```
function nonrevMFI2DM(theta,plotRun,calcIndicators,n)
%% Non-reversible 2D Mean Field Ising model

% -------------------------------------------------------------------------
%% Initialise Variables

%Check number of input arguments
if nargin < 1
    theta = 0.01;
    plotRun = 0;
    calcIndicators = 0;
    n = 5000;
end

if nargin < 2
    plotRun = 0;
    calcIndicators = 0;
    n = 5000;
end

if nargin < 3
    calcIndicators = 0;
    n = 5000;
end

if nargin < 4
    n = 5000;
end

%Number of samples
%n = 5000;

%Samples (#,value/dim,direction)
samples = zeros(n,1,2);

%Number of spins (quadratic number)
%N = 64;
%N = 225;
N = 900;
%N = 2500;
NN = sqrt(N);

%Temperatur and inverse temperatur
T = 2/log(1 + sqrt(2));
beta = 1/T;

%Exchange energy
J = 1;

%Theta
%theta = 0.001;

%Magnetization
%M = 0;
%M = randsrc*2*(randi(N + 1) - 1);
M = randsrc*2*(randi(N/NN + 1) - 1);
samples(1,:,1) = M;

%Direction
dir = randsrc;
```

```
samples(1,:,2) = dir;

%Accepted samples in step A
accepted = 0;

%Prepare true distribution
x = -N:2:N;
yy = zeros(1,N+1);
for i = 1:N+1
    m = x(i);
    factor = exp(gammaln(N+1) - (gammaln(-m/2 + N/2 + 1) + gammaln(m/2 + N/2 + 1)));
    yy(i) = factor*exp(beta*J*m^2/(2*N));
end
z = sum(yy);
yy = yy/z;

%Prepare checkerboard
if M == 0
   C = ones(1,N);
   B = randperm(numel(C));
   C(B(1:N/2)) = -1;
   A = reshape(C,NN,NN);
else
   m = N/2 - sign(M)*M/2;
   C = sign(M)*ones(1,N);
   B = randperm(numel(C));
   C(B(1:m)) = -sign(M);
   A = reshape(C,NN,NN);
end

%Prepare figure
close all;
figure;
set(gcf, 'Position', get(0,'Screensize'));


% ------------------------------------------------------------------------
%% Calculate Samples and Draw Checkerboard

for i = 2:n

    %Step A ----------------------------------------
    %Read last step
    M = samples(i-1,:,1);
    dir = samples(i-1,:,2);
    proposal = M + 2*dir;

    %Acceptance probability
    factor = (N - dir*M)/(N + dir*M + 2);
    acceptance = factor*exp(2*beta*J*(dir*M + 1)/N);

    if rand < acceptance
             samples(i,:,1) =  proposal;
             samples(i,:,2) = -dir;
             accepted = accepted + 1;
             acc = 1;
         else
             samples(i,:,:) = samples(i-1,:,:);
             acc = 0;
    end
```

```
%end step A

%Step B ----------------------------------------
if rand < 1 - theta
    %Keep direction
    samples(i,:,2) = -samples(i,:,2);
end
%end step B

%Update checkerboard ----------------------------
if acc

A = A(:);
k = 1; u1 = 1; u2 = 1; look1 = randsrc; look2 = randsrc;
if look2 == 1; look2 = NN;
else look2 = -NN; end

%Read move
move = (proposal - M)/2;

%Draw move
while k <= 3

    %Propose location
    rr = randi(N);

    %Correct proposed location
    while A(rr) == move && u1 < 3 && u2 < 3

        %move horizontally
        if rand < 0.5
            if rr == N && look1 == 1
               look1 = -1;
               u1 = u1 + 1;
            elseif rr == 1 && look1 == -1
               look1 = 1;
               u1 = u1 + 1;
            else
               rr = rr + look1;
            end

        %move vertically
        else
            if rr > N - NN && look2 == NN
                look2 = -NN;
                u2 = u2 + 1;
            elseif rr <= NN && look2 == -NN
                look2 = NN;
                u2 = u2 + 1;
            else
                rr = rr + look2;
            end
        end %horizontal, vertical

    end %correct

    %Make touchy move
    if rr == N || rr == 1
        A(rr) = move;
        break;
```

```
        elseif A(rr+1) == move || A(rr-1) == move
            A(rr) = move;
            break;
        elseif k == 3
            A(rr) = move;
            break;
        end

        k = k + 1;

    end %while

    A = reshape(A,NN,NN);

    end %if
    %end update checkerboard

    %Plot -------------------------------------------
    if plotRun == 1

    %Clear current figure
    clf;

    %Plot samples
    subplot(2,1,1);
    title('Sample Histogramm');
    grid off;
    xlim([-ceil(N/sqrt(NN)) ceil(N/sqrt(NN))]);
    set(gca, 'XTickLabel',[],'YTickLabel',[]);
    xlabel('Magnetization');
    ylabel('Probability');
    hold('on');
    y = samples(1:i,:,1);
    a = histc(y,x);
    zz = sum(a);
    bar(x,a/zz,'b','EdgeColor',[0 0 0.6])
    plot(x,yy,'g','LineWidth',3);
    legend('Samples','True Distribution');

    %Plot checkerboard
    subplot(2,1,2);
    title('Checkerboard');
    set(gca, 'XTickLabel',[],'YTickLabel',[]);
    hold('on');
    colormap(gray);
    imagesc(A);
    axis image;

    %Pause
    tic; while toc < 0.00001; end
    drawnow;

    end %if
    %end plot

end %for
%end sampling

% ---------------------------------------------------------------------
%% Output after Calculation
```

```
if plotRun == 0

    %Plot samples
    subplot(2,1,1);
    title('Sample Histogramm');
    grid off;
    xlim([-N/ceil(sqrt(NN)) N/ceil(sqrt(NN))]);
    xlabel('Magnetization');
    ylabel('Probability');
    hold('on');
    y = samples(:,:,1);
    a = histc(y,x);
    zz = sum(a);
    bar(x,a/zz,'b','EdgeColor',[0 0 0.6]);
    plot(x,yy,'g','LineWidth',3);
    legend('Samples','True Distribution');

    %Movement of magnetization
    subplot(2,1,2);
    title('Movement');
    ylabel('Magnetization');
    xlabel('Samples');
    hold('on');
    y = y(1:end);
    plot(1:length(y),y,'r','LineWidth',1);

end %if
%end plot

%Display indicators
if calcIndicators

   samples = y;
   L = 100;
   m = mean(samples);
   v = cov(samples);
   autocorrs = zeros(L,1);

   for i = 1:L
       autosum = 0;
       for j = 1:n-i
           autosum = autosum + (samples(j,:)-m)/(2*v)*(samples(j+i,:)-m)';
       end
       autocorrs(i) = autosum/(n-i-1);
    end

    INEFFICIENCY = 1+2*sum(autocorrs)
    ACCEPTANCERATE = accepted/n

end %indicators

end %main
```