MATH 577, ASSIGNMENT 4. Version: January, 2006

In the following two problems we will use Markov Chain Monte Carlo (MCMC). In the first problem we will learn to use it to clean up a noisy image via a Markov chain procedure. In the second problem we will learn to estimate parameters in a partial differential equation.

1. Included below is a matlab code which cleans up a noisy image using a uniform prior. Your assigment is to modify the code in order to use an Ising prior. The Ising prior, as you recall, is

$$P(\mathbf{x}) \propto e^{-2J\#\mathbf{x}},$$

where $J = 0$ corresponds to the uniform case, and $J > 0$ is the Ising case. Set $J = 0.5 - 2$ in your code modification.

In order to run the existing code you'll need a clean image called `cleanimage.bmp`. You do not have to use bmp format, you can use some other image format. But it should be one that contains pixels of values 0 or 1 (a binary image). What I used is a gif of a symbol (a peace sign) blown up so that it looks jagged.

The code will first dirty it up, as you can see, and then attempt to recover it. Use the sample code, modify it to implement the Ising case.

```
clear
%reconstype = input('enter 1=regular, 2=Ising \n')
reconstype = 1;

F=imread('cleanimage.bmp');
[M,N]=size(F);
sigma = 2;
d=double(F); d(find(d==0))=-1;
d = d + sigma*randn(size(d));
figure(1);
subplot(1,2,1),imagesc(F);set(gca,'Visible','off');
colormap gray;
axis square;
subplot(1,2,2),imagesc(d);set(gca,'Visible','off');
colormap gray;
axis square;
drawnow;
```

```
pause
J=0.5;
f=ones(M,N);
figure(3)
subplot(1,2,1),hf=imagesc(f);set(gca,'Visible','off')
colormap gray; axis square; drawnow;
mf = zeros(M,N);

subplot(1,2,2),hm=imagesc(mf);set(gca,'Visible','off')
colormap gray; axis square; drawnow;
SS= 10000;
misfit = [];
adj=[1 1 0 0 ; 0 0 -1 1];
iter = 0;

while 1
  ix = ceil(N*rand(1)); iy=ceil(M*rand(1));
 pos = iy + M*(ix-1); fp=-f(pos);
 if reconstype == 1
    LkdRat = exp(d(pos)*(fp-f(pos))/sigma.^2);
    alpha = LkdRat;
 else
    % place the Ising modification here
 end
 if rand < alpha % Prob of acceptance = min(1,alpha)
     f(pos) = fp;
    end
 iter = iter + 1;
 if rem(iter,SS) == 0
     mf = mf + f;
     NS = iter/SS;
     iter;
     set(hf,'CData',f);
     set(hm,'CData',mf);
     drawnow
     if iter/SS > length(misfit)
         misfit = [ misfit,zeros(100,1)];
         misfit(iter/SS) = sum(sum(d-f).^2)/sigma;
     end
 end
```

```
end
```

2. The Fisher equation

$$Z_t = rZ(K - Z) + DZ_{xx}$$

is a classic equation of mathematical biology. The parameters to determine are carrying capacity K, the growth rate r, and D the diffusion. You will be solving this equation with zero Dirichlet boundary conditions. At a time $t = T_{max}$ and at x values between $x = 0$ and $x = L$, we want to estimate the growth rate r, carrying capacity K and diffusivity D.

For initial conditions assume that the population density $Z(x, t)$ is given by 1, for $0.75L$ and $0.8L$, and zero otherwise. Here, $L = 10$. At positions $x = x_1, x_2, ..., x_K$ the density is measured and given by the data vector

$$y_i(T_{max}) = Z(x_i, T_{max}) + \eta_i,$$

where $i = 1, 2, ..., M$, and η_i is normally distributed with variance $s = 0.03$. $T_{max} = 2$. Some suggested values for the discretization are $M = 25$, and take about 100 time steps. A good guess of the parameters is $r = 1$, $K = 2$, and $D = 1.5$. The parameters are all non-negative.

Ordinarily the data comes from the laboratory/field. But here we have to make it up: you will also need to produce some data $y_i(T_{max})$. Use the Fisher equation solver with noise to synthetically produce the data.

The equation solver to use is

```
Z=Z0;

for tt=2:tdim
    ttp=tt-1;
    for i=1:M
        if i == 1
           Z(tt,i)=Z(ttp,i)+r.*q1.*Z(ttp,i).*(K-Z(ttp,i))...
             + D.*q2.*(Z(ttp,i+1)-2.*Z(ttp,i));
        elseif i == M
           Z(tt,i)=Z(ttp,i)+r.*q1.*Z(ttp,i).*(K-Z(ttp,i))...
             + D.*q2.*(-2.*Z(ttp,i)+Z(ttp,i-1));
```

```
        else
           Z(tt,i)=Z(ttp,i)+r.*q1.*Z(ttp,i).*(K-Z(ttp,i))...
             + D.*q2.*(Z(ttp,i+1)-2.*Z(ttp,i)+Z(ttp,i-1));
       end
    end
end
```

$q1 = dt$ and $q2 = dt/dx^2$ are the discretization parameters. Here i is the index associated with x_i. To this solver you should pass the parameters r, K, D which are to be estimated.

Generate trials for the vector Xp in terms of the old X, where $X = [r, K, D]$:

```
Xp=X+w*(2*rand-1);
if (Xp>0) % use the trial otherwise generate a different one
   do MCMC{solve for Z(T_{max})
end
```

You will need something like tens of thousands of MCMC trials.

Plot the estimated solution at T_{max} as the parameters are estimated. Also plot the parameter mean along with their uncertainties. You can calculate the mean and the uncertainty "on the fly" so that you do not have to keep values of X_p.