sim_board.m

```
function [matrix_result, H, T_e, max_dT, dT_characteristic] =
 sim_board(matrix_in, sim_duration, K, Cw, ro, A, D, h, P, dt,
 Epsilon, debug)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here

Ke0 = 273;
T1 = Ke0 + 10;
T2 = Ke0 + 80;
T3 = Ke0 + 20;

dx = A/size(matrix_in,1);
dy = A/size(matrix_in,1);
dT_characteristic = [];

% preprocessing the out matrix
matrix_out = ones(size(matrix_in,1),size(matrix_in,1)).*T3;

for i = 1 : size(matrix_in,1)
    for j = 1 : size(matrix_in,1)
        if matrix_in(i,j) == 0
            matrix_out(i,j) = Ke0;
            continue;
        end
        if matrix_in(i,j) == 1
            matrix_out(i,j) = T1;
            continue;
        end
        if matrix_in(i,j) == 8
            matrix_out(i,j) = T1;
            continue;
        end
    end
end

% simulation main loop
for H = 1 : dt : sim_duration
    max_dT = 0;
    for i = 1 : size(matrix_in,1)
        for j = 1 : size(matrix_in,1)

            if matrix_in(i,j) == 0
                matrix_out(i,j) = Ke0;
                continue;
            end

            if matrix_in(i,j) == 1
                matrix_out(i,j) = T1;
                continue;
            end
```

```
            if matrix_in(i,j) == 8
                matrix_out(i,j) = matrix_out(i,j) + P*dt/
(Cw*D^2*h*ro);
            end

            dT_x = ((K*dt)/(Cw*ro*(dx^2)))*(matrix_out(i
+1,j)-2*matrix_out(i,j)+matrix_out(i-1,j));
            dT_y = ((K*dt)/(Cw*ro*(dy^2)))*(matrix_out(i,j
+1)-2*matrix_out(i,j)+matrix_out(i,j-1));

            if(abs(dT_x + dT_y) > max_dT && matrix_in(i,j) ~= 8)
                max_dT = abs(dT_x + dT_y);
            end

            matrix_out(i,j) = matrix_out(i,j) + dT_x;
            matrix_out(i,j) = matrix_out(i,j) + dT_y;
        end
    end

    dT_characteristic = [dT_characteristic max_dT];

    % STOP
    if(max_dT/dt < Epsilon && H > 1)
        T_e = 0;
        cells = 0;
        for ii = 1 : size(matrix_in,1)
            for jj = 1 : size(matrix_in,1)
                if matrix_in(ii,jj) == 0
                    continue;
                end
                cells = cells +1;
                T_e = T_e + matrix_out(ii,jj);
            end
        end
        T_e = T_e/cells;
        matrix_result = matrix_out;
        return
    end

    % debug %
    if((mod(H,100) == 0 || H == 0) && debug == 1)
        %pause(1)
        figure(123)
        subplot(2,1,1)
        imagesc(matrix_out);
        title(H)

        subplot(2,1,2)
        plot(dT_characteristic)
        title("max dT")
    end
end
T_e = -1;
```

```
H = sim_duration;
matrix_result = matrix_out;
end
```

Published with MATLAB® R2020b