,State[i][j]);
        }
      }
      return EE/((Nx-1)*(Ny-1));
  }
```

evolution of bubbles at $T = 1.5$, $H = -0.2$

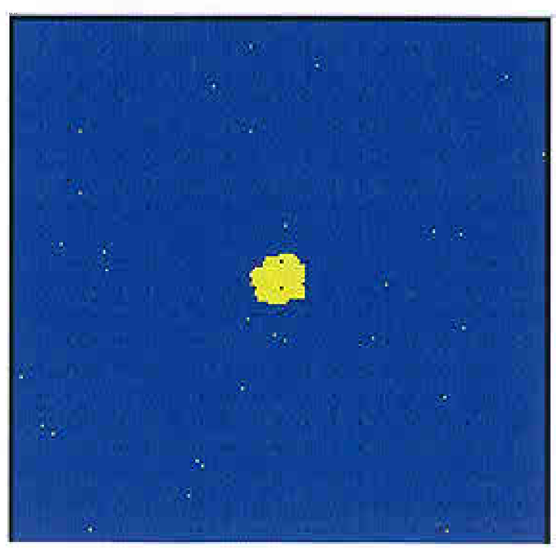


with $R = 6$, the almost every bubble grows



with $R = 4$, almost every bubble shrinks

but here is an example of a bubble growing from $R = 4$



3.) a.)

$$\frac{d\rho_n}{dt} = a\left(n\rho_{n-1}N_2 - n\rho_n N_1\right.$$
$$\left. - (n+1)\rho_n N_2 + (n+1)\rho_{n+1}N_1\right)$$

the first line is the effect of $n-1 \leftrightarrow n$ transitions
the second line is the effect of $n \leftrightarrow n+1$ transitions

$$\langle n\rangle = \sum_0^\infty n\rho_n \qquad \sum_0^\infty \rho_n = 1$$

so

$$\frac{d\langle n\rangle}{dt} = a\sum_n n\left\{ n\rho_{n-1}N_2 - (n+1)\rho_n N_2 \right.$$
$$\left. - n\rho_n N_1 + (n+1)\rho_{n+1}N_1 \right\}$$

shift $n \to (n\pm1)$

$$= a\sum_n \rho_n \left\{ (n+1)^2 N_2 - n(n+1)N_2 \right.$$
$$\left. - n^2 N_1 + (n-1)n\, N_1 \right\}$$

$$= a\sum_n \rho_n \left((n+1)N_2 - nN_1\right)$$

$$\frac{d\langle n\rangle}{dt} = a\left[(1+\langle n\rangle)N_2 - \langle n\rangle N_1\right]$$
$$> 0 \quad \text{for } N_2 > N_1$$

b.) let $\epsilon = \frac{N_2 - N_1}{N_1}$

$$\frac{d\langle n\rangle}{dt} = a\,N_1\left[(1+\langle n\rangle)(1+\epsilon) - \langle n\rangle\right]$$

$$= a\,N_1\left[1+\varepsilon + \varepsilon\langle n\rangle\right]$$

if $\varepsilon < 0$, there is an equilibrium $\frac{d\langle n\rangle}{dt} = 0$

for $\langle n\rangle = \frac{1-|\varepsilon|}{|\varepsilon|} \sim \frac{1}{|\varepsilon|}$

the general solution of the equation for $\langle n\rangle(t)$ with $\langle n\rangle(0) = n_0$

is

$$\langle n\rangle = \frac{1-|\varepsilon|}{|\varepsilon|} + \left(n_0 - \frac{1-|\varepsilon|}{|\varepsilon|}\right)e^{-aN_1|\varepsilon|\,t}$$

The relaxation time is then $t_r = \frac{1}{aN_1|\varepsilon|} \sim \frac{1}{|\varepsilon|}$

so $\langle n\rangle \sim |\varepsilon|^{-\nu}$ $\quad t_r \sim |\varepsilon|^{-\eta}$ with $\nu = \eta = 1$

c.) $\frac{d\rho_n}{dt} = 0$ for

$$n\rho_{n-1} N_2 - (n\rho_n N_1 + (n+1)\rho_n N_2) + (n+1)\rho_{n+1} N_1 = 0$$

$$\rho_{n+1} = \rho_n \left(\frac{N_2}{N_1} + \frac{n}{n+1} \right) - \frac{n}{n+1} \frac{N_2}{N_1} \rho_{n-1}$$

so, in terms of ρ_0

$$\rho_1 = \frac{N_2}{N_1} \rho_0$$

$$\rho_2 = \left(\frac{N_2}{N_1} + \frac{1}{2} \right) \rho_1 - \frac{1}{2} \frac{N_2}{N_1} \rho_0$$

$$= \left[\left(\frac{N_2}{N_1} \right)^2 + \frac{1}{2} \frac{N_2}{N_1} - \frac{1}{2} \frac{N_2}{N_1} \right] \rho_0$$

$$= \left(\frac{N_2}{N_1} \right)^2 \rho_0$$

try $\rho_n = \left(\frac{N_2}{N_1} \right)^n \rho_0$ with this as the induction

hypothesis for $n=n$

$$\frac{\rho_{n+1}}{\rho_0} = \left(\frac{N_2}{N_1} \right)^n \left(\frac{N_2}{N_1} + \frac{n}{n+1} \right) - \left(\frac{N_2}{N_1} \right)^{n-1} \frac{N_2}{N_1} \frac{n}{n+1}$$

$$= \left(\frac{N_2}{N_1} \right)^{n+1} \quad \checkmark$$

If $N_2/N_1 = e^{-\beta \Delta E}$, $\rho_n = \rho_0 e^{-n\beta \Delta E}$

d.) Now let $\frac{N_2 - N_1}{N_1} = \varepsilon$ $\quad \frac{N_2}{N_1} = 1+\varepsilon$

$$\rho_n = \rho_0 (1+\varepsilon)^n = \rho_0 \left(1 + \frac{n\varepsilon}{n}\right)^n$$

for ε small and < 0 $\quad \cong \rho_0 e^{-n|\varepsilon|}$

This seems to be a scaling form

$$\rho_n(\varepsilon) \sim n^{-\tau} D(n|\varepsilon|^{\nu})$$

with $\tau = 0$ $\quad D(x) = e^{-x}$