Laplace and Poisson's Equations

Thermal Behaviour of the Secondary Rocket Stage

Dikshita Kothari
500539170

Date: 27th May 2022

In this report, the thermal behaviour of a simplified model of the secondary rocket stage of the SpaceX Falcon-9 is analysed. A schematic of the rocket stage and simplified model is as shown below.

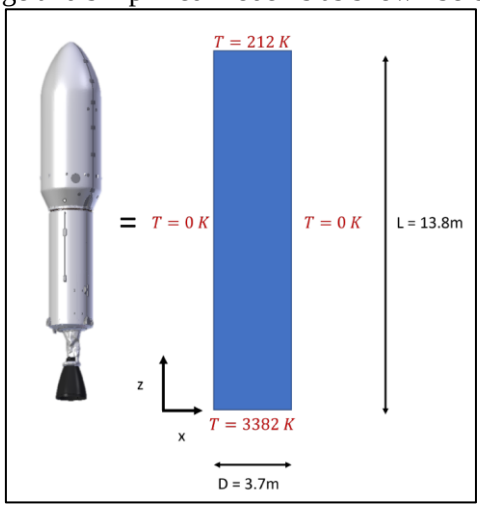

Figure 1. Schematic of the SpaceX Falcon 9 stage 2 rocket and its simplified model.

Section 1: Analytical Solution to the Laplace Equation

In this section, an analytical solution to the Laplace equation is developed. The temperature at each point in the rocket is evaluated. This temperature (T), can be modelled using the governing equation given by,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} = 0$$

This can be written as,

$$T_{xx} + T_{zz} = 0$$

1. Temperature Boundary Conditions

The temperature boundary conditions are illustrated in the given simplified model of the stage 2 rocket. They can be mathematically stated as follows.

For the bottom boundary,

$$T(x, 0) = T_0 = 3382\ K$$

For the top boundary,

$$T(x, L) = T_1 = 212\ K$$

Where L is the length of the rocket (L = 13.8 m).

For the left boundary,

$$T(0, z) = 0\ K$$

For the right boundary,

$$T(D, z) = 0\ K$$

Where D is the diameter of the rocket (D = 3.7 m).

2. Steady State Temperature Distribution

The steady state temperature distribution can be computed by solving two simpler component problems. This rocket model has two non-zero boundaries. The solution for these two non-zero boundaries can be computed separately and then added to give the final solution. This has been illustrated in the figure below.

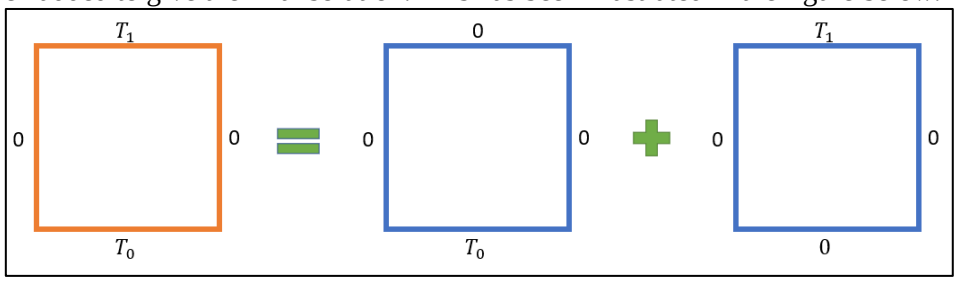

Figure 2. Illustration of complete solution as a sum of two simpler component problems.

First, the solution for the top non-zero boundary ($T_1 = 212\ K$) is computed. Then the z coordinates are adjusted to obtain the solution for the bottom non-zero boundary ($T_0 = 3382\ K$). These two solutions are then added together to give the final solution for the given problem.

3. Solution of Top Boundary Component

This is the solution of the top non-zero boundary component problem where the temperature is 212 K at the top boundary given by z = 13.8 m.

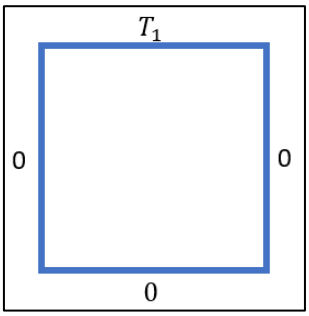

Figure 3. Illustration of top non-zero boundary component problem. T_1 = 212 K.

Boundary Conditions

The boundary conditions can be mathematically stated as follows.
For the top boundary,

$$T(x, L) = T_1 = 212\ K$$

Where L is the length of the rocket (L = 13.8 m).
For the bottom boundary,

$$T(x, 0) = 0\ K$$

For the left boundary,

$$T(0, z) = 0\ K$$

For the right boundary,

$$T(D, z) = 0\ K$$

Where D is the diameter of the rocket (D = 3.7 m).

Separation of Variables

The solution can be represented as a product of two functions in the x and z spatial domain.

$$T(x, z) = F(x) \times G(z)$$

Taking double derivative with respect to x we get,

$$T_{xx} = G \times F_{xx}$$

And with respect to z we get,

$$T_{zz} = F \times G_{zz}$$

The governing equation can then be written as,

$$T_{xx} + T_{zz} = G \times F_{xx} + F \times G_{zz} = 0$$

Rearranging the above equation we get,

$$\frac{F_{xx}}{F} = -\frac{G_{zz}}{G} = -p^2$$

This equation holds true for all x and z. The left-hand side is independent of z and the right-hand side is independent of x. Therefore, each side must be a constant. This is a non-trivial solution as $-p^2$ will always be a negative constant.
Rearranging, we get two equations one in x,

$$F_{xx} + p^2 F = 0$$

And the other in z,

$$G_{zz} - p^2 G = 0$$

Solution to the ODEs

A possible solution to the ODE in x is,

$$F(x) = a\cos(px) + b\sin(px)$$

At x = 0, F(x) is zero as this is the left boundary. This implies that a = 0.
At x = D, F(x) is zero as this is the right boundary. This implies that, $b\sin(pD) = 0$.
For this to be true,

$$p = \frac{n\pi}{D}$$

Where n is the mode.
Therefore,

$$F(x) = b\sin px$$

For each mode,

$$F_n(x) = b_n \sin px = b_n \sin \frac{n\pi x}{D}$$

A possible solution to the ODE in z for each mode is,

$$G_n(z) = A_n e^{pz} + B_n e^{-pz}$$

At z = 0, G(z) is zero as this is the bottom boundary. This implies that,

$$A_n + B_n = 0$$

Therefore,

$$A_n = -B_n$$

The solution to the ODE in z for each mode is therefore,

$$G_n(z) = A_n e^{pz} - A_n e^{-pz} = 2A_n \sinh\left(\frac{n\pi z}{D}\right)$$

Complete Solution

The complete solution can be written as a product of two functions in the x and z spatial domain.

$$T(x,z) = F(x) \times G(z)$$

The two functions can be written as a sum of different modes.

$$T(x,z) = \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{D}\right) \times 2A_n \sinh\left(\frac{n\pi z}{D}\right)$$

The coefficient is defined as $A_n^* = 2 \times A_n \times b_n$ and substituted in the equation,

$$T(x,z) = \sum_{n=1}^{\infty} A_n^* \sinh\left(\frac{n\pi z}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

As the non-zero boundary is at z = L, the equation can be written as,

$$T(x,L) = \sum_{n=1}^{\infty} A_n^* \sinh\left(\frac{n\pi L}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

The Fourier coefficient is now defined as,

$$B_n^* = A_n^* \times \sinh\left(\frac{n\pi L}{D}\right)$$

Therefore, we obtain an expression for the temperature at the top boundary.

$$T(x,L) = \sum_{n=1}^{\infty} B_n^* \sin\left(\frac{n\pi x}{D}\right) = T_1$$

The Fourier coefficient can be evaluated from the boundary condition. The boundary condition is illustrated below. Here D is the diameter and T_1 is the temperature 212 K.

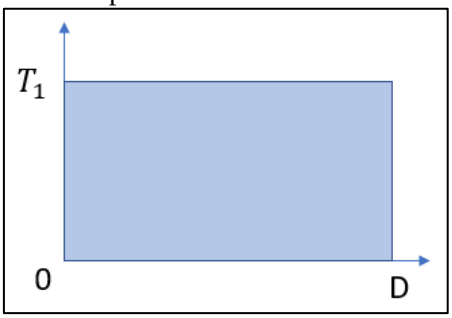

Figure 4. Illustration of the boundary condition at the top boundary.

Therefore, the Fourier coefficient can be written as,

$$B_n^* = \frac{2T_1(1 - \cos n\pi)}{n\pi}$$

Where n is the mode. The coefficient A_n^* can now be written as,

$$A_n^* = \frac{2T_1(1 - \cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)}$$

The complete solution for the top side is given as,

$$T(x,z) = \sum_{n=1}^{\infty} \frac{2T_1 \times (1 - \cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)} \sinh\left(\frac{n\pi z}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

Where, T_1 is the temperature at the top boundary, 212 K, L is the length, 13.8 m, D is the diameter, 3.7 m and n is the mode.

4. Solution of Bottom Boundary Component

The bottom side solution can be obtained using a coordinate transformation as explained in section 1.2. To transform the coordinates a variable q is chosen such that q = z – L.

At z = 0, q = − L, and at z = L, q = 0.

The general solution in the q coordinate system is,

$$T(x,q) = \sum_{n=1}^{\infty} A_n^* \sinh\left(\frac{n\pi q}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

The non-zero value is at z = 0 or q = − L. The temperature at this boundary can be given by,

$$T(x,-L) = \sum_{n=1}^{\infty} A_n^* \sinh\left(\frac{n\pi(-L)}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

The Fourier coefficient is now defined as,

$$B_n^* = A_n^* \times \sinh\left(\frac{n\pi(-L)}{D}\right)$$

Therefore, we obtain an expression for the temperature at the bottom boundary.

$$T(x,-L) = \sum_{n=1}^{\infty} B_n^* \sin\left(\frac{n\pi x}{D}\right) = T_0$$

The Fourier coefficient can be evaluated from the boundary condition. The boundary condition is illustrated below. Here D is the diameter and T_0 is the temperature 3382 K.

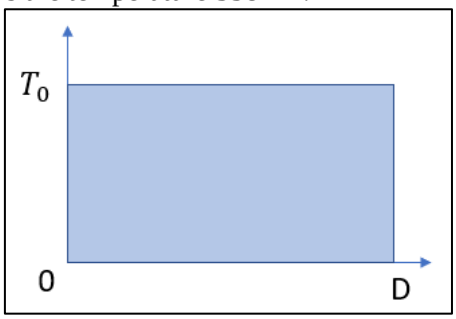

Figure 5. Illustration of the boundary condition at the bottom boundary.

Therefore, the Fourier coefficient can be written as,

$$B_n^* = \frac{2T_0(1-\cos n\pi)}{n\pi}$$

Where n is the mode. The coefficient A_n^* can now be written as,

$$A_n^* = \frac{-2T_0(1-\cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)}$$

Therefore, the general solution in the q coordinate system is,

$$T(x,q) = \sum_{n=1}^{\infty} \frac{-2T_0(1-\cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)} \sinh\left(\frac{n\pi q}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

The solution in the z coordinate system is therefore,

$$T(x,z) = \sum_{n=1}^{\infty} \frac{-2T_0 \times (1-\cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)} \sinh\left(\frac{n\pi(z-L)}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

Where, T_0 is the temperature at the bottom boundary, 3382 K, L is the length, 13.8 m, D is the diameter, 3.7 m and n is the mode.

The complete solution is a summation of the top side and bottom side solutions. It is given by the equation below.

$$T(x,z) = \sum_{n=1}^{\infty} \frac{2T_1 \times (1-\cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)} \sinh\left(\frac{n\pi z}{D}\right) \sin\left(\frac{n\pi x}{D}\right) + \sum_{n=1}^{\infty} \frac{-2T_0 \times (1-\cos n\pi)}{n\pi \times \sinh\left(\frac{n\pi L}{D}\right)} \sinh\left(\frac{n\pi(z-L)}{D}\right) \sin\left(\frac{n\pi x}{D}\right)$$

Where, T_0 is the temperature at the bottom boundary, 3382 K, L is the length, 13.8 m, D is the diameter, 3.7 m and n is the mode.

5. MATLAB Implementation

The derived solution was implemented on MATLAB. The grid is chosen so that the temperature is sampled at every centimetre in both x and z directions. 60 Fourier terms are used for the implementation.

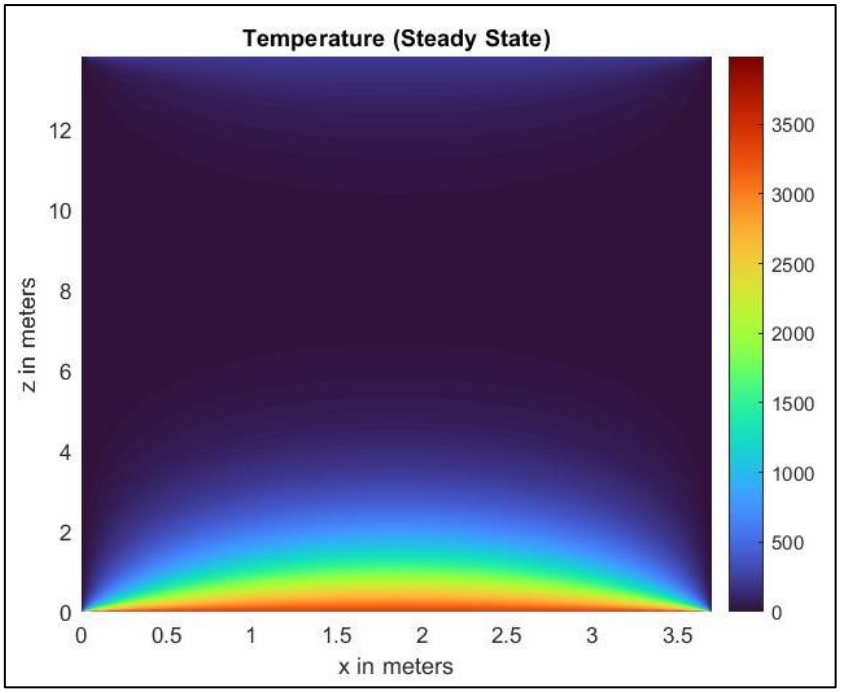

Figure 6. Labelled contour plot of the analytically obtained steady state temperature distribution.

The graph is as expected. The temperature at the top boundary is represented by deep red/red which indicates a temperature of 3382 K. The temperature at the bottom boundary is represented by blue which indicates a temperature of 212 K. The right and left boundaries are at deep blue/black indicating 0 K temperature.

Section 2: Numerical Solution to the Laplace Equation

In this section, a Gauss Seidel numerical solution to the Laplace equation is analysed. The temperature at each point in the rocket is evaluated. This temperature (T), can be modelled using the Gauss Seidel stencil given by,

$$T_{i,j}^{n+1} = \alpha\left(T_{i+1,j}^{n} + T_{i-1,j}^{n+1}\right) + \gamma\left(T_{i,j+1}^{n} + T_{i,j-1}^{n+1}\right)$$

Where α and γ are regularisation parameters given by,

$$\alpha = \frac{\Delta z^2}{2\Delta x^2 + 2\Delta z^2}$$

$$\gamma = \frac{\Delta x^2}{2\Delta x^2 + 2\Delta z^2}$$

And Δx and Δz are grid sizes in x and z direction, respectively.

1. Order of Accuracy

The order of accuracy in both directions will be the same as the stencils are identical in both directions. The order of accuracy in both directions are calculated and tabulated below. For simplicity the forward in time is ignored. Central in Space stencil is used in both the x and z directions.

The order of accuracy is derived for the x direction. The Taylor series forward in space expansion can be written as,

$$T_{i+1} = T_i + \Delta x T_{i_x} + \frac{\Delta x^2}{2!} T_{i_{xx}} + \frac{\Delta x^3}{3!} T_{i_{xxx}} + \frac{\Delta x^4}{4!} T_{i_{xxxx}} \dots$$

The Taylor series backward in space expansion can be written as,

$$T_{i-1} = T_i - \Delta x T_{i_x} + \frac{\Delta x^2}{2!} T_{i_{xx}} - \frac{\Delta x^3}{3!} T_{i_{xxx}} + \frac{\Delta x^4}{4!} T_{i_{xxxx}} \dots$$

Adding both these equations we get,

$$T_{i+1} + T_{i-1} = 2T_i + 2\frac{\Delta x^2}{2!} T_{i_{xx}} + 2\frac{\Delta x^4}{4!} T_{i_{xxxx}} \dots$$

Rearranging we get,

$$T_{i_{xx}} = \frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2} - \frac{\Delta x^2}{2} T_{i_{xxxx}} - \cdots$$

The higher orders of Δx are truncated in the central approximation which is,

$$T_{i_{xx}} = \frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2} - O(\Delta x^2)$$

This is known as truncation error. Therefore, the central scheme in x direction is 2nd order accurate.

Similarly in the z direction,

$$T_{k_{zz}} = \frac{T_{k+1} + T_{k-1} - 2T_k}{\Delta z^2} - O(\Delta z^2)$$

Therefore, the central scheme in the z direction is 2nd order accurate.
The orders of accuracy for the two stencils are shown in the table below:

Direction	Scheme	Order of Accuracy
X	Central in Space	2^{nd}
Z	Central in Space	2^{nd}

Table 1. Order of Accuracy of the given stencil.

2. Sequence of Steps for MATLAB Implementation

The sequence of steps to implement the given Gauss Seidel stencil in MATLAB is given below.
Steps:

1. Start
2. Declare parameters. Length L = 13.8 m. Diameter D = 3.7 m.
3. Declare boundary conditions. Temperature of the top boundary TT = 3382 K. Temperature of the bottom boundary TB = 212 K. Temperature of the left boundary TL = 0 K. Temperature of the right boundary TR = 0 K.
4. Declare n = 2.
5. Calculate $n_{grid} = n^2 + 1$. Assign nx = nz = n_{grid}.
6. Create row matrices x and z with ngrid elements each.
7. Create empty matrix Tnp1 which has nx number of columns and ny number of rows.
8. Calculate grid spacing. Grid spacing in x direction is calculated as dx = D/(nx-1). Grid spacing in z direction is calculated as dz = L/(nz-1).
9. Initialise error as some large number much greater than 1.
10. Set tolerance to 1×10^{-10}.
11. Set alpha as, $\alpha = \frac{\Delta z^2}{2\Delta x^2+2\Delta z^2}$
    And gamma as, $\gamma = \frac{\Delta x^2}{2\Delta x^2+2\Delta z^2}$
12. If error is greater than tolerance, continue. Else skip to 25.
13. Update solution array for this timestep as Tn = Tnp1.
14. Set i = 2.
15. Set j = 2.
16. Calculate Tnp1 element j^{th} row and i^{th} column as,
$$T_{i,j}^{n+1} = \alpha\left(T_{i+1,j}^{n} + T_{i-1,j}^{n+1}\right) + \gamma\left(T_{i,j+1}^{n} + T_{i,j-1}^{n+1}\right)$$
17. If j less than or equal to nz – 1, continue. Else skip to 20.
18. Update j as j+1.
19. Go to 16.
20. If i less than or equal to nx – 1, continue. Else skip to 23.
21. Update i as i+1.
22. Go to 15.
23. Calculate error as L1 norm. Update error.
24. Go to 12.
25. Set output variable T as Tnp1.
26. Repeat steps 5 to 25 for n = 3,4,5,6,7 and 8.

It is possible to check if the solution has converged by checking the error calculated against the tolerance value chosen.

3. Number of Iterations

The MATLAB solution was implemented for different grid sizes. The number of iterations required for the solution to converge for different grid sizes is tabulated below.

n	n_{grid}	Iterations
2	5	57
3	9	241
4	17	956
5	33	3673
6	65	14028
7	129	53912
8	257	206481

Table 2. Number of iterations required for the solution to converge at different grid sizes.

It can be seen from the table that the number of iterations increases with increasing grid size. The iteration values were plotted against n for further analysis. The iterations were plotted on a logarithmic y axis and the n value was plotted on the x axis.

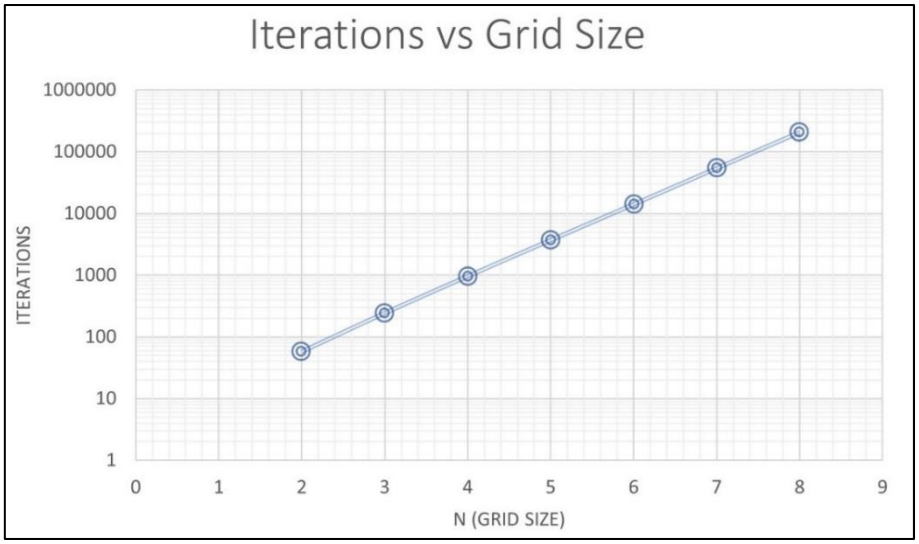

Figure 7. Relationship between iterations and n.

It is clear from the graph that the iterations and n have a logarithmic relationship. That means that the number of iterations required for the solution to converge increases exponentially with increasing n (grid size).

4. Numerically Determined Steady State Temperature Distribution

A labelled contour plot of the numerically determined steady-state temperature distribution for the $n_{grid} = 2^8 + 1$ case is presented below.

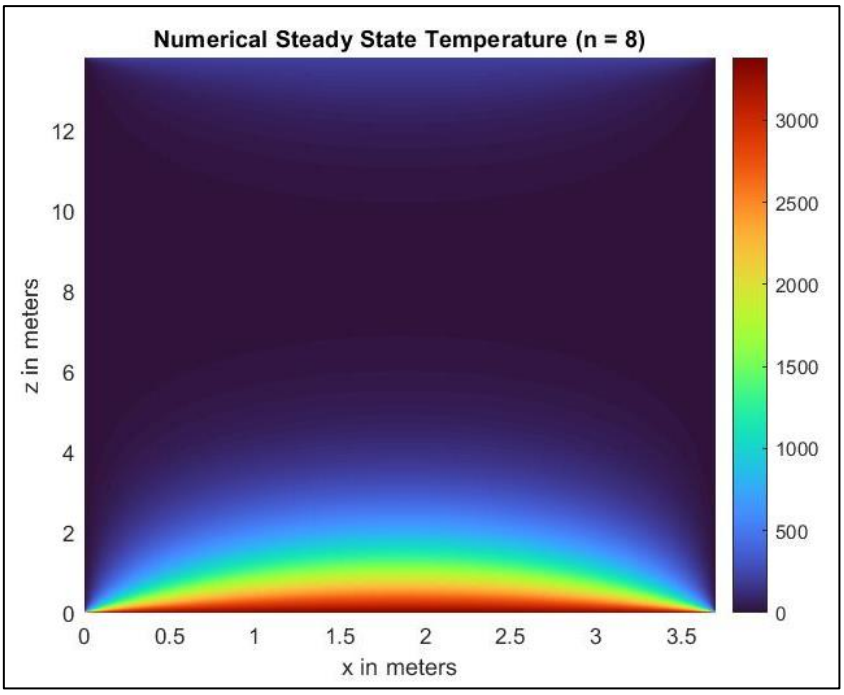

Figure 8. Labelled contour plot of the numerically obtained steady state temperature distribution.

The graph is as expected. The temperature at the top boundary is represented by deep red/red which indicates a temperature of 3382 K. The temperature at the bottom boundary is represented by blue which indicates a temperature of 212 K. The right and left boundaries are at deep blue/black indicating 0 K temperature.
There is no observable difference between the analytical and numerical contour plot at n = 8. This indicates that this grid size is appropriate to approximate the analytical solution.

5. Convergence Analysis

The method of Divided Differences is used to compute the order of accuracy. The divided difference method compares the change of a variable at a control point across three different grid sizes. The control point is chosen as T(1.85, 6.9). It can be given by the equation below.

$$n = \log_2\left(\frac{T(2\Delta x) - T(4\Delta x)}{T(x) - T(2\Delta x)}\right)$$

Where n is the order of accuracy. The table below shows the control point values of different grid sizes and the corresponding order of accuracy, n.

Grid Size, n_{grid}	**Control Point, T(1.85, 6.9)**	**Order of Accuracy, n**
5	42.6505	-
9	20.3656	-

17	14.8665	2.0188
33	13.5134	2.0229
65	13.1769	2.0076
129	13.0929	2.0021
257	13.0719	2.0000

Table 3. Divided difference method for convergence analysis.

The given numerical scheme is 2nd order accurate. From the table it is evident that although the grid sizes 17 through 129 have accuracy close to 2nd order, grid size 257 is exactly 2nd order accurate. This means that the solution converges at n = 8 or n_{grid} = 257.

Section 3: Numerical Solution to Poisson's Equation

In this section, the numerical thermal problem is extended to an axial deformation problem which occurs due to thermal loading. One-dimensional axial deformation of a thermally loaded object with constant material and geometric properties is given by the equation below.

$$w_{zz} - \beta T_z = 0$$

Where β is the thermal expansion coefficient and w is the z-axis deformation.
The SpaceX rocket is modelled with $\beta = 21 \times 10^6$ m/K.

1. Average One-Dimensional Temperature Profile

The average 1-D temperature profile is computed along z. To obtain these values, the mean temperature along x for each z value is calculated.

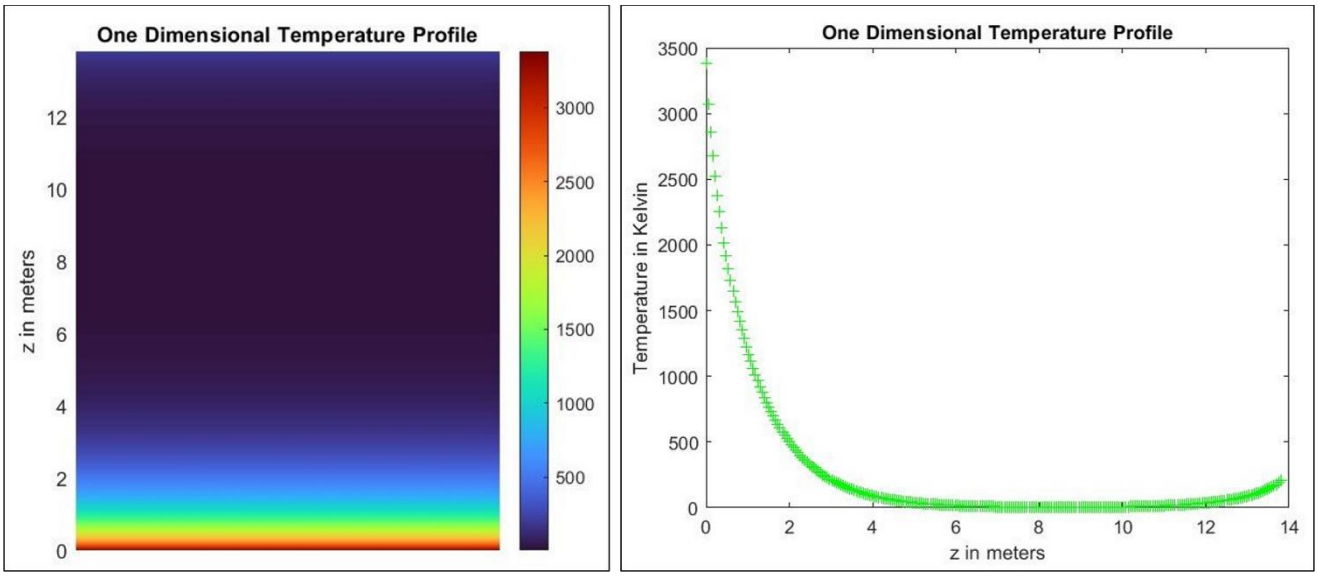

Figure 9. (Left) Labelled contour plot of 1-D temperature profile.
(Right) Graph showing mean temperature along the length for each x.

The graph is as expected with an average temperature of 3382 K at the top boundary and an average temperature of 212 K at the bottom boundary. The maximum average temperature is observed at the top boundary. The minimum average temperature (3.9060 K) is observed at a length of 5.33 m from the top boundary.

2. Derivation of One-Dimensional Gauss Seidel Stencil

One-dimensional axial deformation of a thermally loaded object with constant material and geometric properties is given by the equation below.

$$w_{zz} - \beta T_z = 0$$

Where β is the thermal expansion coefficient and w is the z-axis deformation.
The Taylor series forward in space expansion for the z-axis deformation can be written as,

$$w_{k+1} = w_k + \Delta z w_{k_z} + \frac{\Delta z^2}{2!} w_{k_{zz}} + \frac{\Delta z^3}{3!} w_{k_{zzz}} + \frac{\Delta z^4}{4!} w_{k_{zzzz}} \dots$$

The Taylor series backward in space expansion can be written as,

$$w_{k-1} = w_k - \Delta z w_{k_z} + \frac{\Delta z^2}{2!} w_{k_{zz}} - \frac{\Delta z^3}{3!} w_{k_{zzz}} + \frac{\Delta z^4}{4!} w_{k_{zzzz}} \dots$$

Adding both these equations we get,

$$w_{k+1} + w_{k-1} = 2w_k + 2\frac{\Delta z^2}{2!} w_{k_{zz}} + 2\frac{\Delta z^4}{4!} w_{k_{zzzz}} \dots$$

Rearranging we get,

$$w_{k_{zz}} = \frac{w_{k+1} + w_{k-1} - 2w_k}{\Delta z^2} - \frac{1}{\Delta z^2}\frac{\Delta z^4}{2} w_{k_{zzzz}} - \cdots$$

The higher orders of Δz are truncated in the central approximation which is,

$$w_{k_{zz}} = \frac{w_{k+1} + w_{k-1} - 2w_k}{\Delta z^2} - O(\Delta z^2)$$

This is known as truncation error. The central scheme in z direction is 2nd order accurate.

The Taylor series forward in space expansion for temperature can be written as,

$$T_{k+1} = T_k + \Delta z T_{k_z} + \frac{\Delta z^2}{2!} T_{k_{zz}} + \frac{\Delta z^3}{3!} T_{k_{zzz}} + \frac{\Delta z^4}{4!} T_{k_{zzzz}} \cdots$$

The Taylor series backward in space expansion for temperature can be written as,

$$T_{k-1} = T_k - \Delta z T_{k_z} + \frac{\Delta z^2}{2!} T_{k_{zz}} - \frac{\Delta z^3}{3!} T_{k_{zzz}} + \frac{\Delta z^4}{4!} T_{k_{zzzz}} \cdots$$

Subtracting the second equation from the first we get,

$$T_{k+1} - T_{k-1} = 2\Delta z T_{k_z} + 2\frac{\Delta z^3}{3!} T_{k_{zzz}} + \cdots$$

Rearranging we get,

$$T_{k_z} = \frac{T_{\mathrm{k+1}} - T_{k-1}}{2\Delta \mathrm{z}} - \frac{1}{2\Delta \mathrm{z}} \frac{\Delta z^3}{3!} T_{k_{zz}} - \cdots$$

The higher orders of Δz are truncated in the forward approximation which is,

$$T_{k_z} = \frac{T_{\mathrm{k+1}} - T_{k-1}}{2\Delta \mathrm{z}} - O(\Delta z^2)$$

This is known as truncation error. The central difference scheme in z direction is 2nd order accurate.

We use the given equation, the central in space scheme for z-axis deformation and the central difference scheme for temperature to calculate the z-axis deformation at a particular point.

The given equation is,

$$w_{zz} - \beta T_z = 0$$

Substituting the central in space scheme for z-axis deformation and the forward in space scheme for temperature we get,

$$\frac{w_{k+1} + w_{k-1} - 2w_k}{\Delta z^2} = \beta \frac{T_{\mathrm{k+1}} - T_{k-1}}{2 \times \Delta \mathrm{z}}$$

Rearranging we get,

$$w_k = \frac{w_{k+1} + w_{k-1}}{2} - \beta \times \Delta \mathrm{z} \times \frac{T_{\mathrm{k+1}} - T_{k-1}}{4}$$

We can alter the above expression by considering the time steps to obtain the Gauss Seidel expression.

$$w_k^{n+1} = \frac{w_{k+1}^n + w_{k-1}^{n+1}}{2} - \beta \times \Delta \mathrm{z} \times \frac{T_{\mathrm{k+1}} - T_{k-1}}{4}$$

This is not valid at the Neumann Boundary due to the given boundary condition. The scheme at the Neumann Boundary is derived below.

The Taylor series backward in space expansion can be written as,

$$w_{k-1} = w_k - \Delta z w_{k_z} + \frac{\Delta z^2}{2!} w_{k_{zz}} - \frac{\Delta z^3}{3!} w_{k_{zzz}} + \frac{\Delta z^4}{4!} w_{k_{zzzz}} \cdots$$

The Neumann Boundary condition states that $\frac{\partial w}{\partial z} = 0$. This implies that all higher order derivatives of w_k will also be zero. Substituting these values in the above equation and rearranging we get,

$$w_k = w_{k-1}$$

We can alter the above expression by considering the time steps to obtain the Gauss Seidel expression.

$$w_k^{n+1} = w_{k-1}^{n+1}$$

The numerical stencil for this problem is shown below. It has two forms depending upon whether you are at the Neumann boundary (at z = L) or within the rest of the domain. This is necessary to enforce a gradient at z = L rather than simply defining a value.

$$w_k^{n+1} = \begin{cases} \frac{w_{k+1}^n + w_{k-1}^{n+1}}{2} - \beta \times \Delta \mathrm{z} \times \frac{T_{\mathrm{k+1}} - T_{k-1}}{4} & k = 2{:}nz-1 \\ w_{k-1}^{n+1} & k = nz \end{cases}$$

The obtained scheme is 2nd order accurate.

3. z – Displacement for the rocket

The 1-D temperature profile is as obtained in section 3.1. The SpaceX rocket is modelled with $\beta = 21 \times 10^6$ m/K which is the thermal expansion coefficient. w is the z-axis deformation. The domain has a Dirichlet boundary condition at z = 0 which is w = 0 and a Neumann boundary condition at z = L which is $\frac{\partial w}{\partial z} = 0$.

The stencil is as derived previously.

$$w_k^{n+1} = \begin{cases} \frac{w_{k+1}^n + w_{k-1}^{n+1}}{2} - \beta \times \Delta \mathrm{z} \times \frac{T_{\mathrm{k+1}} - T_{k-1}}{4} & k = 2{:}nz-1 \\ w_{k-1}^{n+1} & k = nz \end{cases}$$

The error value used is 1×10^{10}.

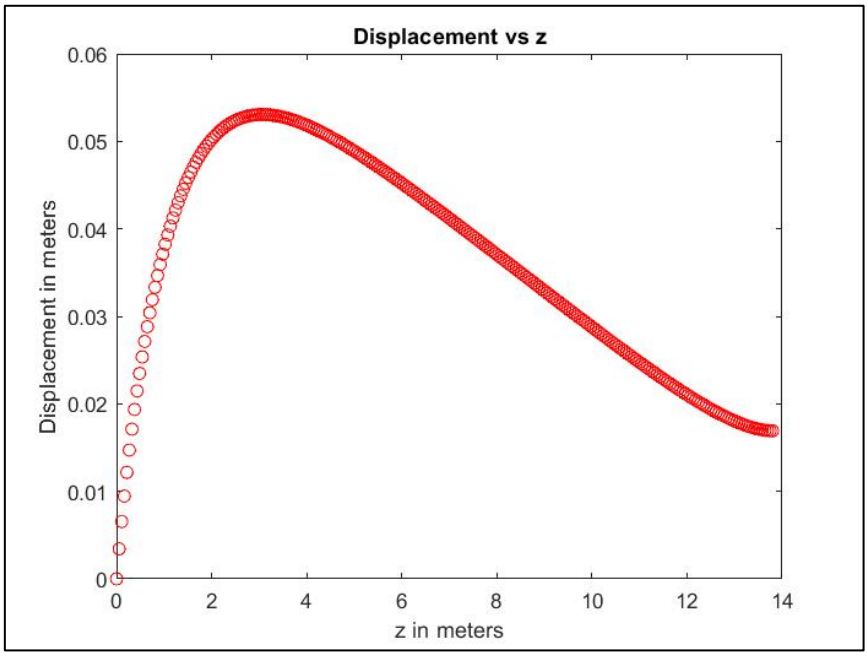

Figure 10. Graph showing z-axis displacement.

It can be seen from figure 10 that the boundary conditions are as required. The displacement at the bottom boundary (z = 0 m) is zero. The gradient of displacement at the top boundary (z = 13.8 m) is zero. The maximum displacement is 0.0531 m which occurs at 3.13 meters from the bottom boundary. The maximum displacement is 0.385% of the total length of the rocket. This is an acceptable value. Minimum displacement occurs at maximum temperature.

4. Thermal Stress

The thermal stress on the rocket can be calculated using the following equation.

$$\sigma = E \times w_z$$

Where E is the Young's Modulus and is equal to 78 GPa.

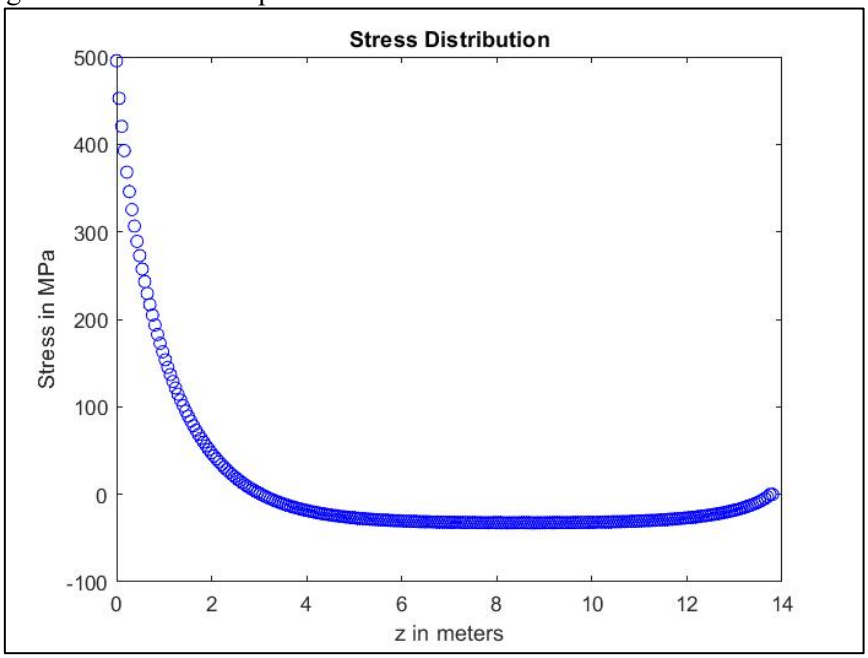

Figure 11. Stress Distribution along z axis.

The plot of stress distribution along the z-axis is analogous to the one-dimensional temperature distribution shown in figure 9. The maximum stress (495.4352 MPa) is observed at the point with maximum temperature at the bottom boundary and the minimum stress (-32.4936 MPa) is observed at the point with minimum temperature at a length of 5.33 m from the top boundary.

5. Thermal Load

The stress has a maximum value of 495.4352 MPa and a minimum value of -32.4936 MPa. These values are within the allowable $\pm$600MPa range. The stress distribution or thermal load on the rocket therefore does not propose a problem.

APPENDIX

Analytical Solution

```
function Tss = laplaceanalytical()

% File name: laplaceanalytical
%
% Description: This file is the analytical solution to the Laplace equation describing the
temperature profile for the rocket.
%
% Input parameters:
% T0       - concentration at the bottom boundary
% T1       - concentration at the top boundary
%
%              T1
%           _____________
%         |               |
%         |               |
%     0   |               |  0
%         |               |
%         |_______________|
%
%              T0
%

% Spatial parameters
L = 13800;                                          % Length of rocket (z dimension)
D = 3700;                                           % Diameter of rocket (x dimension)
dL = 1;                                             % Spatial step size in z
dD = 1;                                             % Spatial step size in x
nL = round((L/dL)+1);                              % Number of grid points in z
nD = round((D/dD)+1);                               % Number of grid points in x
z = 0:dL:L;                                         % Vector of grid points in z
x = 0:dD:D;                                         % Vector of grid points in x
nfs = 60;                                           % Number of Fourier terms

% Temperatures
T0 = 212;                                       % Temperature at bottom boundary
T1 = 3382;                                      % Temperature at top boundary


% Calculate Fourier Coefficients
B0 = zeros(1,nfs);                              % Bottom side
B1 = zeros(1,nfs);                              % Top side
for m=1:nfs
    B0(m) = (2*T0)/(m*pi)*(1-cos(m*pi));    % Bottom side
    B1(m) = (2*T1)/(m*pi)*(1-cos(m*pi));    % Top side
end

% Initialise solution array
Tss = zeros(nD,nL);

% Analytical solution
for i=1:nD
   for j=1:nL
      for m=1:nfs
         % Fill in the analytical solution here
         Tss(i,j) = Tss(i,j) + B1(m)*(sinh((m*pi*z(j))/D)/sinh((m*pi*L/D))*sin(m*pi*x(i)/D)) +
B0(m)*(sinh((m*pi*(z(j)-L))/D)/sinh((m*pi*-L/D))*sin(m*pi*x(i)/D));
      end
   end
end

% Display the steady-state result
```

```
pcolor(x,z,Tss'),shading interp,title('Temperature (Steady
State)'),xlabel('x'),ylabel('z'),colorbar;

end
```

Laplace Numerical

```
function [X, Y, T, count] = LaplaceEquation( a, b, nx, ny, TL, TR, TT, TB )
% LaplaceEquation - Solves the 2D Laplace equation for domain 0 < x < a and
% 0 < y < b with nx and ny points in the x and y dimensions
% Input :
%    a - maximum value of the x domain [m]
%    b - maximum value of the y domain [m]
%    nx - number of points in the x domain
%    ny - number of points in the y domain
%    TL, TR, TT, TB - temperature at left, right, top, and bottom
%         boundaries [deg C]
% Output :
%    X - matrix of x-coordinate values for each point in the domain [m]
%    Y - matrix of y-coordinate values for each point in the domain [m]
%    T - steady state temperature distribution [deg C]

% Create domain
x = linspace(0, a, nx);
y = linspace(0, b, ny);
[X, Y] = meshgrid(x,y);

% Calculate grid spacing and maximum stable timestep
dx = a/(nx-1);
dy = b/(ny-1);
dt = min(dx,dy)^2/4;

% Initialise solution array for timestep n+1
% Note indexing: rows = y, columns = x to agree with meshgrid output
Tnp1 = zeros(ny, nx);

% Set boundary conditions
Tnp1(:,1) = TL;   % Left boundary
Tnp1(:,end) = TR; % Right boundary
Tnp1(1,:) = TB;   % Bottom boundary
Tnp1(end,:) = TT; % Top boundary

% Initialise error and set tolerance for convergence
err = 1;
tol = 1e-10;
count = 0;
alpha = 0.5*dy^2/(dx^2+dy^2);
gamma = 0.5*dx^2/(dx^2+dy^2);

while  err >= tol
    % Update solution array for this timestep
    Tn = Tnp1;

    count = count + 1;

    % Loop over internal points
    for i = 2:nx-1
        for j = 2:ny-1
            Tnp1(j,i) = (gamma*(Tn(j+1,i) + Tnp1(j-1,i)) + alpha*(Tn(j,i+1) + Tnp1(j,i-1)));
        end
    end

    % Compute error as maximum change in domain (absolute value)
    err = norm((Tnp1(:) - Tn(:)),1)*dx*dy;
end
```

```
% Set output variable
T = Tnp1;
end

clc
clear all

[X, Y, T, count] = LaplaceEquation( 3700, 13800, 257, 257, 0, 0, 3382, 212 );
check = T(257,257);
%%
% Display the steady-state result
pcolor(X,Y,T),shading interp,title('Numerical Steady State Temperature (n = 8)'),xlabel('x in
centimeters'),ylabel('z in centimeters'),colorbar;
%%
a = 3700;
b = 13800;
nx = 257;
ny = 257;
x = linspace(0, a, nx);
y = linspace(0, b, ny);
[X, Y] = meshgrid(x,y);
onedim = zeros(ny,1)
for i = 1:ny
    onedim(i,:) = mean(T(i,:));
end
size(onedim)
%%
plot(onedim, y, 'b+');
hold on
ylabel 'z in centimeters'
xlabel 'Temperature in Kelvin'
title 'One Dimensional Temperature Profile'
%%
pcolor(1:2, y, onedim),shading interp,title('One Dimensional Temperature Profile'),ylabel('z in
centimeters'),colorbar;
```

Poisson Numerical

```
function [z, W] = poisson( b, nz, WB , T)

% Create domain
z = linspace(0, b, nz);

% Calculate grid spacing
dz = b/(nz-1);

% Initialise solution array for timestep n+1
Wnp1 = zeros(1,nz);

% Set boundary conditions
Wnp1(1) = WB;   % Bottom boundary

% Initialise error and set tolerance for convergence
err = 1;
tol = 1e-10;

% Set parameters
beta = 21 * 10^-6;

while  err >= tol

    % Update solution array for this timestep
    Wn = Wnp1;

    % Loop over internal points
```

```
    for j = 2:nz-1
         Wnp1(j) = (Wn(j+1)+Wnp1(j-1))/2 - 0.25*beta*dz*(T(j+1)-T(j-1));
    end

    % For the Neumann boundary
    Wnp1(nz) = Wnp1(nz-1);

    % Compute error as maximum change in domain (norm L1)
    err = norm((Wnp1(:) - Wn(:)),1)*dz;
end

% Set output variable
W = Wnp1;
end
```

```
clc
clear
close all

onedim = importdata("onedim.mat")
b = 13.8;
nz = 257;
WB = 0;
T = onedim;

[z, W] = poisson( b, nz, WB , T);

% Calculate grid spacing
dz = b/(nz-1);

% Set equation parameters
E = 7800; % MPa

% Empty Matrix for solution
    Wz = zeros(1, nz);

% Loop over points to calculate stress
    for j = 1:nz-1
         Wz(j) = E*(W(j+1)-W(j))/dz;
    end

% For the Neumann boundary
    Wz(nz) = 0;

% Stress equation S=E*Wz in MPa
    %S = E*Wz;

% Plot the stress distribution
figure
plot(z,Wz,'bo')
hold on
title 'Stress Distribution'
xlabel 'z in meters'
ylabel 'Stress in MPa'
```