

```

% make sure we have a clean environment
clear
rand('state',0);
randn('state',0);

% Global variables
% H    - the recorded head for each time
% TM   - the times the head was recorded
% SIGMA - the standard deviation for a time
% D    - the distance between the wells
% Q    - the volume of the slug
global H;
global TM;
global SIGMA;
global Q;
global D;
global stepsize

%
% Load the data. Head is measured to the nearest centimeter.
%
load slugdata.mat
%
%
% Fixed parameter values.
%
D=60;
Q=50;

% We'll use sigma=1cm.
SIGMA=0.01*ones(size(H));

S_sig = 2e-5;
T_sig = 4e-2;
stepsize = [S_sig; T_sig];
S0 = 5e-3; T0 = 1;
m0 = [S0;T0];
N = 2e5; BURNIN = 1e4;
skip = 1000;
4.(a) Identify the number of samples that remain after the thinning process. Plot the hisstory of the
thinned posterior samples, and include the true parameters.
logprior = @(m) Logprior(m);
loglikelihood = @(m) Loglikelihood(m);
generate = @(x,stepsize) Generate(x);
logproposal = @(x,y,stepsize) Logproposal(x,y);

[mout,mMAP,pacc]=mcmc(logprior,loglikelihood,generate,logproposal,m0,N)
%downsample results to reduce correlation
k=(BURNIN:skip:N);
mskip=mout(:,k);
disp(['The corresponding number of samples = ',num2str(length(mskip)),' samples']);

```

```

mtrue = [0.00207;0.585];

%plot parameter sample histories
figure(1)
clf
for i=1:2
    subplot(2,1,i)
    plot([1 length(mskip)],[mtrue(i) mtrue(i)],'Color',[0.6 0.6 0.6],'LineWidth',3);
    hold on
    plot(mskip(i,:), 'ko')
    hold off
    if i~=2
        set(gca,'Xticklabel',[]);
    end
    xlim([1 length(mskip)])
end
xlabel('Sample Number')
subplot(2,1,1)
ylabel('S')
title('Displaying the true parameters and thinned sample histories')
subplot(2,1,2)
ylabel('T')

```

b) Plot autocorrelations of S and T both before and after thinning. Discuss if there was enough thinning with skipping every 1000 samples.

```

figure(2)
clf
laglen=1000;
lags=(-laglen:laglen)';
acorr=zeros(2*laglen+1,2);
acorrn=zeros(2*laglen+1,2);

for i=1:2
    acorrn(:,i)=calc_corr(mout(i,:),laglen);
end

for i=1:2
    acorr(:,i)=calc_corr(mskip(i,:),laglen);
    subplot(2,1,i);
    plot([0 laglen],[0 0],'Color',[0.7 0.7 0.7],'LineWidth',3); hold on
    plot(lags(laglen+1:10:laglen*2+1),acorr(laglen+1:10:laglen*2+1,i),'bo');
    plot(lags(laglen+1:10:laglen*2+1),acorrn(laglen+1:10:laglen*2+1,i),'ro');
    legend('','thinned', 'non thinned',Location='best')
    hold off
    if i == 1
        ylabel(['A (S)'])
        title('Autocorrelation of thinned parameter samples ')
    else
        ylabel(['A (T)'])
    end
end

```

```

if i~=2
    set(gca,'Xticklabel',[]);
end
xlabel('Lag')

```

(c) Estimate the 95% credible intervals by sorting the ensemble of parameter estimates, as is done in Exercise 11.4.

% estimate the 95% credible intervals

```

for i=1:2
    msort=sort(mskip(i,:));
    m2_5(i) = msort(round(2.5/100*length(mskip)));
    m97_5(i) = msort(round(97.5/100*length(mskip)));
    disp(['95% confidence interval for m', num2str(i), ' is [', num2str(m2_5(i)),',', num2str(m97_5(i)),']'])
end

```

(d) Estimate the Bayesian confidence intervals as we've done previously, using the standard deviation of the MCMC samples.

```

for i=1:2
    msort=sort(mskip(i,:)); Mmap = 1.96*std(msort);
    m97_5b = mMAP(i) + Mmap;
    m2_5b = mMAP(i) - Mmap;
    disp(['95% confidence interval for m', num2str(i), ' is [', num2str(m2_5b),',', num2str(m97_5b),']'])
end

```

(e) As in Figure 11.16, plot the sampled posterior distribution S vs T with the true model as a large black dot, and the MAP estimate with an open circle. On the same graph plot the thinned samples as gray dots and 95% credible intervals by a box.

%plot a scatter plot and histogram of the posterior distribution

mlims=[0.0018 0.0023; 0.52 0.64];

```

figure(3)
clf
for i=1:2
    for j=1:2
        if i==j
            continue
        else
            plot(mskip(j,:),mskip(i,:), 'k.', 'Markersize',6, 'Color',[0.6 0.6 0.6]);
            hold on
            % plot the true answer as a large black dot
            plot(mtrue(j),mtrue(i), 'k.', 'Markersize',24);
            % plot the accepted answers as gray dots
            plot(mMAP(j),mMAP(i), 'ko', 'Markersize',12, 'LineWidth',3);
            % plot the 95% ci as a box
            plot([m2_5(j),m97_5(j)], [m2_5(i),m2_5(i)], 'k-', 'LineWidth',1);
            plot([m2_5(j),m97_5(j)], [m97_5(i),m97_5(i)], 'k-', 'LineWidth',1);
            plot([m2_5(j),m2_5(j)], [m2_5(i),m97_5(i)], 'k-', 'LineWidth',1);
            plot([m97_5(j),m97_5(j)], [m2_5(i),m97_5(i)], 'k-', 'LineWidth',1);
            xlim(mlims(j,:));
            ylim(mlims(i,:));
        end
    end
end

```

```

        xlabel('S'); ylabel('T')

        hold off
    end
end
end

figure(4)
for i = 1:2
    subplot(2,1,i)
    qqplot(sort(mskip(i,:)))
    ylabel('Quantiles of Input Sample');
    xlabel('Standard Normal Quantiles');
    if i == 1
        title('Q-Q plot for S')
    else
        title('Q-Q plot for T')
    end
end
end

function l = Loglikelihood(m)
    global H;
    global TM;
    global SIGMA;
    global D;
    global Q;

    % Compute the standardized residuals.
    fvec = fun(m);

    % The log likelihood is  $(-1/2) \sum (fvec(i)^2, i=1..n)$ ;
    l =  $(-1/2) \sum (fvec.^2)$ ;
end

function lp = Logprior(m)
    if (m(1)>=0) && (m(1)<=0.01) && (m(2)>=0) && (m(2)<=2)
        lp=0;
    else
        lp=-Inf;
    end
end

function y=Generate(x)
    global stepsize
    y=x+stepsize.*randn(2,1);
end

function lr=Logproposal(x,y)
    global stepsize

```

```
    lr=(-1/2)*sum((x-y).^2./stepsize.^2);  
end
```