# Spreadsheet based thermal resistance network analysis

Ross Wilcoxon

*This presentation presents an approach for filling the gap between simple one-dimensional thermal calculations and FEM/CFD analyses. Excel has substantial analysis capabilities and can also provide relatively simple user interfaces. The tutorial will discuss methods to estimate the thermal resistances within a system and how to automatically populate those values into a thermal resistance matrix. This thermal resistance matrix can then be solved using Excel matrix functions to determine system temperatures.*

# *Why do thermal analysis using a spreadsheet?*

- There are a lot of powerful software tools targeted at solving thermal problems.  They are so simple that any fool can run them and produce believable looking results.  So why think about using a spreadsheet?

- For starters, because any fool can run those tools and produce believable results...

# *Seriously, when should we use a spreadsheet?*

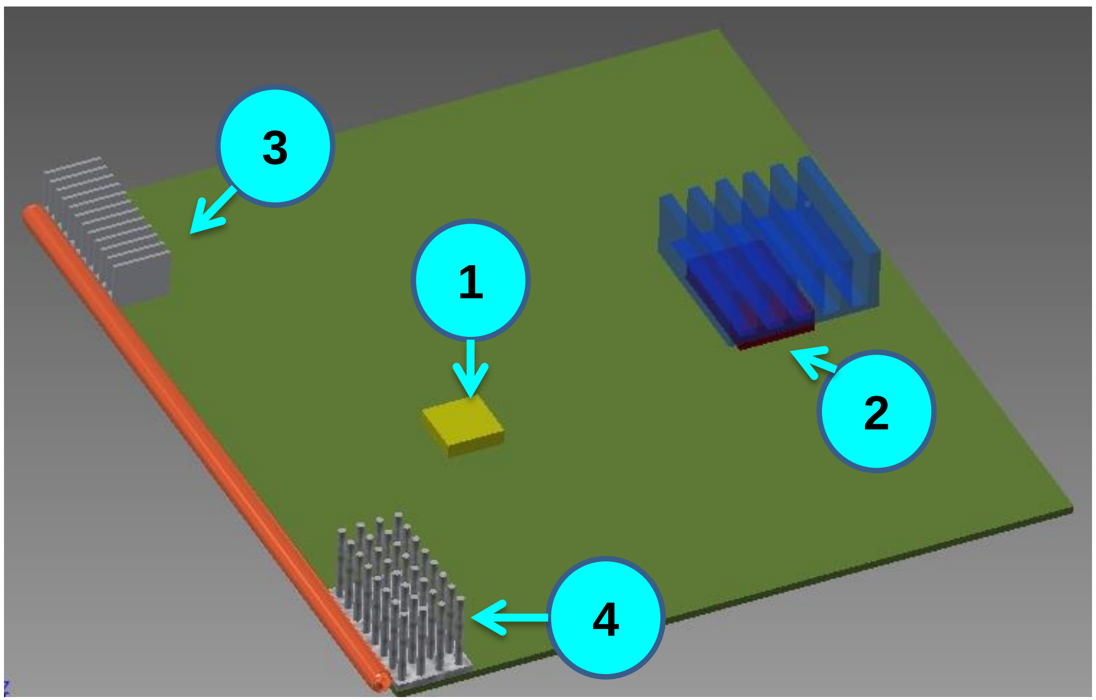
- Spreadsheet-based analysis may be good for:
  - quickly identifying critical design factors
  - early designs that are not well enough defined to generate solid models
  - creating tools that can be shared with other people who don't use dedicated analysis tools such as ANSYS, Flotherm, etc.
  - analyzing simple situations so that you aren't tying up licenses of dedicated analysis tools
  - running a quick validation analysis to make sure that you didn't make a mistake with your normal analysis tool
- Spreadsheets won't:
  - replace analysis tools like Flotherm, ANSYS, etc.
  - be unaffected by fat finger mistakes

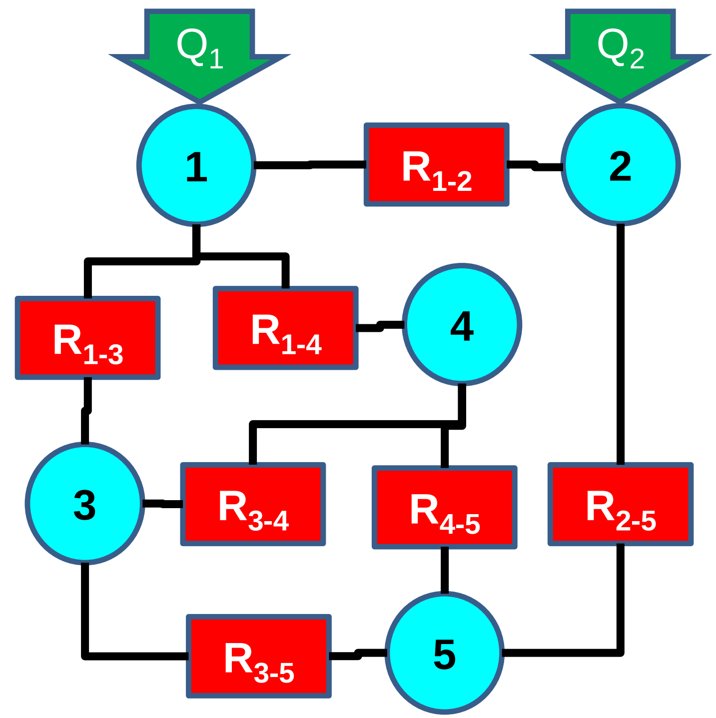
# *Resistance Network*

- We can model a thermal system as a network of individual nodes that are connected with thermal resistances (R)
- For this analysis, we will only look at steady state thermal conditions, so we won't include thermal capacitance (mass*specific heat)
- Thermal resistance determines the heat transfer (Q) between two nodes at different temperatures (T): $Q_{a\text{-}b} = (T_a - T_b)/R_{a\text{-}b}$

# *An Unrealistic Example of a Thermal Resistance Network*

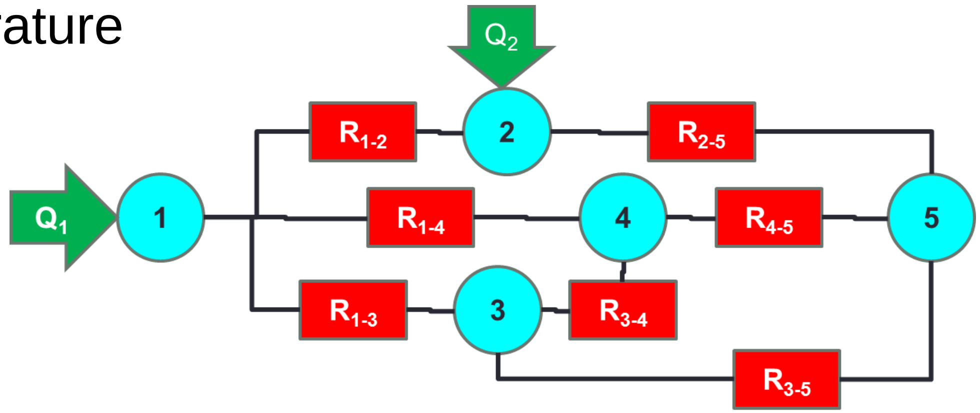
- Two heat dissipating components on a board with three heat sinks and a heat pipe
- Node 5 is the ambient air



# *Solving a Resistance Network*

- Energy balance applied to each node: energy in + energy out = 0 (steady state conditions)
- $T_5$ is the ambient temperature, which stays the same, so we don't need to do an energy balance on it
  - node 1: $Q_1 - (T_1 - T_2) / R_{1\text{-}2} - (T_1 - T_3) / R_{1\text{-}3} - (T_1 - T_4) / R_{1\text{-}4} = 0$
  - node 2: $Q_2 + (T_1 - T_2) / R_{1\text{-}2} - (T_2 - T_5) / R_{2\text{-}5} = 0$
  - node 3: $(T_1 - T_3) / R_{1\text{-}3} - (T_3 - T_4) / R_{3\text{-}4} - (T_3 - T_5) / R_{3\text{-}5} = 0$
  - node 4: $(T_1 - T_4) / R_{1\text{-}4} + (T_3 - T_4) / R_{3\text{-}4} - (T_4 - T_5) / C_{4\text{-}5} = 0$
  - node 5: $T_5$ = ambient temperature



# *Rearranging Terms*

- Rearranged equations and using conductance (C = 1/R) =>
  - $(C_{1\text{-}2}+C_{1\text{-}3}+C_{1\text{-}4})T_1 \; - \; C_{1\text{-}2}T_2 \; - \; C_{1\text{-}3}T_3) - C_{1\text{-}4}T_4 = Q_1$
  - $-C_{1\text{-}2}T_1 + (C_{1\text{-}2}+C_{2\text{-}5})T_2 = Q_2 + C_{2\text{-}5}T_5$
  - $-C_{1\text{-}3}T_1 + (C_{1\text{-}3}+C_{3\text{-}4}+C_{3\text{-}5})T_3 - C_{3\text{-}4}T_4 = C_{3\text{-}5}T_5$
  - $-C_{1\text{-}4}T_1 - C_{3\text{-}4}T_3 + (C_{1\text{-}4}+C_{3\text{-}4}+C_{4\text{-}5})T_4 = C_{4\text{-}5}T_5$

- In matrix notation, this is [C]{T} = [Q]

$$\begin{bmatrix} (C_{1-2}+C_{1-3}+C_{1-4}) & -C_{1-2} & -C_{1-3} & -C_{1-4} \\ -C_{1-2} & (C_{1-2}+C_{2-5}) & 0 & 0 \\ -C_{1-3} & 0 & (C_{1-3}+C_{3-4}+C_{3-5}) & -C_{3-4} \\ -C_{1-4} & 0 & -C_{3-4} & (C_{1-4}+C_{3-4}+C_{4-5}) \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 + C_{2-5}T_5 \\ C_{3-5}T_5 \\ C_{4-5}T_5 \end{bmatrix}$$

- Solve for each temperature using matrix algebra:
  - $\{T\} = [C]^{-1}[Q]$

# *Three steps to use a Resistance Network*

**Focus of this Presentation**

- Set up the network
  - define nodes
  - estimate thermal resistances between each node
- Convert the individual resistances and boundary conditions into matrices
  - the conductance matrix, [C], comes from the resistances
  - the boundary condition matrix, [Q], comes from the heat loads and the ambient temperature (and resistances that are connected to ambient)
- Apply matrix algebra to solve for temperatures

# *Step 1: Defining Thermal Resistances*

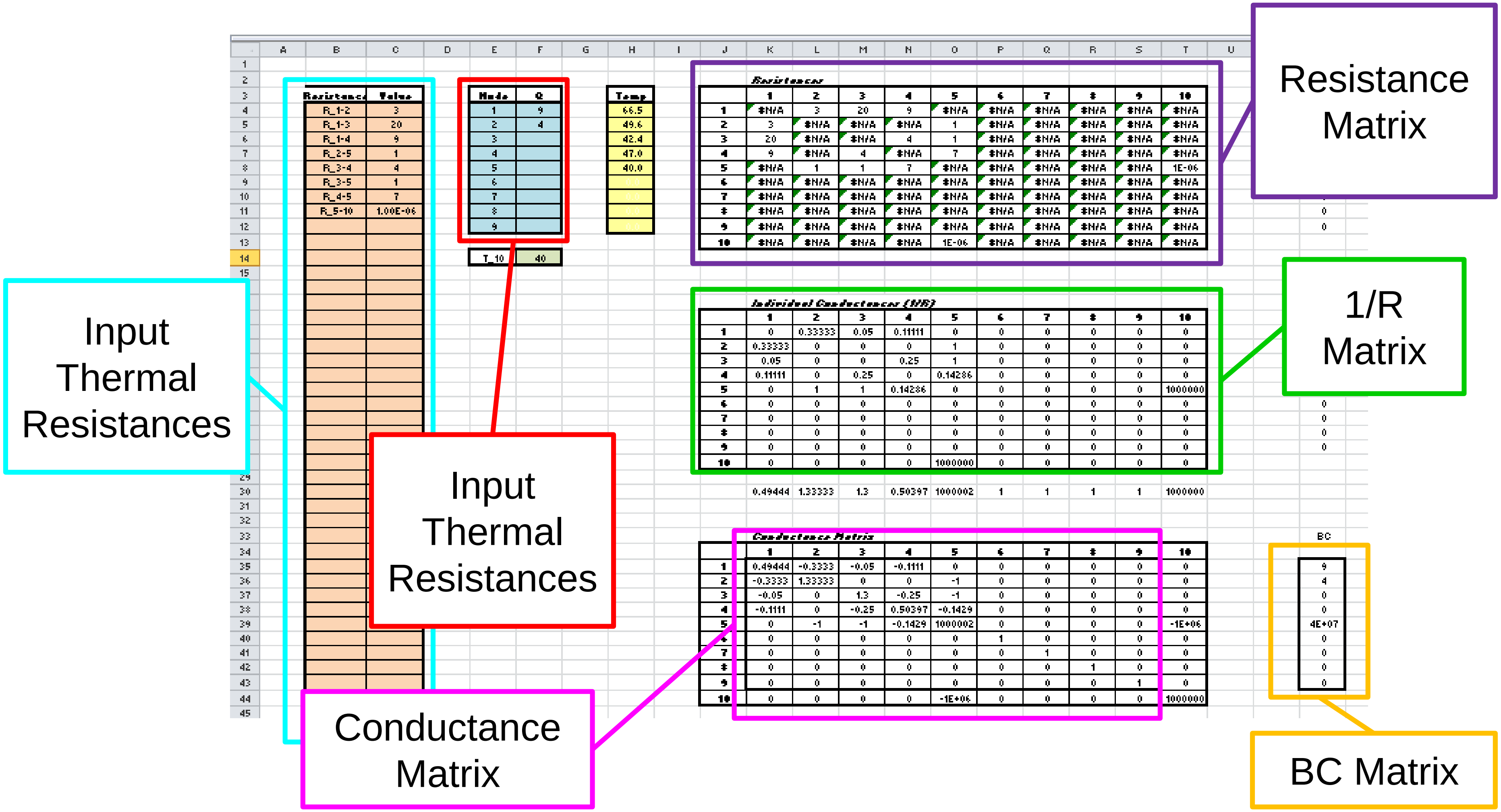
- Real systems are inherently 3-dimensional; thermal resistances are based on an assumption of 1-dimensional heat flow
- Types of thermal resistances
  - 1-D conduction; $R = L/kA$ for planar, $R = \ln(r_2/r_1)/2\pi Lk)$ for radial, etc.
  - uniform surface temperature convection ($R = 1/hA$)
  - interface resistance due to TIMs, contact resistance, etc. ($R = R^*/A$ )
    - $R^*$ = thermal impedance , such as C $cm^2$/ W
  - component resistance ($\theta_{j\text{-}a}$, $\theta_{j\text{-}b}$, etc.)
  - various methods to treat 2-D heat flow as 1-D
    - spreading resistance, ‘cooling circle’, etc.

- For this presentation, I am assuming that you can come up with the thermal resistance values that you need
  - spreadsheets do give you some flexibility for implementing these and modifying them by only changing a cell or two
  - you can create generic resistance tools on self contained worksheets

# *Step 2: Generating the Conductance and Boundary Condition Matrices*

- There are boatloads of ways to do this; I am just suggesting my way of doing it
- The goal here isn't elegance or computational efficiency
  - the goal is to make the best use of the capabilities of spreadsheet in order to make our lives easier!
- So, how do we let the spreadsheet do all the work to convert lists of thermal resistances and power inputs into the [C] and [Q] matrices?
  - first generate a resistance matrix
  - convert values in the resistance matrix to 1/R values
  - then generate a conductance matrix

# Generic Resistance Network Solver




| Resistance | Value |
|---|---|
| R_1-2 | 3 |
| R_1-3 | 20 |
| R_1-4 | 9 |
| R_2-5 | 1 |
| R_3-4 | 4 |
| R_3-5 | 1 |
| R_4-5 | 7 |
| R_5-10 | 1.00E-06 |

| Node | Q | Temp |
|---|---|---|
| 1 | 9 | 66.5 |
| 2 | 4 | 49.6 |
| 3 | | 42.4 |
| 4 | | 47.0 |
| 5 | | 40.0 |
| 6 | | 0.0 |
| 7 | | 0.0 |
| 8 | | 0.0 |
| 9 | | 0.0 |

| T_10 | 40 |
|---|---|

*Resistances*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | #N/A | 3 | 20 | 9 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| 2 | 3 | #N/A | #N/A | #N/A | 1 | #N/A | #N/A | #N/A | #N/A | #N/A |
| 3 | 20 | #N/A | #N/A | 4 | 1 | #N/A | #N/A | #N/A | #N/A | #N/A |
| 4 | 9 | #N/A | 4 | #N/A | 7 | #N/A | #N/A | #N/A | #N/A | #N/A |
| 5 | #N/A | 1 | 1 | 7 | #N/A | #N/A | #N/A | #N/A | #N/A | 1E-06 |
| 6 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| 7 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| 8 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| 9 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| 10 | #N/A | #N/A | #N/A | #N/A | 1E-06 | #N/A | #N/A | #N/A | #N/A | #N/A |

*Individual Conductances (1/R)*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0.33333 | 0.05 | 0.11111 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.33333 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0.05 | 0 | 0 | 0.25 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0.11111 | 0 | 0.25 | 0 | 0.14286 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 1 | 0.14286 | 0 | 0 | 0 | 0 | 0 | 1000000 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 1000000 | 0 | 0 | 0 | 0 | 0 |
| | 0.49444 | 1.33333 | 1.3 | 0.50397 | 1000002 | 1 | 1 | 1 | 1 | 1000000 |

*Conductance Matrix*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.49444 | -0.3333 | -0.05 | -0.1111 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | -0.3333 | 1.33333 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| 3 | -0.05 | 0 | 1.3 | -0.25 | -1 | 0 | 0 | 0 | 0 | 0 |
| 4 | -0.1111 | 0 | -0.25 | 0.50397 | -0.1429 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | -1 | -1 | -0.1429 | 1000002 | 0 | 0 | 0 | 0 | -1E+06 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 | -1E+06 | 0 | 0 | 0 | 0 | 1000000 |

| BC |
|---|
| 9 |
| 4 |
| 0 |
| 0 |
| 4E+07 |
| 0 |
| 0 |
| 0 |
| 0 |

# *Data Input Portion of Spreadsheet*

- Created an array of resistance values for each resistance between nodes
- Labels in the first column follow a consistent format of R_a-b, where a and b are the two nodes
- Also input the heat loads for each node
- Input ambient temperature at node 10

| Resistance | Value |
|---|---|
| R_1-2 | 3 |
| R_1-3 | 20 |
| R_1-4 | 9 |
| R_2-5 | 1 |
| R_3-4 | 4 |
| R_3-5 | 1 |
| R_4-5 | 7 |
| R_5-10 | 1.00E-06 |
| | |
| | |
| | |
| | |
| | |

| Node | Q |
|---|---|
| 1 | 9 |
| 2 | 4 |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |

| | |
|---|---|
| T_10 | 40 |

# *Resistance Matrix*

- With the resistance values in two columns with consistent formatting, you can generate a resistance matrix
- Start with an NxN array with numbers from 1 to N along the top and the left edge (N is the maximum # of nodes)

```
=IF($J4<K$3,OFFSET($J$3,K$3,$J4),VLOOKUP("R_"&K$3&"-"&$J4,$B$4:$C$48,2,FALSE))
```

- Populates cells with the resistance values defined in the input region
- #N/A if no value available

**Resistances**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | #N/A | 3 | 20 | 9 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| **2** | 3 | #N/A | #N/A | #N/A | 1 | #N/A | #N/A | #N/A | #N/A | #N/A |
| **3** | 20 | #N/A | #N/A | 4 | 1 | #N/A | #N/A | #N/A | #N/A | #N/A |
| **4** | 9 | #N/A | 4 | #N/A | 7 | #N/A | #N/A | #N/A | #N/A | #N/A |
| **5** | #N/A | 1 | 1 | 7 | #N/A | #N/A | #N/A | #N/A | #N/A | 1E-06 |
| **6** | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| **7** | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| **8** | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| **9** | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| **10** | #N/A | #N/A | #N/A | #N/A | 1E-06 | #N/A | #N/A | #N/A | #N/A | #N/A |

# *Converting Inputs into the [C] and [Q] Matrices*

- The [C] matrix that we saw before followed a pattern:
  - terms in each cell is -1/R, except along the diagonal (row = column) the term is the sum of 1/R for that row (or column)

$$\begin{bmatrix} (C_{1-2}+C_{1-3}+C_{1-4}) & -C_{1-2} & -C_{1-3} & -C_{1-4} \\ -C_{1-2} & (C_{1-2}+C_{2-5}) & 0 & 0 \\ -C_{1-3} & 0 & (C_{1-3}+C_{3-4}+C_{3-5}) & -C_{3-4} \\ -C_{1-4} & 0 & -C_{3-4} & (C_{1-4}+C_{3-4}+C_{4-5}) \end{bmatrix}$$

- Terms in the [Q] matrix are the heat dissipation plus the conductance * ambient temperature

- $$\begin{bmatrix} Q_1 \\ Q_2 + C_{2-5}T_5 \\ C_{3-5}T_5 \\ C_{4-5}T_5 \end{bmatrix}$$

# 1/R and Conductance Matrices

=IF(ISERROR(K4),0,1/K4)

*If the value in the R matrix is #N/A, value is 0, if not, value is 1/R*

*Individual Conductances (1/R)*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0.33333 | 0.05 | 0.11111 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.33333 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0.05 | 0 | 0 | 0.25 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0.11111 | 0 | 0.25 | 0 | 0.14286 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 1 | 0.14286 | 0 | 0 | 0 | 0 | 0 | 1000000 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 1000000 | 0 | 0 | 0 | 0 | 0 |
| | 0.49444 | 1.33333 | 1.3 | 0.50397 | 1000002 | 1 | 1 | 1 | 1 | 1000000 |

*adds up the 1/R values in the same column; if the sum = 0, set the value to 1*

=IF(SUM(K19:K28)=0,1,SUM(K19:K28))

*nce Matrix*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.49444 | -0.33333 | -0.05 | -0.11111 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | -0.33333 | 1.33333 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| 3 | -0.05 | 0 | 1.3 | -0.25 | -1 | 0 | 0 | 0 | 0 | 0 |
| 4 | -0.11111 | 0 | -0.25 | 0.50397 | -0.14286 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | -1 | -1 | -0.14286 | 1000002 | 0 | 0 | 0 | 0 | -1000000 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 | -1000000 | 0 | 0 | 0 | 0 | 1000000 |

=IF(K$34=$J35,K$30,-K19)

*if the row and column numbers are the same, use the sum term for the column, otherwise use the -1/R term*

# *Step 3: Matrix Analysis to Solve for Temps*

- Excel has functions for dealing with matrices
  - Minverse() calculates the inverse of a matrix
  - Mmult() multiplies to matrices
- To use, highlight entire region where result will go, type in equation and then hit ctl-shift-enter
  - resulting matrix equation will have curly brackets (you don't put in)

{=MINVERSE(K35:S43)}

*calculates the inverse of [C]*

{=MMULT(K50:S58,W35:W43)}

*calculates the inverse of* $[C]^{-1}[Q]$

***Inverse Conductance Matrix***

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 2.650988 | 0.662748 | 0.236966 | 0.70202 | 0.000001 | 0 | 0 | 0 | 0 |
| **2** | 0.662748 | 0.915688 | 0.059242 | 0.175506 | 0.000001 | 0 | 0 | 0 | 0 |
| **3** | 0.236966 | 0.059242 | 0.871534 | 0.484581 | 0.000001 | 0 | 0 | 0 | 0 |
| **4** | 0.70202 | 0.175506 | 0.484581 | 2.379411 | 0.000001 | 0 | 0 | 0 | 0 |
| **5** | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0.000001 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Node | Temp |
|---|---|
| **1** | 66.5 |
| **2** | 49.6 |
| **3** | 42.4 |
| **4** | 47.0 |
| **5** | 40.0 |
| **6** | 0.0 |
| **7** | 0.0 |
| **8** | 0.0 |
| **9** | 0.0 |

# *General things for generic solver*

- The spreadsheet was set up with node 10 defined as the ambient temperature
  - to make things work for my example and maintain the same node notations, I connected node 5 to node 10 with a very, very low thermal resistance
- I used Data Validation to limit what names I could give the resistances (so that I couldn't type a name in incorrectly)*
- I used Conditional Formatting to hide the temperatures calculated for inactive nodes*

**see back up information for details*

# *Spreadsheet Setup*

- It helps if you can treat different solution steps as separate subroutines, with each subroutine being on a different worksheet
- One worksheet is where you input geometry and other conditions
- Other worksheets use those inputs to calculate individual thermal resistance values
- The generic matrix solver is just another subroutine / worksheet that you can reuse in other analyses

# *Summary*

- Spreadsheets are handy for doing quick analysis – particularly if you want to share the analysis tool with other users
- Break analysis into separate, reusable 'sub-routines' that consist of self-contained worksheets
  - a matrix solver is just another of these sub-routines
- Spreadsheet based thermal resistance analysis can help the gap between 'back of the envelope' initial feasibility calculations and detailed FEM/CFD optimization/design verification

# SPREADSHEET EXAMPLE

THERMAL LIVE 2015

# Thanks for attending!

***Don’t miss Thermal Live 2016!***
***Fall 2016***
***www.thermallive2016.com***

# Back up Information

# *Generating a Resistance Matrix*

- Start with a list of all of your resistances in your system
- Key is to use a standard naming convention
  - use something like R_1-2 for the resistance between nodes 1 and 2
  - be consistent!  if you use “R_” for one, use them for all
  - no need to define R_2-1, because it will be the same as R_1-2
    - but you must be consistent and show the smaller number first (or the larger – just be consistent!)
- Create a list of your resistances with the names in one column and the values for each resistance in the adjacent column
  - when I use the term ‘name’ here, I am being informal; DO NOT explicitly name the values using the ‘Name Manager’ on the “Formulas” tab

# Data Entry Format for Resistances

- Create your list of thermal resistance values in two adjacent columns with the left column showing the name
  - name needs to include the numbers (or letters – however you define the nodes) for each node
- Ideally, the values of the resistances are calculated elsewhere in your spreadsheet based on resistance equations
  - (you know, those ones that I said I wouldn't talk about...)
- Likewise, generate a list of power dissipations for each node

| Resistance | Value |
|---|---|
| R_1-2 | 3 |
| R_1-3 | 20 |
| R_1-4 | 9 |
| R_2-5 | 1 |
| R_3-4 | 4 |
| R_3-5 | 1 |
| R_4-5 | 7 |
| R_5-10 | 1.00E-06 |
| | |
| | |
| | |
| | |
| | |
| | |

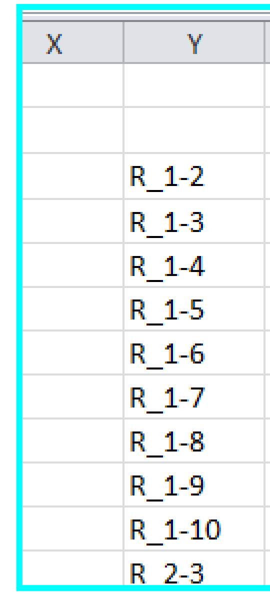
| Node | Q |
|---|---|
| 1 | 9 |
| 2 | 4 |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |

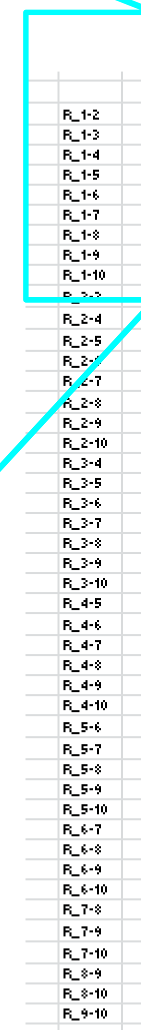
| | |
|---|---|
| T_10 | 40 |

*To help ensure that you use the right format for the resistance name, you might use Data Validation (see appendix)*

# Data Validation

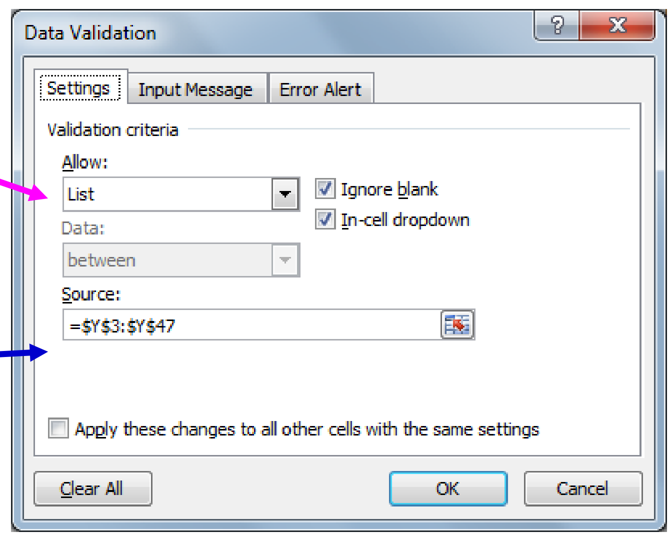
- To start, make a list of all the possible resistance names to choose from
- Highlight the cells in which you would enter the resistance name
- On the Data tab, select Data Validation / Data Validation



select 'List'

select range where the predefined list of names is




| Resistance | Value |
|---|---|
| R_1-2 | 3 |
| R_1-3 | 20 |
| R_1-4 | 9 |
| R_2-5 | 1 |
| R_3-4 | 4 |
| R_3-5 | 1 |
| R_4-5 | 7 |
| R_5-10 | 1.00E-06 |

# *Details on Function for R Matrix*

```
=IF($J4<K$3,OFFSET($J$3,K$3,$J4),VLOOKUP("R_"&K$3&"-
"&$J4,$B$4:$C$48,2,FALSE))
```

| Visual Basic | What it Does? | Why it Does it? |
|---|---|---|
| =IF($J4<K$3, | If the row # is less than the column #, then... | if a cell is below the diagonal of the matrix, this will grab the corresponding value in the symmetric matrix (i.e, says that R_2-1 = R_1-2) |
| OFFSET($J$3,K$3,$J4), | look at the value in the cell in which the row & column #'s are switched | |
| VLOOKUP( | looks up up a value in in the same row of an array | If the cell is above or on the diagonal, its value will be equal to thermal resistance (in the list) for R_a-b, where a is the column # and b is the row number.<br><br>If no value is found, the vlookup() function returns #N/A |
| R_"&K$3&"-"&$J4, | generates a text block that consists of R_column #-row # | |
| $B$4:$C$48, | defines what array to query | |
| 2, | defines which column in the array to get the data from | |
| FALSE)) | says that names in the 1st column may not be in order | |

# *Conditional Formatting*

- Highlight over a cell in the (yellow) temperature output array
- On the Home tab, select Conditional Formatting / New Rule
- Select ‘Use a formula to determine which cells to format’
- in the Rule Description, enter =(cell ref) = 0, where cell ref is the location of your cell (ex. H4)
- under Format, change the font color to match the yellow background

| Resistance | Value |
|---|---|
| R_1-2 | 3 |
| R_1-3 | 20 |
| R_1-4 | 9 |
| R_2-5 | 1 |
| R_3-4 | 4 |
| R_3-5 | 1 |
| R_4-5 | 7 |
| R_5-10 | 1.00E-06 |
| | |
| | |

| Node | Q |
|---|---|
| 1 | 9 |
| 2 | 4 |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |

| Temp |
|---|
| 66.5 |
| 49.6 |
| 42.4 |
| 47.0 |
| 40.0 |
| 0.0 |
| 0.0 |
| 0.0 |
| 0.0 |