EE 5780 Advanced VLSI CAD Project II Report

# Thermal Analysis Problem in 2D Solution

Chentao Zhao 4/25/2014

# 1. Problem description

The thermal problem of a 2D plane can be coarsely modeled as N x N elements matrix. For project II, I need to solve the 2D thermal analysis problem illustrated as figure1. Elements in the matrix connect with each other with the heat resistance value of 0.1 (Celsius per Watt). We assume 3 sides out of 4 isolate from heat (the West, North and South boundary), the other one (the east boundary) is connecting with a constant temperature of 20 Celsius.  We try to figure out the temperature feature of the whole plane if we apply some heat sources onto any of the elements.

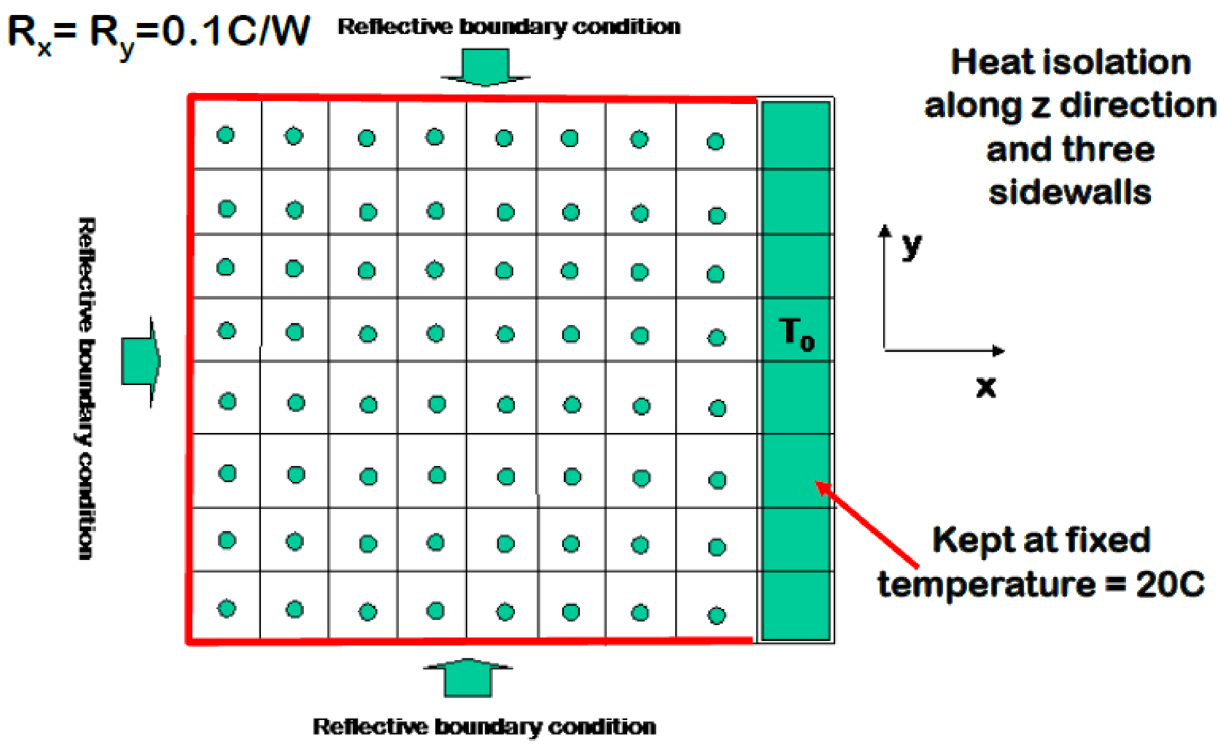


Figure 1: 2D thermal problem

# 2. Problem analysis

Law in the Heat problem can be comprehended as the similar circuit Ohm's law, that is:

***Temperature (Celsius) = Heat resistance(C/W) x Heat power (Watt)***

It is reasonable to assume that the heat problem can also be solved by the linear matrix form with the similar way we solve circuit problem. For circuit problem, especially those of all resistance grid, we can stamp circuit elements using particular stamping strategy. Then the problem can be solved with the proper matrix solver.

For a 1D heat problem, obviously the elements resistance or consistence can be stamping to the matrix using the similar stamping law of the circuit problem. However, for 2D heat problem, the stamping strategy is not so clear.

## 2.1 Detail and Key point

First of all, we need to refine the key problem points from the details.

For a heat conduction issue, we need the heat conduction or dissipation equation as the principle key; we also notice this project only concerns about the static temperature solution of the elements. Specific simplified equation would be important for summarizing the stamping strategy.

For elements dimension, we have 8x8 elements temperature, which means we have 64 unknowns. If we choose vector for these temperature, we will end up with 64 by 64 matrix as the conductance matrix. The results would probably be something looks like:

***G x T = I;      where G is 64 x64 matrix; T and I are 64x1 vectors.***

Next, how to arrange the elements into these Matrix and vectors is also important issue. A particular stamping strategy would be the last issue to solve this problem.

## 2.2 Problem solution

We solve this problem just as the sequence of this key points mentioned above.

### *2.2.1 Equation derivation issues*

Heat conduction has been studied and PDE equation are introduced as:

$$\rho C_p \frac{\partial T(x,y,z,t)}{\partial t} = \nabla[\kappa \nabla T(x,y,z,t)] + p(x,y,z,t)$$

This equation describes the law when heat dissipating process in general 3D space. The good news is we can simplify the equation with the given condition in our problem. A simpler equation can be easily obtained since this problem only involve the 2D dissipating process issue:

$$\rho C_p \frac{\partial T(x,y,t)}{\partial t} = \nabla[\kappa \nabla T(x,y,t)] + p(x,y,t)$$

Another important key is only the static solution concerned, which means the temperature state value will not be the function of time. Thus the time derivative terms can be replaced with zero:

$$0 = \nabla[\kappa \nabla T(x,y)] + p(x,y)$$

The final step of simplification is discretizing the equation for finite elements:

$$p(x,y) = G_x\big(T(x,y) - T(x-1,y)\big) + G_x\big(T(x,y) - T(x+1,y)\big)$$

$$+G_y\big(T(x,y) - T(x,y-1)\big) + G_y\big(T(x,y) - T(x,y+1)\big);$$

Where (x, y) is the position coordinates of the elements in the plane. $G_x$ and $G_y$ are the elements' heat conductance in the horizon and vertical directions: They are all given as heat resistance form; representing the physical values of $G_x = \kappa \times \frac{\Delta y}{\Delta x}$ and $G_y = \kappa \times \frac{\Delta x}{\Delta y}$ separately.

## *2.2.2 Boundary issues*

Next key issue is the boundary conditions. We have four boundaries: East boundaries (8 elements) connects with 20 Celsius constant temperature; the other three boundaries are isolated from heat transfer.

For the constant temperature boundary, we can use the similar stamping way as we stamping the voltage source in the circuit problem; A 'hybrid' I vector will be generate with some temperature rows at the bottom of the vector and several '1' elements will be stamped into the conductance matrix G.

The other three boundaries conditions states are not as clear as the east one. Here reflective boundaries means, all the heat transferred from the boundaries elements will be reflected without any absorption; as a consequence, we can guess any 'dissipating terms' of the heat conductance equation directed to the reflective boundaries will be zero as demonstrated in figure2; here 'dissipating terms' means the terms determine the conductance behavior of a certain direction of dissipation.

$$p(x,y) = G_x\big(T(x,y) - T(x-1,y)\big) + G_x\big(T(x,y) - T(x+1,y)\big)$$

Dissipating term 3

$$+G_y\big(T(x,y) - T(x,y-1)\big) + G_y\big(T(x,y) - T(x,y+1)\big);$$

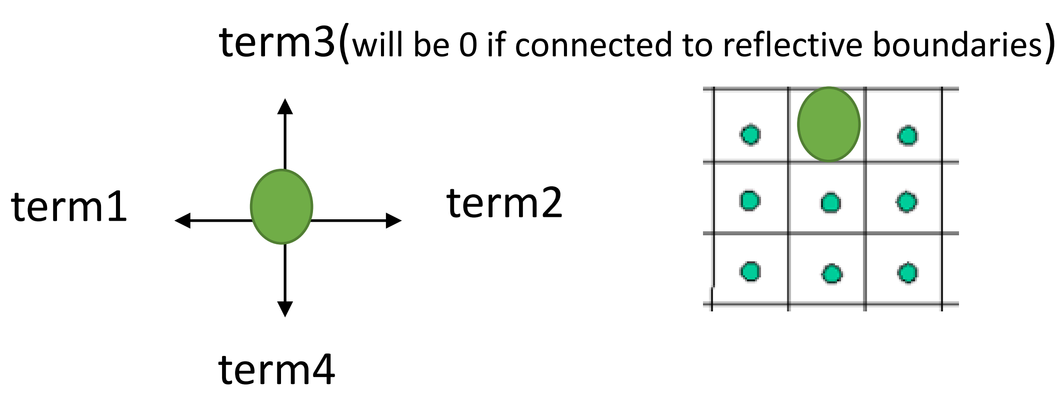


Figure 2: typical element with four dissipating terms

The other way we treat this reflective boundaries is that they are effective as the same temperature of the elements they attach; same temperature can also remove the corresponding dissipating terms in the equation.

### *2.2.3 Stamping strategy*

The fact that four terms exist in each equation of math and four conductive directions exist during physics heat dissipation indicate we will have complex conduction matrix G compared with 1D problem.

Before stamping strategy, one important rules should be mentioned first: We need to stamp the elements temperature vector *T* as **the column by column sequence** rather than **row by row sequence** as below illustration figure:

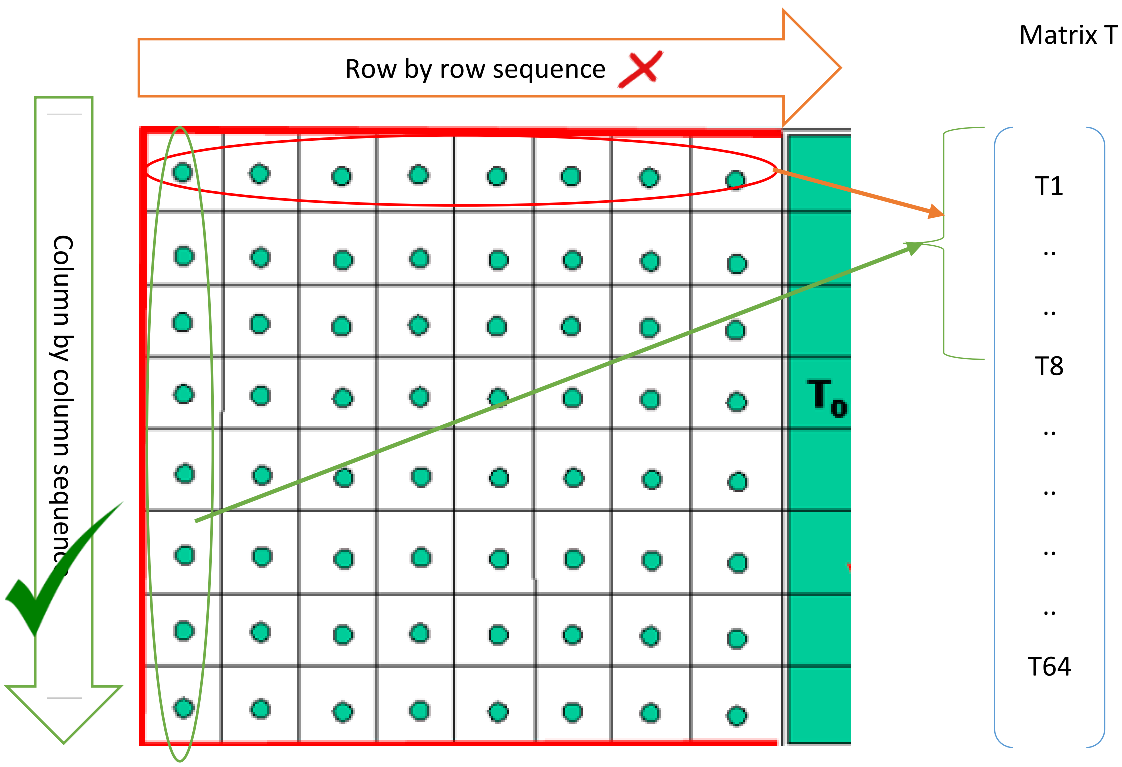


Figure 3: Row sequence stamping other than column sequence stamping

The reason is the boundaries issue and the object that we prefer to put the last boundary **column** at the bottom of the matrix. Failing to do that will probably lead to the irregular matrix form which is intersected by isolated constant temperature elements in our case.

Then our stamping strategy for this problem can be summarized as general steps as below:

1. We have 5 kinds of 8x8 basic submatrix (totally 64 submatrix), namely Matrix A,B,C,Z,D
2. Among them, submatrix Z is all zero matrix, D is unit matrix.
3. Matrix A B C are all diagonal matrix, they presents as something looks like:

$$A=\begin{pmatrix}
G_x+G_y & -G_y & \square & \square & \square & \square & \square & \square \\
-G_y & G_x+2G_y & -G_y & \square & \square & \square & \square & \square \\
\square & -G_y & G_x+2G_y & -G_y & \square & \square & \square & \square \\
\square & \square & -G_y & G_x+2G_y & -G_y & \square & \square & \square \\
\square & \square & \square & -G_y & G_x+2G_y & -G_y & \square & \square \\
\square & \square & \square & \square & -G_y & G_x+2G_y & -G_y & \square \\
\square & \square & \square & \square & \square & -G_y & G_x+2G_y & -G_y \\
\square & \square & \square & \square & \square & \square & -G_y & G_x+G_y
\end{pmatrix};$$

$$B=\begin{pmatrix}
2G_x+G_y & -G_y & \square & \square & \square & \square & \square & \square \\
-G_y & 2G_x+2G_y & -G_y & \square & \square & \square & \square & \square \\
\square & -G_y & 2G_x+2G_y & -G_y & \square & \square & \square & \square \\
\square & \square & -G_y & 2G_x+2G_y & -G_y & \square & \square & \square \\
\square & \square & \square & -G_y & 2G_x+2G_y & -G_y & \square & \square \\
\square & \square & \square & \square & -G_y & 2G_x+2G_y & -G_y & \square \\
\square & \square & \square & \square & \square & -G_y & 2G_x+2G_y & -G_y \\
\square & \square & \square & \square & \square & \square & -G_y & 2G_x+G_y
\end{pmatrix};$$

$$C=\begin{pmatrix}
-G_x & \square & \square & \square & \square & \square & \square & \square \\
\square & -G_x & \square & \square & \square & \square & \square & \square \\
\square & \square & -G_x & \square & \square & \square & \square & \square \\
\square & \square & \square & -G_x & \square & \square & \square & \square \\
\square & \square & \square & \square & -G_x & \square & \square & \square \\
\square & \square & \square & \square & \square & -G_x & \square & \square \\
\square & \square & \square & \square & \square & \square & -G_x & \square \\
\square & \square & \square & \square & \square & \square & \square & -G_x
\end{pmatrix};$$

Figure 4: stamping strategy of the submatrix

4. Then assemble them into G matrix as something looks like:

$$G=\begin{pmatrix} A & C & \square & \square & \square & \square & \square & \square \\ C & B & C & \square & \square & \square & \square & \square \\ \square & C & B & C & \square & \square & \square & \square \\ \square & \square & C & B & C & \square & \square & \square \\ \square & \square & \square & C & B & C & \square & \square \\ \square & \square & \square & \square & C & B & C & \square \\ \square & \square & \square & \square & \square & C & B & C \\ \square & \square & \square & \square & \square & \square & \square & D \end{pmatrix};$$

Figure 5: stamping strategy of assembling

For general case of similar boundary condition, more elements may be involved in the case. We just need to adjust the dimension of all these diagonal submatrix, with B and C submatrix diagonal addition and keeping the amount of A and D submatrix to be 1.

### *2.2.4 Matrix solving*

For simple 8 x 8 elements problem, the G matrix have the dimension of 64 by 64. This linear matrix equation can be solved by MATLAB built-in solver efficiently. Additional fast calculation issue will emerge if we have too many elements.

# 3. Result and Discussion

To present the heat dissipation data as images, we need to transfer the results of temperature vector into square matrix and show them in bitmap. Bitmap resolution is important if we want to present high resolution for the temperature difference.

We present the results under 4 different circumstances:

a) For 8x8 elements, no heat power is added;
b) For 8x8 elements, 45W heat power is applied on random elements, say: $20^{th}$ elements of column sequence;
c) For 8x8 elements, three random 500W heat powers are applied; say: $6^{th}$, $24^{th}$ and $53^{th}$ elements of column sequence;
d) For 25x25 elements, three random 500W heat powers are applied; say: $18^{th}$, $355^{th}$ and $516^{th}$ elements of column sequence;

## 3.1 Result a:

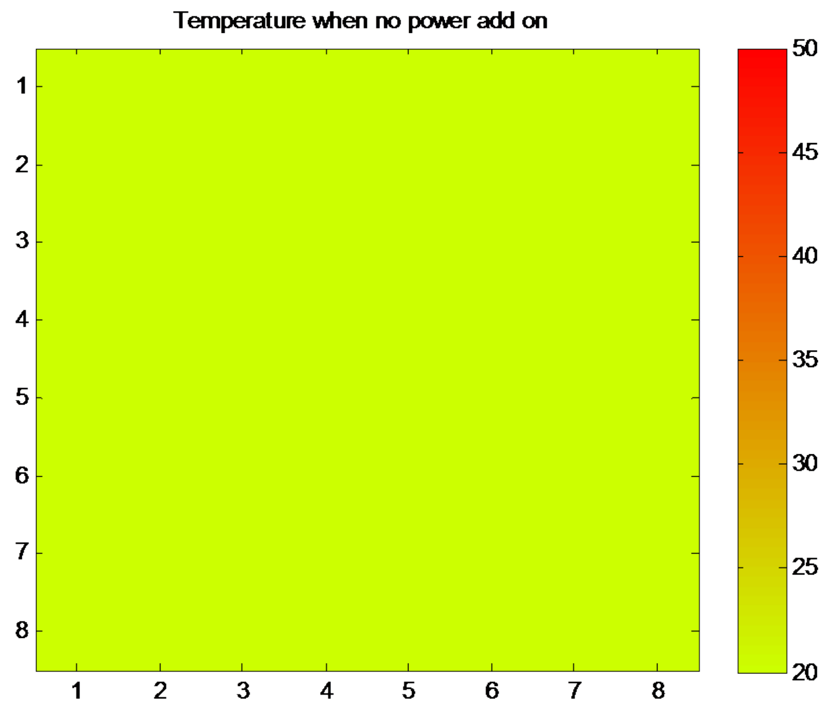


Figure 6: result a

In this case, the heat only comes from the constant 20 Celsius boundary; the results would make sense if only all the elements ended up with 20 Celsius.

## 3.2 Result b:

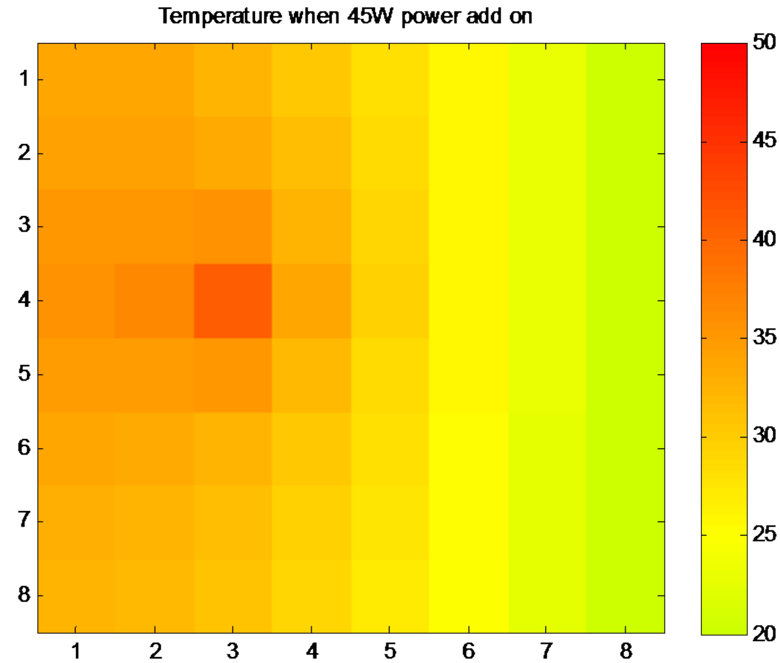


Figure 6: result b

When we apply a heat source on one element far from the east boundary, the heat will dissipate in 4 direction from this element. The coolest section would be the elements connecting to the constant temperature boundary.

## 3.3 Result c:

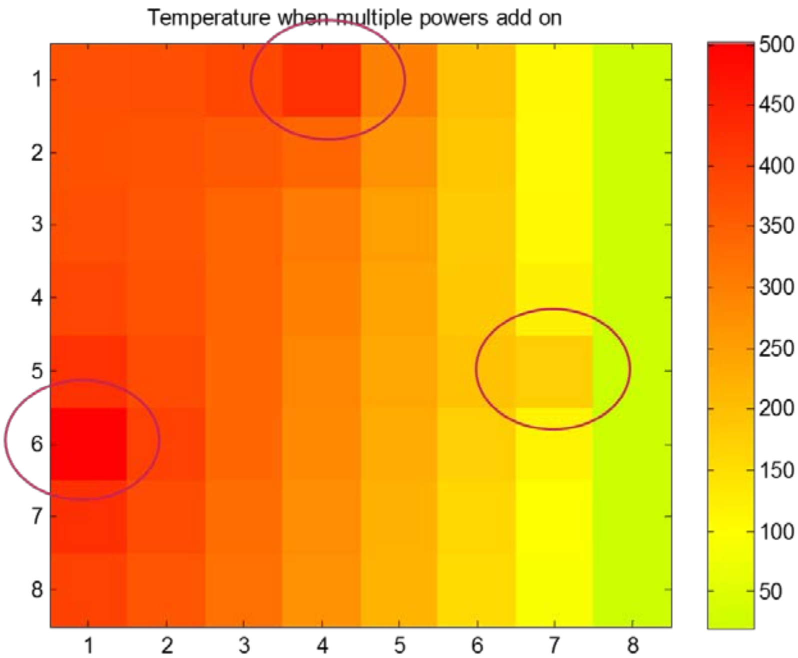


Figure 7: result c

Heat of the elements close to the constant temperature will be absorbed by the boundary; on the other way, the corner elements close to the reflective boundary will have higher temperature due to poor heat dissipation.

## 3.4 Result d:

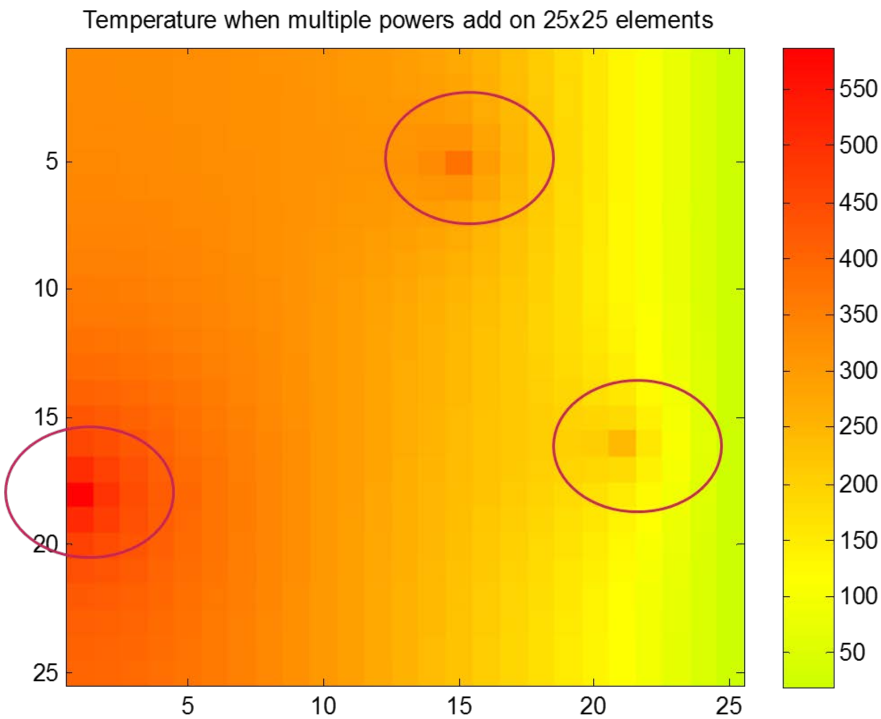


Figure 8: result d

For more elements problem, we have better resolution with the same color map. The result is similar to results c. we can see the corner heat source raise the temperature of the corner section compared with the corner has no heat source around.

# 4. Conclusion

In this project, problem-solving oriented method was applied to analyze the 2D thermal problem; General stamping strategy for similar thermal problem of the similar boundary condition was summarized.

We can have better ‘visual results’ if we have higher resolution heat maps compared with MATLAB built-in one; Even though I have adjusted the heat map by the heat map tools, the temperature difference of each elements still does not look sharp enough.

About the efficiency of the method. If we had parse and automatic stamping code, this method would be perfect for solving similar problem as long as:

1. All resistance type 2D thermal grid.

2. Similar type of boundary conditions

3. The element number is not as large as the limitation of the MATLAB matrix solver, otherwise we would take fast estimation algorithms issue into account.