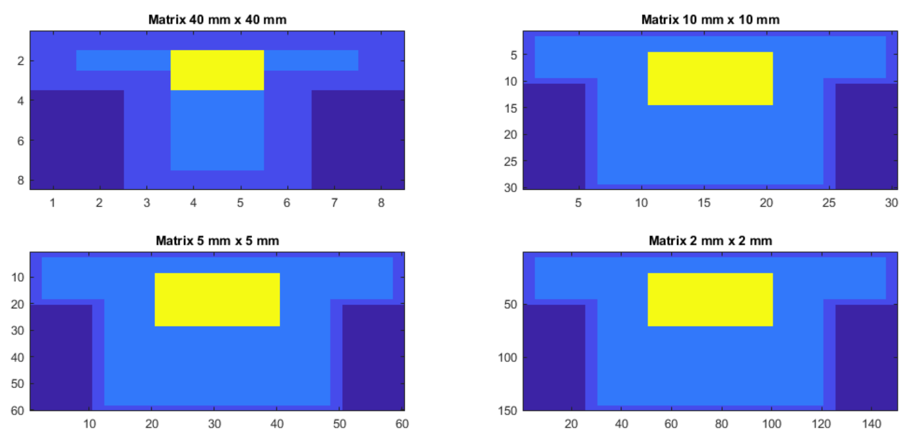
Heat transfer simulation
Wojciech Dziuba

Table of Contents

Creating matrices that represent the simulation resolution

```
clc; close all;

matrix_4x4 = [
    1 1 1 1 1 1 1 1;
    1 2 2 8 8 2 2 1;
    1 1 1 8 8 1 1 1;
    0 0 1 2 2 1 0 0;
    0 0 1 2 2 1 0 0;
    0 0 1 2 2 1 0 0;
    0 0 1 2 2 1 0 0;
    0 0 1 1 1 1 0 0;
];

matrix_1x1 = [
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
    1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 2 2 2 2 2 1;
    1 2 2 2 2 2 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 2 2 2 2 2 1;
    1 1 1 1 1 1 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 1 1 1 1 1 1;
    0 0 0 0 0 1 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 8 8 8 8 8 8 8 8 8 8 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
```

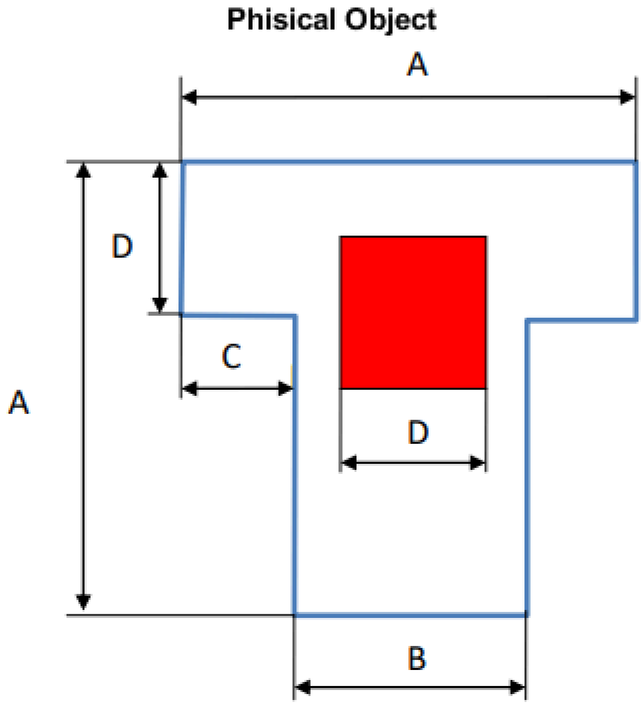
```
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 0 0 0 0 0;
    0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0;
];

matrix_05x05 = repelem(matrix_1x1,2,2);
matrix_02x02 = repelem(matrix_1x1,5,5);


figure('name', 'Resolution Matrices', 'Position', [300 300 1200 500])
subplot(2,2,1)
imagesc(matrix_4x4);
title("Matrix 40 mm x 40 mm")
subplot(2,2,2)
imagesc(matrix_1x1);
title("Matrix 10 mm x 10 mm")
subplot(2,2,3)
imagesc(matrix_05x05);
title("Matrix 5 mm x 5 mm")
subplot(2,2,4)
imagesc(matrix_02x02);
title("Matrix 2 mm x 2 mm")
```



Phisical & Simulation parameters of diffrent materials

```
% Stainles Steel
ss_K = 58;
ss_Cw = 450;
```

```
ss_ro = 7860;
ss_Epsilon = 0.001;

% Coper
cu_K = 401;
cu_Cw = 380;
cu_ro = 8920;
cu_Epsilon = 0.001;

% Aluminum
al_K = 237;
al_Cw = 900;
al_ro = 2700;
al_Epsilon = 0.001;

figure('name', 'Object', 'Position', [300 300 1200 500])
imshow("Przechwytywanie.PNG")
title("Phisical Object")

A = 0.3;
N = 0.2;
C = 0.05;
D = 0.1;
h = 0.005;
P = 100;

dt = 0.01;
sim_duration = 10000;
```



Running the simulation

```
% Stainless Steel
fprintf("steel...");
[ss_4x4_out,   ss_4x4_H,   ss_4x4_T_e,   ss_max_4x4_dT]    =
 sim_board(matrix_4x4,     sim_duration, ss_K, ss_Cw, ss_ro, A, D, h,
 P, dt, ss_Epsilon, 0);
[ss_1x1_out,   ss_1x1_H,   ss_1x1_T_e,   ss_max_1x1_dT]    =
 sim_board(matrix_1x1,     sim_duration, ss_K, ss_Cw, ss_ro, A, D, h,
 P, dt, ss_Epsilon, 0);
[ss_05x05_out, ss_05x05_H, ss_05x05_T_e, ss_max_05x05_dT]  =
 sim_board(matrix_05x05,   sim_duration, ss_K, ss_Cw, ss_ro, A, D, h,
 P, dt, ss_Epsilon, 0);
[ss_02x02_out, ss_02x02_H, ss_02x02_T_e, ss_max_02x02_dT]  =
 sim_board(matrix_02x02,   sim_duration, ss_K, ss_Cw, ss_ro, A, D, h,
 P, dt, ss_Epsilon, 0);
fprintf("done\n");

% Cooper
fprintf("cooper...");
[cu_4x4_out,   cu_4x4_H,   cu_4x4_T_e,   cu_max_4x4_dT]    =
 sim_board(matrix_4x4,     sim_duration, cu_K, cu_Cw, cu_ro, A, D, h,
 P, dt, cu_Epsilon, 0);
```

```
[cu_1x1_out,   cu_1x1_H,   cu_1x1_T_e,   cu_max_1x1_dT]    =
 sim_board(matrix_1x1,     sim_duration, cu_K, cu_Cw, cu_ro, A, D, h,
 P, dt, cu_Epsilon, 0);
[cu_05x05_out, cu_05x05_H, cu_05x05_T_e, cu_max_05x05_dT]  =
 sim_board(matrix_05x05,   sim_duration, cu_K, cu_Cw, cu_ro, A, D, h,
 P, dt, cu_Epsilon, 0);
[cu_02x02_out, cu_02x02_H, cu_02x02_T_e, cu_max_02x02_dT]  =
 sim_board(matrix_02x02,   sim_duration, cu_K, cu_Cw, cu_ro, A, D, h,
 P, dt/10, cu_Epsilon, 0);
fprintf("done\n");

% Aluminum
fprintf("aluminum...");
[al_4x4_out,   al_4x4_H,   al_4x4_T_e,   al_max_4x4_dT]    =
 sim_board(matrix_4x4,     sim_duration, al_K, al_Cw, al_ro, A, D, h,
 P, dt, al_Epsilon, 0);
[al_1x1_out,   al_1x1_H,   al_1x1_T_e,   al_max_1x1_dT]    =
 sim_board(matrix_1x1,     sim_duration, al_K, al_Cw, al_ro, A, D, h,
 P, dt, al_Epsilon, 0);
[al_05x05_out, al_05x05_H, al_05x05_T_e, al_max_05x05_dT]  =
 sim_board(matrix_05x05,   sim_duration, al_K, al_Cw, al_ro, A, D, h,
 P, dt, al_Epsilon, 0);
[al_02x02_out, al_02x02_H, al_02x02_T_e, al_max_02x02_dT]  =
 sim_board(matrix_02x02,   sim_duration, al_K, al_Cw, al_ro, A, D, h,
 P, dt/10, al_Epsilon, 0);
fprintf("done\n");
```

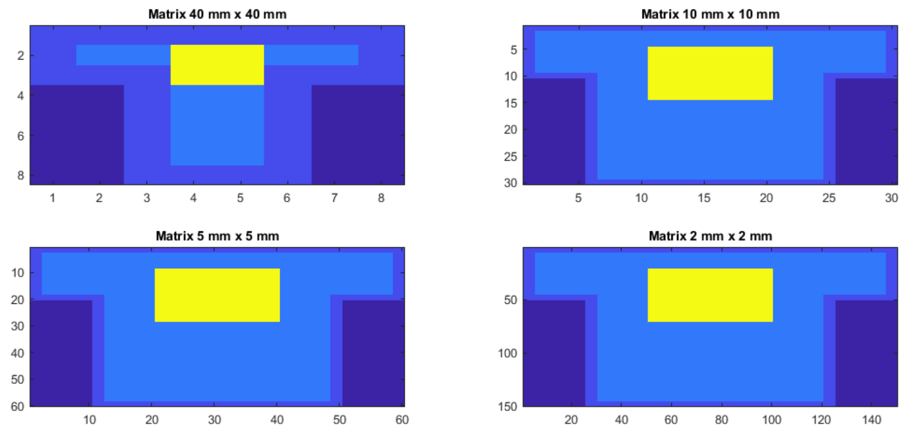
steel...done
cooper...done
aluminum...done

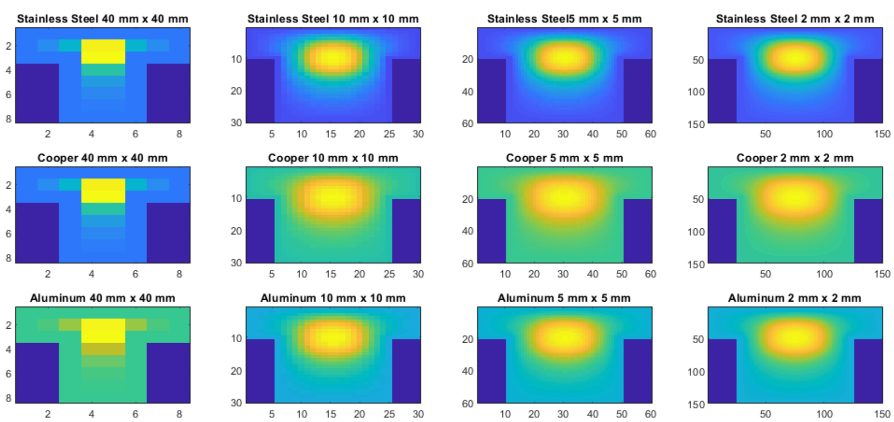
Acquired Data

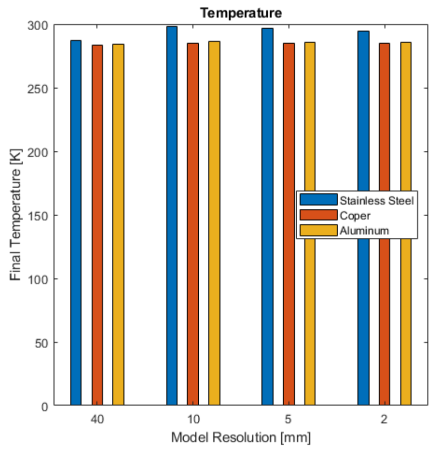
```
figure('name', 'Resolution Matrices', 'Position', [300 300 1200 500])
subplot(3,4,1)
imagesc(ss_4x4_out);
title("Stainless Steel 40 mm x 40 mm")
subplot(3,4,2)
imagesc(ss_1x1_out);
title("Stainless Steel 10 mm x 10 mm")
subplot(3,4,3)
imagesc(ss_05x05_out);
title("Stainless Steel5 mm x 5 mm")
subplot(3,4,4)
imagesc(ss_02x02_out);
title("Stainless Steel 2 mm x 2 mm")

subplot(3,4,5)
imagesc(ss_4x4_out);
title("Cooper 40 mm x 40 mm")
subplot(3,4,6)
imagesc(cu_1x1_out);
title("Cooper 10 mm x 10 mm")
```

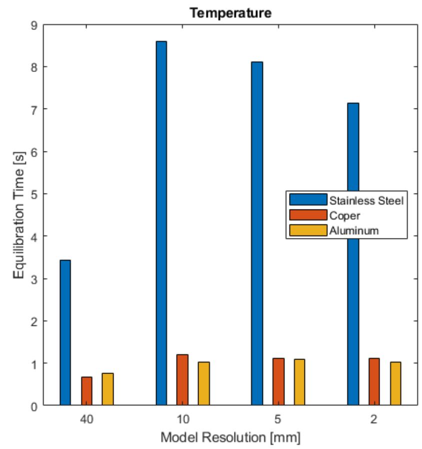
```
subplot(3,4,7)
imagesc(cu_05x05_out);
title("Cooper 5 mm x 5 mm")
subplot(3,4,8)
imagesc(cu_02x02_out);
title("Cooper 2 mm x 2 mm")

subplot(3,4,9)
imagesc(al_4x4_out);
title("Aluminum 40 mm x 40 mm")
subplot(3,4,10)
imagesc(al_1x1_out);
title("Aluminum 10 mm x 10 mm")
subplot(3,4,11)
imagesc(al_05x05_out);
title("Aluminum 5 mm x 5 mm")
subplot(3,4,12)
imagesc(al_02x02_out);
title("Aluminum 2 mm x 2 mm")


T_e_data = [
    ss_4x4_T_e   cu_4x4_T_e   al_4x4_T_e;
    ss_1x1_T_e   cu_1x1_T_e   al_1x1_T_e;
    ss_05x05_T_e cu_05x05_T_e al_05x05_T_e;
    ss_02x02_T_e cu_02x02_T_e al_02x02_T_e;
   ];

H_data = [
    ss_4x4_H   cu_4x4_H   al_4x4_H;
    ss_1x1_H   cu_1x1_H   al_1x1_H;
    ss_05x05_H cu_05x05_H al_05x05_H;
    ss_02x02_H cu_02x02_H al_02x02_H;
   ];

figure('name', 'Simulation Data', 'Position', [300 300 1200 500])
subplot(1,2,1)
bar(T_e_data, 0.5)
ax = gca;
%ax.XTickLabel = {'SS','Cu', 'Al'};
ax.XTickLabel = {'40','10', '5', '2'};
%xlabel('Plate Material')
xlabel('Model Resolution [mm]')
ylabel('Final Temperature [K]')
title("Temperature")
legend('Stainless Steel','Coper', 'Aluminum', 'Location', 'east')

subplot(1,2,2)
bar(H_data.*dt, 0.5)
ax = gca;
%ax.XTickLabel = {'SS','Cu', 'Al'};
ax.XTickLabel = {'40','10', '5', '2'};
%xlabel('Plate Material')
xlabel('Model Resolution [mm]')
```

```
ylabel('Equilibration Time [s]')
title("Temperature")
legend('Stainless Steel','Coper', 'Aluminum', 'Location', 'east')

figure('name', 'Table', 'Position', [300 300 300 100])
resolution = {'40 mm';'10 mm';'5 mm'; '2 mm'};
Stainless_Steel =
 [T_e_data(1,1);T_e_data(2,1);T_e_data(3,1);T_e_data(4,1)];
Coper = [T_e_data(1,2);T_e_data(2,2);T_e_data(3,2);T_e_data(4,2)];
Aluminum = [T_e_data(1,3);T_e_data(2,3);T_e_data(3,3);T_e_data(4,3)];
T = table(Stainless_Steel,Coper,Aluminum,'RowNames',resolution);

uitable('Data',T{:,:},'ColumnName',T.Properties.VariableNames,...
    'RowName',T.Properties.RowNames,'Units', 'Normalized', 'Position',
[0, 0, 1, 1]);
```









	Stainless_Steel	Coper	Aluminum
40 mm	286.8484	283.5595	283.9460
10 mm	297.9243	285.0850	286.5514
5 mm	296.7268	284.6650	285.9292
2 mm	294.3228	284.7398	286.0091

Summary

With that we can observe, that the minimum resolution right for the simulation was the 1 mm x 1 mm. Any better resolution yealds the same information about the equilibrium temperature of the plate

Published with MATLAB® R2020b

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