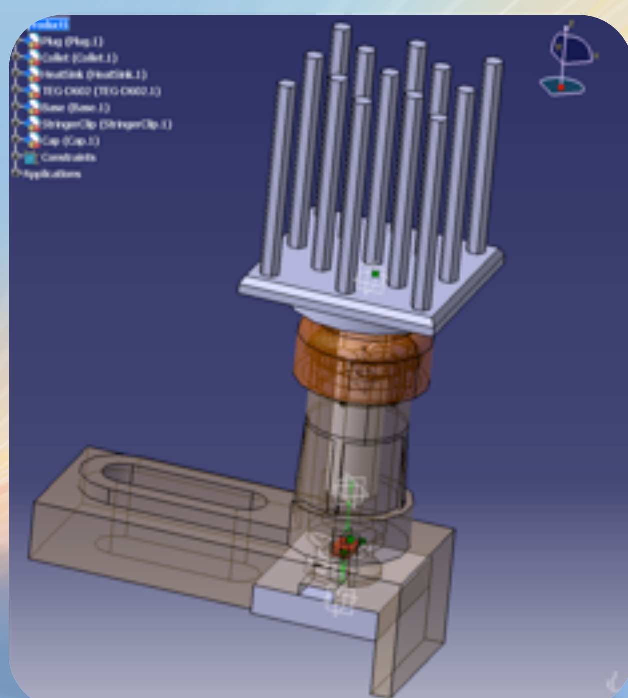
Numerical Heat Transfer Optimization in VBA

ISYS 540 Winter 2010 Final Project

Matthew Zabawa



Executive Summary

For my masters project in mechanical engineering I created a numerical model for the heat transfer through a thermoelectric energy harvester for Boeing. I created the numerical routine in Matlab because that was the language I was familiar with. Had I known how to use VBA, I would have used it. For this project, I converted the Matlab code into an integrated VBA code using a function to calculate a thermal resistance using a numerical heat transfer model and Gauss-Seidel iterative solver. I next used a macro to run Solver to optimize the design of the energy harvester for maximum efficiency. The results from the design tool match my original Matlab, but the new design tool is much simpler and more elegant.

Project Proposal

Conversion of Numerical Heat Transfer Optimization Model to VBA from Matlab

For my masters project in mechanical engineering I created a numerical model for the heat transfer through a thermoelectric energy harvester for Boeing. I created the numerical routine in Matlab because that was the language I was familiar with. Had I known how to use VBA, I would have used it. This project will require me to translate part of the numerical heat transfer code and incorporate VBA functions and Solver to optimize the design. I will send this completed project to my sponsors at Boeing. It will be more useful for them because they don't have easy access to Matlab in their design group and they will be compiling this model with others for a comprehensive design model.

Project Background

Thermoelectric Energy Harvesting

Energy harvesting is an important emerging area as engineers are becoming more concerned about the energy footprint of the systems they design and the machines they build. In particular, there are numerous locations within aircraft where large temperature gradients exist, so thermal energy harvesting systems present very attractive opportunities to aircraft manufacturers. The power necessary to operate many of the sensors and transmitters onboard aircraft is provided by batteries or by hard wiring them to the auxiliary power unit. Hard wiring these devices increases the weight and complexity of the system and use of batteries increases the amount of maintenance required. Therefore, the development of systems capable of harvesting the required power from locally occurring thermal gradients would enable the deployment of wireless networks of sensors and transmitters. In addition, this deployment would be flexible, since the network could be reconfigured without necessitating that the overall system wiring be redone.

The key element in a thermoelectric energy harvesting system is a device capable of directly converting thermal energy into electrical energy. There are several technologies capable of the direct conversion of heat to electricity. Examples include thermionic engines, thermal photovoltaic and thermoelectric devices. Although the fundamental operating principles of these devices have been know for decades, their low efficiencies have limited their use. However, recent advancements in the areas of nanotechnology and thin film manufacturing processes have lead to the development of higher efficiency devices. In particular, nanostructured materials have been shown to exhibit significantly better thermoelectric properties than do conventional materials. This phenomenon is due to the fact that size effects in nanostructred materials reduce their thermal conductivity, while maintaining a suitably high electrical conductivity. Since the performance of a thermoelectric device is proportional to its electrical conductivity and is inversely proportional to its thermal conductivity, the efficiency of thermoelectric generators has been improved, and many expect dramatic improvements in the near future. Therefore, the development of energy harvesting systems based on thermoelectric devices appears very promising.

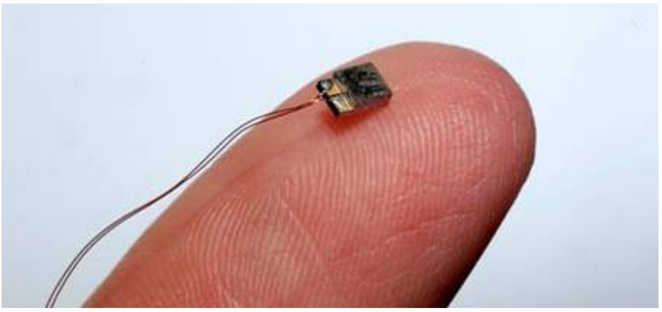


Figure 1: Micropelt Thermoelectric Generator

Boeing Systems Concept Center (SCC)

The SCC is an offsite Skunkworks-like development group within Boeing Commercial Airplanes. Composed of nine engineers from the different systems groups, the charter of the SCC is to look at emerging technologies and find application for them in airplanes. The concepts they come up with are to be incorporated into airplanes anywhere for 2 to 20 years down the line. After I graduated with my bachelors in mechanical engineering, I interned at the SCC for the summer. The research I began at the SCC on thermoelectric energy harvesting was continued into my masters project at BYU. This model is one of the key deliverables from my masters project.

Stringer Clip Energy Harvester

A stringer clip is a plastic clip that anchors onto the outer structure of an airplane. The stringer clips serve as mounting points for the insulation blankets and wiring harnesses that run throughout the airplane. The temperature of the skin of the aircraft while in flight is nominally -30°C, while the inside of the aircraft is kept at 20° C. Since the stringer clip is attached to the skin of the airplane and equipment on the inside of the airplane it has a large temperature gradient across it making it an ideal place to harvest thermal energy.

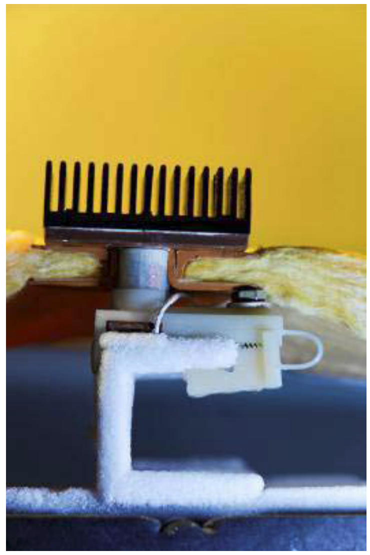


Figure 2: Earlier design of a stringer clip energy harvester

The purpose of my masters project was to develop an alternate design for the stringer clip energy harvester. I developed a design and created a model to predict the performance of the design. This spreadsheet VBA project is a revision of my model that I delivered to Boeing and is intended to be an improved design tool for an engineer.

User Interface

The user interface in the Excel spreadsheet is divided between two pages with a third page providing a detailed visual for reference purposes only. The first sheet "Performance" computes the expected power from the thermoelectric generator based on a temperature gradient and a set of thermal resistances. The second sheet "NHTModel" is the sheet that uses the VBA subroutines to calculate the thermal resistance of the most complex part of the design: the mechanically compliant joint.

Harvester Power Performance Sheet

The "Performance" sheet contains all the inputs necessary to compute the power output for two different thermoelectric generator options except for the thermal resistance of the joint, which it pulls from the "NHTModel" sheet. This remains mostly unchanged from my original deliverable to Boeing except for the necessary changes to the instructions and description.

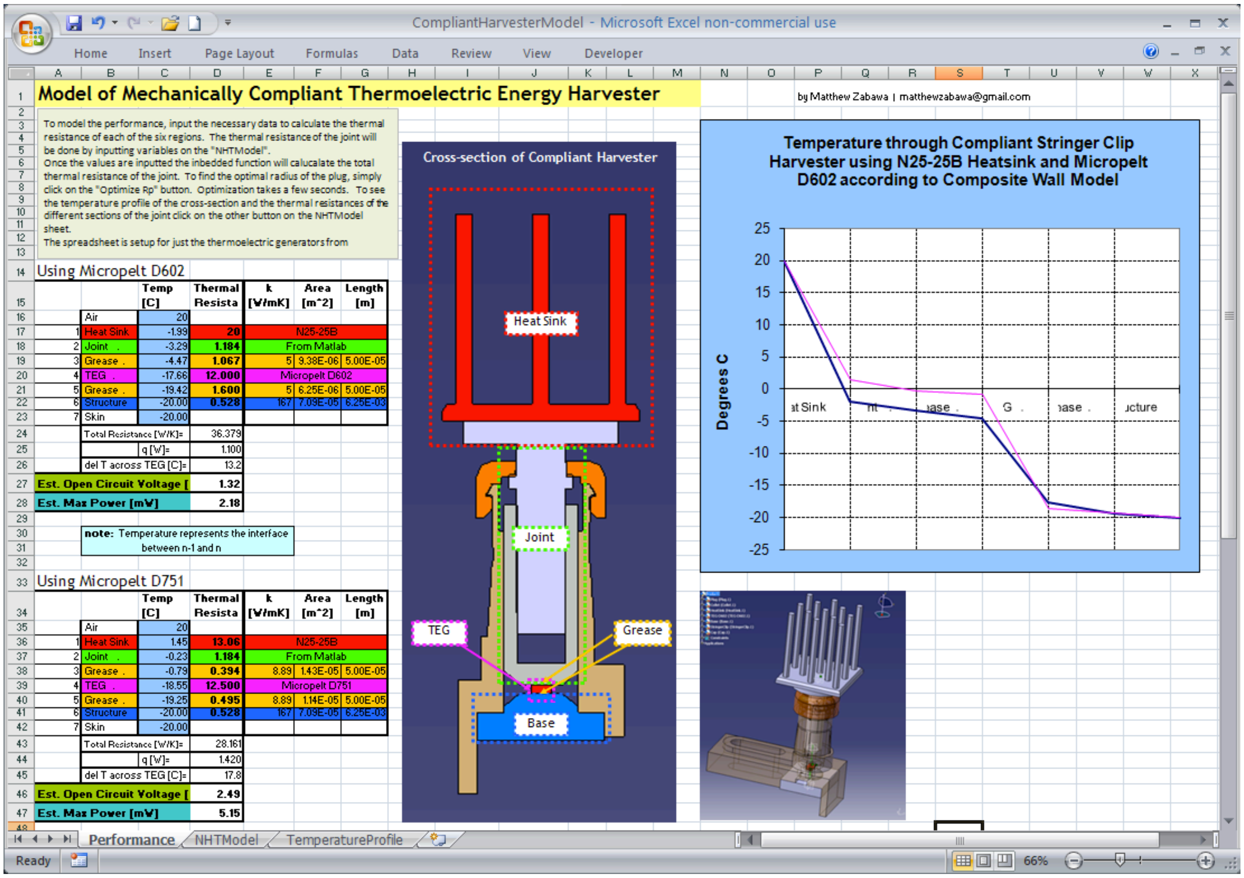


Figure 3: Screen shot of the front sheet of the Excel design tool

Mechanically Compliant Joint Design

The sheet "NHTModel" contains the VBA functions and macros that are the subject of the project. The subroutines are used to calculate a thermal resistance and then optimize the design of the joint to get the lowest thermal resistance possible.

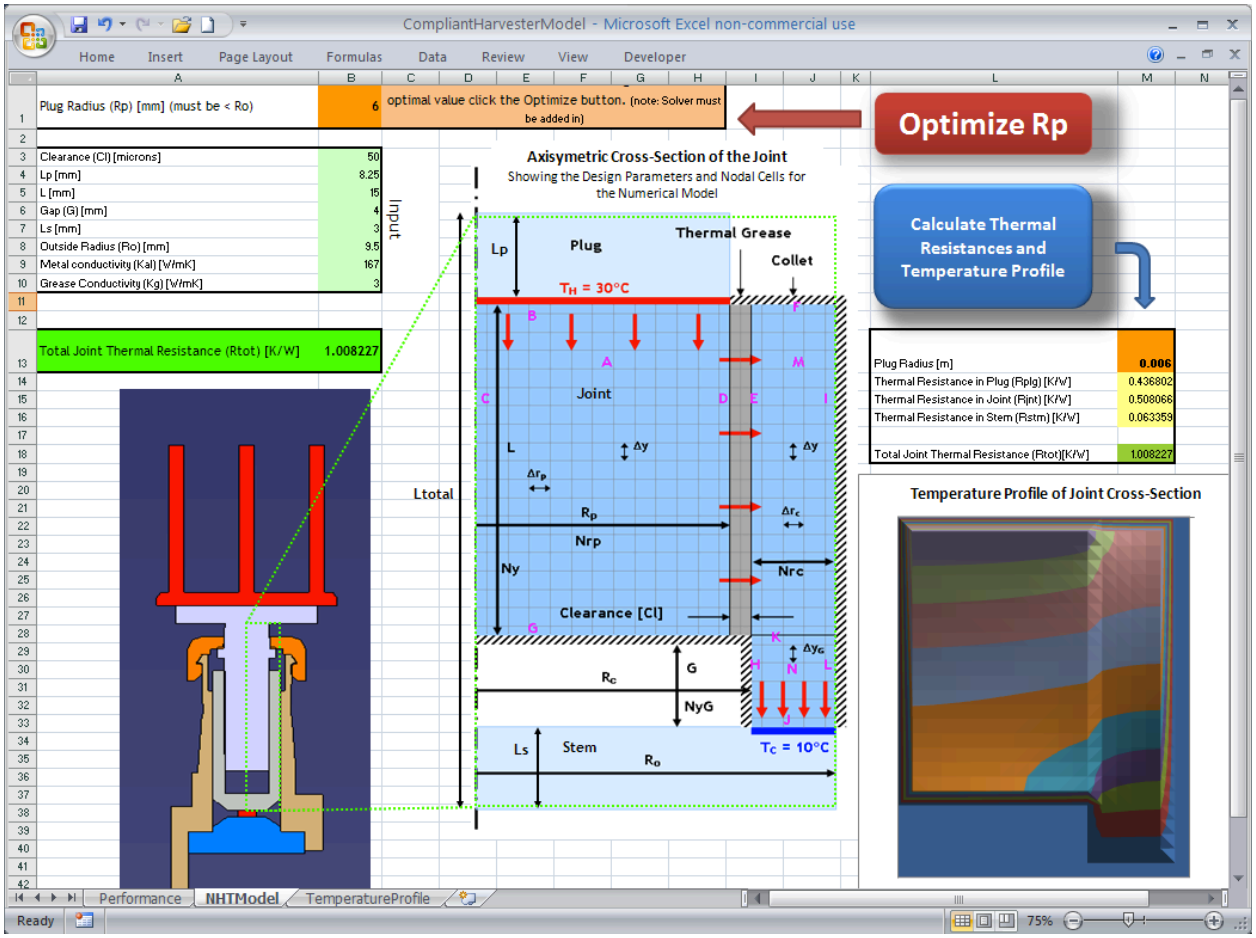


Figure 4: Screen shot of the second page of the Excel design tool

VBA Architecture

To run an optimization of design there are two necessary parts of code. One is the model that takes a set of variables and gives an output. In this case, the model is a numerical heat transfer model that takes design parameters and computes a thermal resistance using numerical methods. The second part is the optimization code. For this project, I am employing the Solver Add-in for Excel to find the radius of the plug that yields the lowest thermal resistance for the part.

Numerical Heat Transfer Function – ThermR()

The thermal resistance of the joint is a function of nine variables. I created a VBA function based on the Matlab code that I originally developed for my masters project. The function follows numerical heat transfer model for the equations and then uses a Gauss-Seidel iterative solver to find the solution. The function is named ThermR() and its use in the page can be seen below.

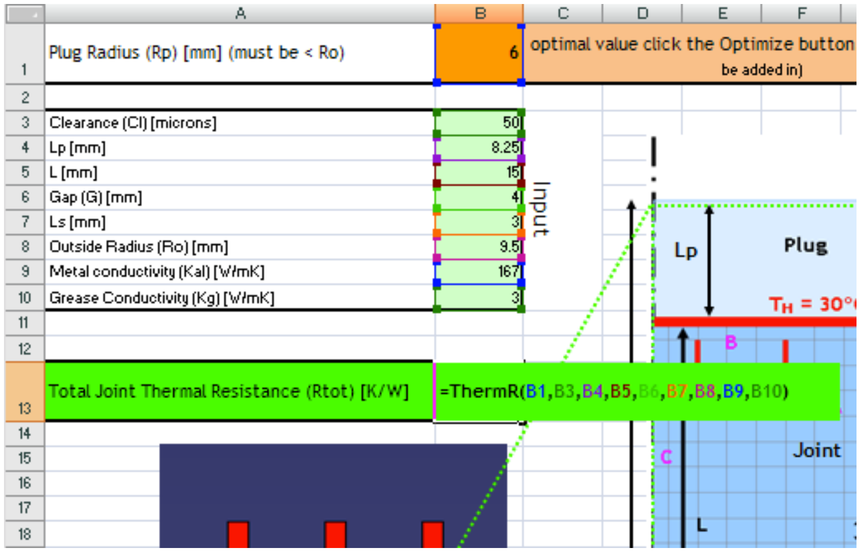


Figure 5: Use of the ThermR() function in the NHTModel sheet

Initially I had planned to use a macro to do the calculating and then use another macro to make Solver run the calculating macro and iterate to an optimal solution. I learned that Solver can only optimize a value in a cell that is the result of the function. This required me to change my calculating macro into a function.

Some of the difficulties I encountered were that I was not aware of some of the rules that limit what a function can do. In the process of debugging the function, I found that a function cannot output to other cells. Initially I had the function outputting the temperature profile and a breakdown of thermal resistance by section of the joint. This is not possible to do with a function so I had to create a separate macro that is run by the user.

Macro for Calculating Thermal Resistances and the Temperature Profile

The thermal resistances of the different sections of the joint are important information to be able to validate that the results are meaningful and to help identify areas of the joint that are poorly designed. The temperature profile of the joint cross-section is also another piece of useful information as it helps to see where the high temperature gradients might exist in the joint. To generate these helpful outputs, I created a macro named "PrintValues". The macro runs the numerical model one more time outputs the thermal resistances and temperature array. In this section of the project I learned how to output an array in Excel and get the right order.

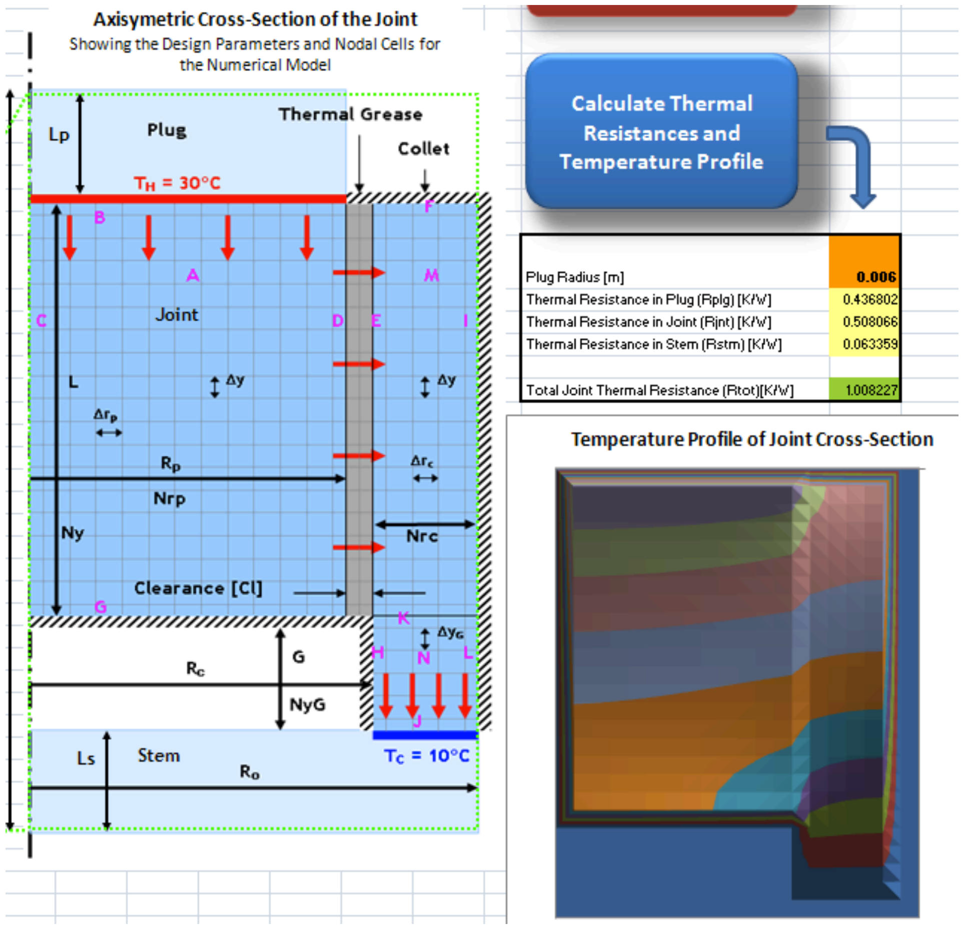


Figure 6: Screen shot of the operator button and the output from the PrintValues macro

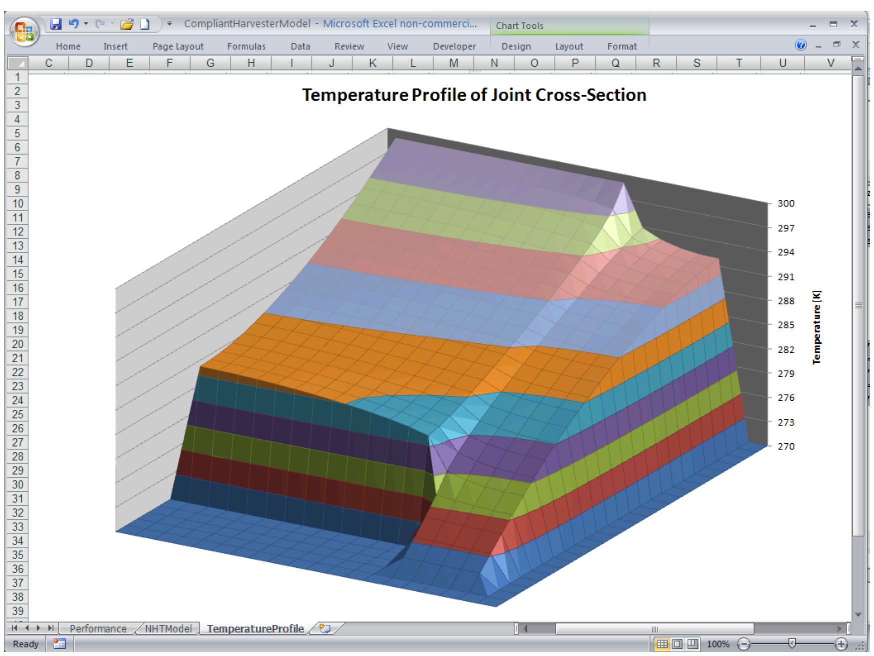


Figure 7: Isometric view of the temperature profile of the joint cross-section

Macro for Optimizing the Design using Solver

To optimize the design of the joint, I chose to create a macro that calls on Solver iterate to find the best solution. I recorded myself setting the Solver parameters including the constraints to create the macro. A few things I learned about using Solver in a macro is the reference to Solve.XLAM needs to be turned on in the VBA environment or it will not work. I also learned that solver will not want to work with variable that are in merged cells. After figuring out those obstacles, I was able to run the macro by pushing the button and found the exact same solution that found using my initial Matlab model.

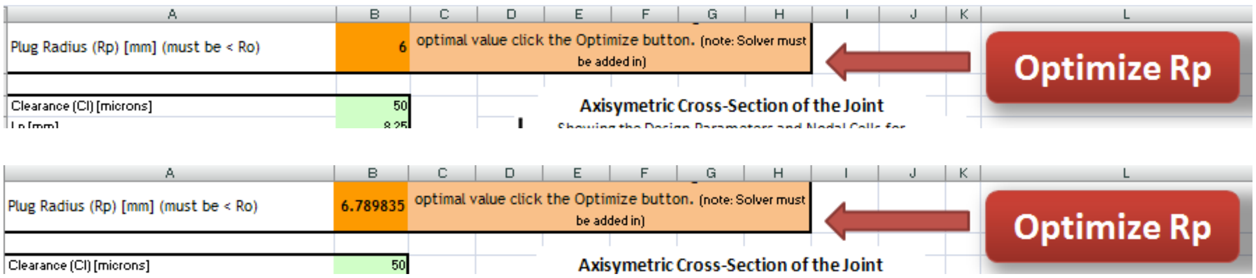


Figure 8: Plug Radius before and after running the optimization routine

The code for the OptimizePlugRadius macro is shown here.

```
'Macro that runs solver to find the optimal radius of the plug for the design
'note that in Excel Options/Add-ins that Solver must be selected
'Also the Reference to SOLVER.XLAM must to include in the VBA Project
Sub OptimizePlugRadius()
'
' OptimizePlugRadius Macro
'

'
    SolverReset
    SolverOk SetCell:="$B$13", MaxMinVal:=2, ValueOf:="0", ByChange:="$B$1"
    SolverAdd CellRef:="$B$1", Relation:=1, FormulaText:="$B$8-$B$3/1000-1"
    SolverOk SetCell:="$B$13", MaxMinVal:=2, ValueOf:="0", ByChange:="$B$1"
    SolverAdd CellRef:="$B$1", Relation:=3, FormulaText:="1.5"
    SolverOk SetCell:="$B$13", MaxMinVal:=2, ValueOf:="0", ByChange:="$B$1"
    SolverSolve
End Sub
```

Conclusion

This project allowed me the opportunity to improve my thermoelectric energy harvester model. The previous setup of having Matlab call values from the spreadsheet, calculate, optimize, and print to the spreadsheet was inelegant and unnecessary. I am much more pleased with this VBA design tool and think that it is much more flexible for the end user.

Appendix

Full Numerical Heat Transfer Function

```
'Function that runs the Numerical Heat Transfer Model to Calculate Total Thermal Resistance
Function ThermR(Rp As Double, Cl As Double, Lp As Double, L As Double, Gap As Double, Ls As
Double, Ro As Double, Kal As Double, Kg As Double)

'Convert to SI units
Rp = Rp / 1000
Cl = Cl / 1000000
Lp = Lp / 1000
L = L / 1000
Gap = Gap / 1000
Ls = Ls / 1000
Ro = Ro / 1000

    a = 0.05 * Ro
    b = 0.95 * Ro

'Debug.Print Rp, Ro, a, b


'Debug.Print Cl, Lp, L, Gap, Ls, Ro, Kal, Kg

Dim Thot As Double
Dim Tcold As Double
Dim Nrp As Double
Dim Nrc As Double
Dim Ny As Double
Dim Nyg As Double

'Declare Temperature settings (don't affect R)
Thot = 300
Tcold = 270

'Declare Array Size Parameters
Nrp = 13
Nrc = 5
Ny = 19
Nyg = 5

'Compute function parameters
Dim Rc As Double
Dim Ct As Double
Dim delr As Double
Dim delrc As Double
Dim dely As Double
Dim delyg As Double
Dim Nrt As Double
Dim Nyt As Double
Dim itmax As Double

Rc = Rp + Cl
Ct = Ro - Rc
delr = Rp / (Nrp - 1)
delrc = Ct / (Nrc - 1)
dely = L / (Ny - 1)
delyg = Gap / Nyg
Nrt = Nrp + Nrc
Nyt = Ny + Nyg
itmax = Nrt * Nyt
'Debug.Print "next"
'Debug.Print Rc, Ct, delr, delrc, dely, delyg, Nrt, Nyt, itmax

'Allocate and initialize matrices
Dim m As Integer
Dim n As Integer
Dim aP(22, 27) As Double
Dim aN(22, 27) As Double
Dim aSo(22, 27) As Double
```

```
Dim aE(22, 27) As Double
Dim aW(22, 27) As Double
Dim T(22, 27)  As Double

For m = 1 To 21
    For n = 1 To 26
            aP(m, n) = 1
            aN(m, n) = 0
            aSo(m, n) = 0
            aE(m, n) = 0
            aW(m, n) = 0
            T(m, n) = 1

    Next
Next

'dimension and set pi
Dim pi As Double

pi = 3.14159265


'Create coefficient matrices
For m = 2 To Nrp + Nrc + 1 + 1
        For n = 2 To Ny + Nyg + 1
            'Vertical Walls and all corners
                If m = 2 Then 'Adiabatic Axisymetric condition "C Eqns"
                        If n = 2 Then 'Top Temperature Set Plug Condition, Top left Corner
                                aN(m, n) = 0
                                aSo(m, n) = Kal * pi * delr ^ 2 / dely
                                aE(m, n) = Kal * pi * dely
                                aW(m, n) = 0
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        ElseIf n = Ny + 1 Then ' Bottom Adiabatic Plug, Gap left corner of plug
                                aN(m, n) = Kal * pi * delr ^ 2 / dely
                                aSo(m, n) = 0
                                aE(m, n) = Kal * pi * dely
                                aW(m, n) = 0
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        ElseIf n > Ny + 1 Then ' Gap on the centerline
                                aN(m, n) = 0
                                aSo(m, n) = 0
                                aE(m, n) = 0 'change
                                aW(m, n) = 0
                                aP(m, n) = 1
                        Else ' Rest of Left side plug wall (center line)
                                aN(m, n) = Kal * pi * delr ^ 2 / dely
                                aSo(m, n) = Kal * pi * delr ^ 2 / dely
                                aE(m, n) = Kal * pi * dely
                                aW(m, n) = 0
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        End If
                ElseIf m = Nrp + 1 Then ' Right plug wall "D Eqns"
                        If n = 2 Then ' Top Temperature Set Plug Condition, Top right Corner of
plug
                                aN(m, n) = 0
                                aSo(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                                aE(m, n) = 2 * Kg * (m + 1) * delr * dely * pi / Cl
                                aW(m, n) = Kal * pi * dely * (2 * (m + 1) - 1)
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        ElseIf n = Ny + 1 Then ' Bottom Adiabatic plug condition, gap right
corner of plug
                                aN(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                                aSo(m, n) = 0
                                aE(m, n) = 2 * Kg * m * delr * dely * pi / Cl
                                aW(m, n) = Kal * pi * dely * (2 * (m + 1) - 1)
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        ElseIf n > Ny + 1 Then ' Interior Adiabatic wall of Gap
                                aN(m, n) = 0
                                aSo(m, n) = 0 ' Change
                                aE(m, n) = 0
                                aW(m, n) = 0
                                aP(m, n) = 1
```

```
                Else ' Rest of right side plug wall
                        aN(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                        aSo(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                        aE(m, n) = 2 * Kg * (m + 1) * delr * dely * pi / Cl
                        aW(m, n) = Kal * pi * dely * (2 * (m + 1) - 1)
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                End If
        ElseIf m = Nrp + 2 Then ' Left collet wall "E Eqns, H Eqns"
                If n = 2 Then ' Top left corner of collet "E"
                        aN(m, n) = 0
                        aSo(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                        aE(m, n) = pi * dely * Kal * (Rc / delrc + 1 / 2)
                        aW(m, n) = Kg * dely * pi * Rc / Cl
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n = Ny + 1 Then ' Interface with gap seciton of collet on left
side "E,H"
                        aN(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                        aSo(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                        aE(m, n) = 2 * pi * dely * Kal * (Rc / delrc + 1 / 2)
                        aW(m, n) = Kg * dely * 2 * pi * Rc / Cl
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n = Ny + Nyg + 1 Then ' Bottom Left corner of collet "H"
                        aN(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                        aSo(m, n) = 0
                        aE(m, n) = Kal * delyg * pi * (2 * Rc / delrc + 1)
                        aW(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n < Ny + 1 Then ' Collet above interface left wall, not corner "E"
                        aN(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                        aSo(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                        aE(m, n) = 2 * pi * dely * Kal * (Rc / delrc + 1 / 2)
                        aW(m, n) = Kg * dely * 2 * pi * Rc / Cl
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                Else  ' Collet in gap area on left side "H"
                        aN(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                        aSo(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                        aE(m, n) = Kal * delyg * pi * (2 * Rc / delrc + 1)
                        aW(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                End If
        ElseIf m = Nrc + Nrp + 1 Then ' Adiabatic Collet Condition "I Eqns, L Eqns"
                If n = 2 Then ' Top right corner of collet "I"
                        aN(m, n) = 0
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aW(m, n) = Kal * dely * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n = Ny + 1 Then ' Interface with gap seciton of collet on right
side "I,L"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aW(m, n) = Kal * dely * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n = Ny + Nyg + 1 Then ' Bottom Right corner of collet "L"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aSo(m, n) = 0
                        aW(m, n) = Kal * delyg * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n < Ny + 1 Then ' Collet above interface right wall, not corner
"I"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aW(m, n) = Kal * dely * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                Else  ' Collet in gap area on right side "L"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aW(m, n) = Kal * delyg * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
```

```
                        End If
                Else ' Horizontal Walls, NO corners
                        If n = 2 Then
                                If m < Nrp + 1 Then ' Top Plug Temp. Defined wall "B Eqns"
                                        aN(m, n) = 0
                                        aSo(m, n) = Kal * 2 * pi * (m + 1) * delr ^ 2 / dely
                                        aE(m, n) = Kal * dely * pi * (2 * (m + 1) + 1) / 2
                                        aW(m, n) = Kal * dely * pi * (2 * (m + 1) - 1) / 2
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                ElseIf m > Nrp + 2 Then ' Top Collet adiabatic condition "F Eqns"
                                        aN(m, n) = 0
                                        aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                        aE(m, n) = pi * Kal * dely * (Rc / delrc + (m + 1) - Nrp
- 1 / 2)
                                        aW(m, n) = pi * Kal * dely * (Rc / delrc + (m + 1) - Nrp
- 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        ElseIf n = Ny + 1 Then ' Adiabatic plug bottom and collet interface to
gap "G Eqns, K Eqns"
                                If m < Nrp + 1 Then ' Adiabatic Plug bottom, NO corner "G Eqns"
                                        aN(m, n) = Kal * 2 * pi * (m + 1) * delrc ^ 2 / delyg
                                        aSo(m, n) = 0
                                        aE(m, n) = Kal * pi * delyg * (2 * (m + 1) + 1)
                                        aW(m, n) = Kal * pi * delyg * (2 * (m + 1) - 1)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                ElseIf m > Nrp + 2 Then ' Collet Interface to gap "K Eqns"
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                        aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
                                        aE(m, n) = pi * Kal * (delyg + dely) * (Rc / delrc + (m +
1) - Nrp - 1 / 2)
                                        aW(m, n) = pi * Kal * (delyg + dely) * (Rc / delrc + (m +
1) - Nrp - 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        ElseIf n = Ny + Nyg + 1 Then ' Temperature set Collet Bottom, No Corner
"J Eqns"
                                If m > Nrp + 2 Then ' To avoid left corner
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
                                        aSo(m, n) = 0
                                        aE(m, n) = pi * Kal * delyg * (Rc / delrc + (m + 1) - Nrp
- 1 / 2)
                                        aW(m, n) = pi * Kal * delyg * (Rc / delrc + (m + 1) - Nrp
- 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        ElseIf n < Ny + 1 Then
                                If m < Nrp + 1 Then ' Plug body "A Eqns"
                                        aN(m, n) = Kal * 2 * pi * (m + 1) * delr ^ 2 / dely
                                        aSo(m, n) = Kal * 2 * pi * (m + 1) * delr ^ 2 / dely
                                        aE(m, n) = Kal * dely * pi * (2 * (m + 1) + 1)
                                        aW(m, n) = Kal * dely * pi * (2 * (m + 1) - 1)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                ElseIf m > Nrp + 2 Then ' Collet above interface "M Eqns"
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                        aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                        aE(m, n) = 2 * pi * Kal * dely * (Rc / delrc + (m + 1) -
Nrp - 1 / 2)
                                        aW(m, n) = 2 * pi * Kal * dely * (Rc / delrc + (m + 1) -
Nrp - 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        Else
                                If m > Nrp + 2 Then ' Gap area of Collet "N Eqns"
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
```

```
                                        aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
                                        aE(m, n) = 2 * pi * Kal * delyg * (Rc / delrc + (m + 1) -
Nrp - 1 / 2)
                                        aW(m, n) = 2 * pi * Kal * delyg * (Rc / delrc + (m + 1) -
Nrp - 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        End If
                End If
        Next
Next


' Initialize Temperatures
For m = 2 To Nrp + 1
        For n = 2 To Ny + 1
        T(m, n) = Thot ' Plug Temperature initialized at Thot
        Next
Next

For m = 2 To Nrc + 1
        For n = 2 To Ny + Nyg + 1
                T(m + Nrp, n) = Tcold  ' Collet Temperature initialized at Tcold
        Next
Next


'Gauss Seidel Solver
Dim it As Integer
Dim i As Integer

For it = 1 To itmax
    For n = 2 To Nyt + 1
        For m = 2 To Nrt + 1
            If n = 2 Then
                If m <= Nrp + 1 Then
                    T(m, n) = Thot
                Else
                    T(m, n) = (aE(m, n) * T(m + 1, n) + aW(m, n) * T(m - 1, n) + aN(m, n) * T(m,
n - 1) + aSo(m, n) * T(m, n + 1)) / aP(m, n)
                End If
            ElseIf n = Ny + Nyg + 1 Then
                If m >= Nrp + 2 Then
                    T(m, n) = Tcold
                End If
            Else
                T(m, n) = (aE(m, n) * T(m + 1, n) + aW(m, n) * T(m - 1, n) + aN(m, n) * T(m, n -
1) + aSo(m, n) * T(m, n + 1)) / aP(m, n)
            End If
        Next
    Next
Next

'sum the temperatures in a row across narrow section of the collet
Dim Ti As Double
Ti = 0

For i = Nrp + 2 To Nrp + Nrc + 1
    Ti = Ti + T(i, Ny + 1)
Next

'Compute the total heat flow through the part and the total resistance
Q = Kal * pi * (Ro * Ro - Rc * Rc) * (Ti / Nrc - Tcold) / Gap
Rplg = Lp / Kal / pi / Rp / Rp
Rjnt = (Thot - Tcold) / Q
Rstm = Ls / Kal / pi / Ro / Ro
Rtot = Rplg + Rjnt + Rstm

'Write to Spreadsheet
'Sheets("NHTModel").Range("B13").Value = Rp   'xlswrite('CompliantHarvester', Rp, 'Matlab',
'B13')
```

```
'Sheets("NHTModel").Range("B14").Value = Rplg 'xlswrite('CompliantHarvester', Rplg, 'Matlab',
'B14')
'Sheets("NHTModel").Range("B15").Value = Rjnt 'xlswrite('CompliantHarvester', Rjnt, 'Matlab',
'B15')
'Sheets("NHTModel").Range("B16").Value = Rstm 'xlswrite('CompliantHarvester', Rstm, 'Matlab',
'B16')
'Sheets("NHTModel").Range("B18").Value = Rtot 'xlswrite('CompliantHarvester', Rtot, 'Matlab',
'B18')



'Output the T array
'For m = 1 To 21
'    For n = 1 To 26
'           Sheets("test").Cells(n, m).Value = T(m, n)
'    Next
'Next

ThermR = Rtot


End Function
```

Full Macro for Calculating Thermal Resistances and the Temperature Profile

```
'Macro that Calculates the Thermal Resistance of the different sections of the Joint
'and outputs the T (temperature) array to the TemperatureProfile sheet
'to be used in the charts that show the temperature profile
Sub PrintValues()

Dim Cl As Double
Dim Lp As Double
Dim L As Double
Dim Gap As Double
Dim Ls As Double
Dim Kal As Double
Dim Kg As Double

'Bring in Values
Rp = Sheets("NHTModel").Range("B1")
Cl = Sheets("NHTModel").Range("B3")
Lp = Sheets("NHTModel").Range("B4")
L = Sheets("NHTModel").Range("B5")
Gap = Sheets("NHTModel").Range("B6")
Ls = Sheets("NHTModel").Range("B7")
Ro = Sheets("NHTModel").Range("B8")
Kal = Sheets("NHTModel").Range("B9")
Kg = Sheets("NHTModel").Range("B10")

'Convert to SI units
Rp = Rp / 1000
Cl = Cl / 1000000
Lp = Lp / 1000
L = L / 1000
Gap = Gap / 1000
Ls = Ls / 1000
Ro = Ro / 1000

    a = 0.05 * Ro
    b = 0.95 * Ro


'Debug.Print Cl, Lp, L, Gap, Ls, Ro, Kal, Kg

Dim Thot As Double
Dim Tcold As Double
Dim Nrp As Double
Dim Nrc As Double
Dim Ny As Double
```

```
Dim Nyg As Double

'Declare Temperature settings (don't affect R)
Thot = 300
Tcold = 270

'Declare Array Size Parameters
Nrp = 13
Nrc = 5
Ny = 19
Nyg = 5

'Compute function parameters
Dim Rc As Double
Dim Ct As Double
Dim delr As Double
Dim delrc As Double
Dim dely As Double
Dim delyg As Double
Dim Nrt As Double
Dim Nyt As Double
Dim itmax As Double

Rc = Rp + Cl
Ct = Ro - Rc
delr = Rp / (Nrp - 1)
delrc = Ct / (Nrc - 1)
dely = L / (Ny - 1)
delyg = Gap / Nyg
Nrt = Nrp + Nrc
Nyt = Ny + Nyg
itmax = Nrt * Nyt

'Debug.Print Rc, Ct, delr, delrc, dely, delyg, Nrt, Nyt, itmax

'Allocate and initialize matrices
Dim m As Integer
Dim n As Integer
Dim aP(22, 27) As Double
Dim aN(22, 27) As Double
Dim aSo(22, 27) As Double
Dim aE(22, 27) As Double
Dim aW(22, 27) As Double
Dim T(22, 27)  As Double

For m = 1 To 21
    For n = 1 To 26
            aP(m, n) = 1
            aN(m, n) = 0
            aSo(m, n) = 0
            aE(m, n) = 0
            aW(m, n) = 0
            T(m, n) = 1

    Next
Next

'dimension and set pi
Dim pi As Double

pi = 3.14159265


'Create coefficient matrices
For m = 2 To Nrp + Nrc + 1 + 1
        For n = 2 To Ny + Nyg + 1
            'Vertical Walls and all corners
                If m = 2 Then 'Adiabatic Axisymetric condition "C Eqns"
                        If n = 2 Then 'Top Temperature Set Plug Condition, Top left Corner
                                aN(m, n) = 0
                                aSo(m, n) = Kal * pi * delr ^ 2 / dely
                                aE(m, n) = Kal * pi * dely
                                aW(m, n) = 0
```

```
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n = Ny + 1 Then ' Bottom Adiabatic Plug, Gap left corner of plug
                aN(m, n) = Kal * pi * delr ^ 2 / dely
                aSo(m, n) = 0
                aE(m, n) = Kal * pi * dely
                aW(m, n) = 0
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n > Ny + 1 Then ' Gap on the centerline
                aN(m, n) = 0
                aSo(m, n) = 0
                aE(m, n) = 0 'change
                aW(m, n) = 0
                aP(m, n) = 1
            Else ' Rest of Left side plug wall (center line)
                aN(m, n) = Kal * pi * delr ^ 2 / dely
                aSo(m, n) = Kal * pi * delr ^ 2 / dely
                aE(m, n) = Kal * pi * dely
                aW(m, n) = 0
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            End If
        ElseIf m = Nrp + 1 Then ' Right plug wall "D Eqns"
            If n = 2 Then ' Top Temperature Set Plug Condition, Top right Corner of
plug
                aN(m, n) = 0
                aSo(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                aE(m, n) = 2 * Kg * (m + 1) * delr * dely * pi / Cl
                aW(m, n) = Kal * pi * dely * (2 * (m + 1) - 1)
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n = Ny + 1 Then ' Bottom Adiabatic plug condition, gap right
corner of plug
                aN(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                aSo(m, n) = 0
                aE(m, n) = 2 * Kg * m * delr * dely * pi / Cl
                aW(m, n) = Kal * pi * dely * (2 * (m + 1) - 1)
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n > Ny + 1 Then ' Interior Adiabatic wall of Gap
                aN(m, n) = 0
                aSo(m, n) = 0 ' Change
                aE(m, n) = 0
                aW(m, n) = 0
                aP(m, n) = 1
            Else ' Rest of right side plug wall
                aN(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                aSo(m, n) = Kal * pi * delr ^ 2 * ((m + 1) - 1 / 4) / dely
                aE(m, n) = 2 * Kg * (m + 1) * delr * dely * pi / Cl
                aW(m, n) = Kal * pi * dely * (2 * (m + 1) - 1)
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            End If
        ElseIf m = Nrp + 2 Then ' Left collet wall "E Eqns, H Eqns"
            If n = 2 Then ' Top left corner of collet "E"
                aN(m, n) = 0
                aSo(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                aE(m, n) = pi * dely * Kal * (Rc / delrc + 1 / 2)
                aW(m, n) = Kg * dely * pi * Rc / Cl
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n = Ny + 1 Then ' Interface with gap seciton of collet on left
side "E,H"
                aN(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                aSo(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                aE(m, n) = 2 * pi * dely * Kal * (Rc / delrc + 1 / 2)
                aW(m, n) = Kg * dely * 2 * pi * Rc / Cl
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n = Ny + Nyg + 1 Then ' Bottom Left corner of collet "H"
                aN(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                aSo(m, n) = 0
                aE(m, n) = Kal * delyg * pi * (2 * Rc / delrc + 1)
                aW(m, n) = 0
                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
            ElseIf n < Ny + 1 Then ' Collet above interface left wall, not corner "E"
                aN(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                aSo(m, n) = Kal * pi * (Rc * delrc + delrc ^ 2 / 4) / dely
                aE(m, n) = 2 * pi * dely * Kal * (Rc / delrc + 1 / 2)
                aW(m, n) = Kg * dely * 2 * pi * Rc / Cl
```

```
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                Else  ' Collet in gap area on left side "H"
                        aN(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                        aSo(m, n) = Kal * pi * delrc * (Rc + delrc / 4) / delyg
                        aE(m, n) = Kal * delyg * pi * (2 * Rc / delrc + 1)
                        aW(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                End If
        ElseIf m = Nrc + Nrp + 1 Then ' Adiabatic Collet Condition "I Eqns, L Eqns"
                If n = 2 Then ' Top right corner of collet "I"
                        aN(m, n) = 0
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aW(m, n) = Kal * dely * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n = Ny + 1 Then ' Interface with gap seciton of collet on right
side "I,L"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aW(m, n) = Kal * dely * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n = Ny + Nyg + 1 Then ' Bottom Right corner of collet "L"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aSo(m, n) = 0
                        aW(m, n) = Kal * delyg * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                ElseIf n < Ny + 1 Then ' Collet above interface right wall, not corner
"I"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / dely
                        aW(m, n) = Kal * dely * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                Else  ' Collet in gap area on right side "L"
                        aN(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aSo(m, n) = Kal * pi * delrc * (Ro - delrc / 4) / delyg
                        aW(m, n) = Kal * delyg * pi * (2 * Ro / delrc - 1)
                        aE(m, n) = 0
                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                End If
        Else ' Horizontal Walls, NO corners
                If n = 2 Then
                        If m < Nrp + 1 Then ' Top Plug Temp. Defined wall "B Eqns"
                                aN(m, n) = 0
                                aSo(m, n) = Kal * 2 * pi * (m + 1) * delr ^ 2 / dely
                                aE(m, n) = Kal * dely * pi * (2 * (m + 1) + 1) / 2
                                aW(m, n) = Kal * dely * pi * (2 * (m + 1) - 1) / 2
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        ElseIf m > Nrp + 2 Then ' Top Collet adiabatic condition "F Eqns"
                                aN(m, n) = 0
                                aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                aE(m, n) = pi * Kal * dely * (Rc / delrc + (m + 1) - Nrp
- 1 / 2)
                                aW(m, n) = pi * Kal * dely * (Rc / delrc + (m + 1) - Nrp
- 3 / 2)
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        End If
                ElseIf n = Ny + 1 Then ' Adiabatic plug bottom and collet interface to
gap "G Eqns, K Eqns"
                        If m < Nrp + 1 Then ' Adiabatic Plug bottom, NO corner "G Eqns"
                                aN(m, n) = Kal * 2 * pi * (m + 1) * delrc ^ 2 / delyg
                                aSo(m, n) = 0
                                aE(m, n) = Kal * pi * delyg * (2 * (m + 1) + 1)
                                aW(m, n) = Kal * pi * delyg * (2 * (m + 1) - 1)
                                aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                        ElseIf m > Nrp + 2 Then ' Collet Interface to gap "K Eqns"
                                aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
```

```
                                        aE(m, n) = pi * Kal * (delyg + dely) * (Rc / delrc + (m +
1) - Nrp - 1 / 2)
                                        aW(m, n) = pi * Kal * (delyg + dely) * (Rc / delrc + (m +
1) - Nrp - 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        ElseIf n = Ny + Nyg + 1 Then ' Temperature set Collet Bottom, No Corner
"J Eqns"
                                If m > Nrp + 2 Then ' To avoid left corner
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
                                        aSo(m, n) = 0
                                        aE(m, n) = pi * Kal * delyg * (Rc / delrc + (m + 1) - Nrp
- 1 / 2)
                                        aW(m, n) = pi * Kal * delyg * (Rc / delrc + (m + 1) - Nrp
- 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        ElseIf n < Ny + 1 Then
                                If m < Nrp + 1 Then ' Plug body "A Eqns"
                                        aN(m, n) = Kal * 2 * pi * (m + 1) * delr ^ 2 / dely
                                        aSo(m, n) = Kal * 2 * pi * (m + 1) * delr ^ 2 / dely
                                        aE(m, n) = Kal * dely * pi * (2 * (m + 1) + 1)
                                        aW(m, n) = Kal * dely * pi * (2 * (m + 1) - 1)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                ElseIf m > Nrp + 2 Then ' Collet above interface "M Eqns"
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                        aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / dely
                                        aE(m, n) = 2 * pi * Kal * dely * (Rc / delrc + (m + 1) -
Nrp - 1 / 2)
                                        aW(m, n) = 2 * pi * Kal * dely * (Rc / delrc + (m + 1) -
Nrp - 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        Else
                                If m > Nrp + 2 Then ' Gap area of Collet "N Eqns"
                                        aN(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
                                        aSo(m, n) = 2 * Kal * pi * (Rc * delrc + delrc ^ 2 * (m +
1) - Nrp * delrc ^ 2 - delrc ^ 2) / delyg
                                        aE(m, n) = 2 * pi * Kal * delyg * (Rc / delrc + (m + 1) -
Nrp - 1 / 2)
                                        aW(m, n) = 2 * pi * Kal * delyg * (Rc / delrc + (m + 1) -
Nrp - 3 / 2)
                                        aP(m, n) = aN(m, n) + aSo(m, n) + aE(m, n) + aW(m, n)
                                End If
                        End If
                End If
        Next
Next


' Initialize Temperatures
For m = 2 To Nrp + 1
        For n = 2 To Ny + 1
        T(m, n) = Thot ' Plug Temperature initialized at Thot
        Next
Next

For m = 2 To Nrc + 1
        For n = 2 To Ny + Nyg + 1
                T(m + Nrp, n) = Tcold  ' Collet Temperature initialized at Tcold
        Next
Next


'Gauss Seidel Solver
Dim it As Integer
Dim i As Integer

For it = 1 To itmax
```

```
    For n = 2 To Nyt + 1
        For m = 2 To Nrt + 1
            If n = 2 Then
                If m <= Nrp + 1 Then
                    T(m, n) = Thot
                Else
                    T(m, n) = (aE(m, n) * T(m + 1, n) + aW(m, n) * T(m - 1, n) + aN(m, n) * T(m,
n - 1) + aSo(m, n) * T(m, n + 1)) / aP(m, n)
                End If
            ElseIf n = Ny + Nyg + 1 Then
                If m >= Nrp + 2 Then
                    T(m, n) = Tcold
                End If
            Else
                T(m, n) = (aE(m, n) * T(m + 1, n) + aW(m, n) * T(m - 1, n) + aN(m, n) * T(m, n -
1) + aSo(m, n) * T(m, n + 1)) / aP(m, n)
            End If
        Next
    Next
Next

'sum the temperatures in a row across narrow section of the collet
Dim Ti As Double
Ti = 0

For i = Nrp + 2 To Nrp + Nrc + 1
    Ti = Ti + T(i, Ny + 1)
Next

'Compute the total heat flow through the part and the total resistance
Q = Kal * pi * (Ro * Ro - Rc * Rc) * (Ti / Nrc - Tcold) / Gap
Rplg = Lp / Kal / pi / Rp / Rp
Rjnt = (Thot - Tcold) / Q
Rstm = Ls / Kal / pi / Ro / Ro
Rtot = Rplg + Rjnt + Rstm



'Write to Spreadsheet
Sheets("NHTModel").Range("M13").Value = Rp   'xlswrite('CompliantHarvester', Rp, 'Matlab', 'B13')
Sheets("NHTModel").Range("M14").Value = Rplg 'xlswrite('CompliantHarvester', Rplg, 'Matlab',
'B14')
Sheets("NHTModel").Range("M15").Value = Rjnt 'xlswrite('CompliantHarvester', Rjnt, 'Matlab',
'B15')
Sheets("NHTModel").Range("M16").Value = Rstm 'xlswrite('CompliantHarvester', Rstm, 'Matlab',
'B16')
Sheets("NHTModel").Range("M18").Value = Rtot 'xlswrite('CompliantHarvester', Rtot, 'Matlab',
'B18')



'Output the T array
For m = 1 To 21
    For n = 1 To 26
           Sheets("TemperatureProfile").Cells(n, m).Value = T(m, n)
    Next
Next

End Sub
```