

AMME2000/BMET2960/BMET9960 - Assignment 2

Due: 11:59 pm Friday 4th June 2021

Introduction

A team is using drones to capture high resolution images of the terrain after recent volcanic activity in Iceland. However, the images are distorted due to variations in air pressure and density caused by the heating from the lava. The team has provided you with some data you can use to obtain a model of the pressure and density fields in the environment - which will then allow them to apply corrections to their drone images.

In this assignment you will use the analytical and numerical techniques developed in this course to obtain a steady-state temperature field in a 2D rectangular slice of the environment using the Laplace Equation. Using this information, you will then solve the Poisson equation for pressure. A schematic of the 2D environment is shown in Figure 1.

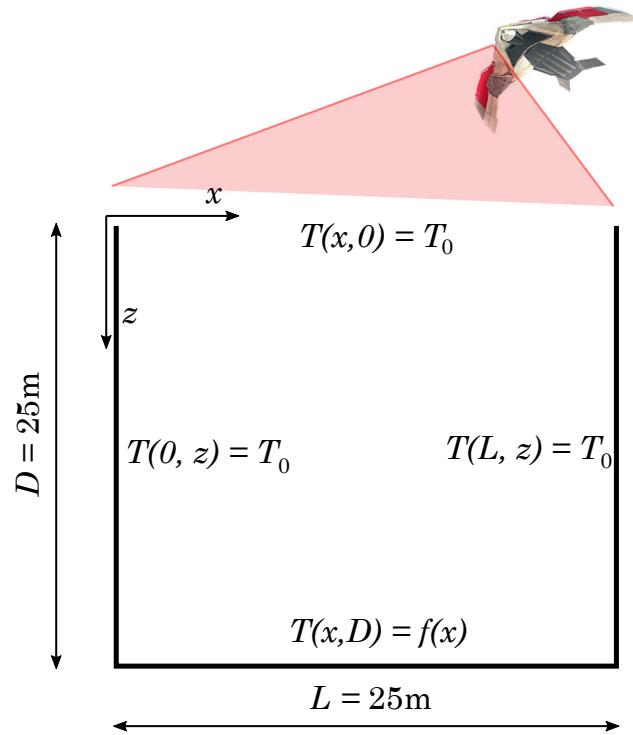


Figure 1: Schematic of the 2D domain.

1 Analytic Solution to the Steady-State Temperature Field (20%)

Taking x to be the horizontal position and z to be the depth (positive downwards) for the 2D domain, the boundary temperature distributions are defined as (Figure 1):

$$T(0, z) = T(L, z) = T(x, 0) = T_0 \quad (1)$$

$$T(x, D) = f(x) = T_0 + T_1 \sin(\pi x/L) + T_2 \sin(5\pi x/L) \quad (2)$$

where $L = 25\text{m}$ is the width of the environment in the x -direction, $D = 25\text{m}$ is the depth of the environment in the z -direction, and the temperatures are $T_0 = 20^\circ\text{C}$, $T_1 = 380^\circ\text{C}$ and $T_2 = 205^\circ\text{C}$. Given that the temperature field is at steady-state we can use the Laplace equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (3)$$

1. Assuming that the temperature distribution has the form $T(x, z) = F(x)G(z)$, use separation of variables to show that:

$$F(x) = A \cos(px) + B \sin(px) \quad (4)$$

$$G(z) = Ce^{pz} + De^{-pz} \quad (5)$$

(5%)

2. By recognising that T_0 appears on all four boundaries, derive the steady-state solution as:

$$T(x, z) = T_0 + \sum_{n=1}^{\infty} B_n \sin\left(\frac{n\pi x}{L}\right) \sinh\left(\frac{n\pi z}{L}\right) \quad (6)$$

Show your working and justify any assumptions you make. (10%)

3. Explain why Fourier coefficient solutions only exist for $n = 1$ and $n = 5$, and hence state the exact analytic solution for the steady-state temperature field. (5%)

2 Numerical Solution to the Temperature Field (40%)

In this section you will solve the temperature field numerically using two different methods: (i) Gauss-Seidel and (ii) Successive Over-Relaxation (SOR). We can discretise the spatial domain into nodes along x and z according to:

$$x_i = (i - 1)\Delta x \text{ where } i \in [1, n_x] \quad (7)$$

$$z_j = (j - 1)\Delta z \text{ where } j \in [1, n_z] \quad (8)$$

and denote the temperature at position (x_i, z_j) by $T_{i,j}$. Because we are considering a steady-state problem, time is not a variable, however we still need to iterate our numerical solution and so will use the superscript n to index the current solution iteration. In other words, $T_{i,j}^n$ represents the numerical temperature estimate at location (x_i, z_j) for iteration n .

The Gauss-Seidel (G-S) method computes successive grid point values according to:

$$T_{i,j}^{n+1} = \frac{1}{4} \left(T_{i+1,j}^n + T_{i-1,j}^{n+1} + T_{i,j+1}^n + T_{i,j-1}^{n+1} \right) \quad (9)$$

Here, provided that you maintain an array $T_{i,j}^n$ for the current iteration and a second array $T_{i,j}^{n+1}$ for the next iteration, you can use this scheme to sequentially compute all the component values $T_{i,j}^{n+1}$. This concept will be illustrated in the lecture.

The Successive Over-Relaxation (SOR) method is a modified form of the G-S method which computes successive grid point values according to:

$$T_{i,j}^{n+1} = T_{i,j}^n + 1.85 \left(\frac{1}{4} \left(T_{i+1,j}^n + T_{i-1,j}^{n+1} + T_{i,j+1}^n + T_{i,j-1}^{n+1} \right) - T_{i,j}^n \right) \quad (10)$$

Using this information, complete the following tasks:

1. Using the information above, explain how you will implement these numerical methods in MATLAB. In your explanation, you should either include a control flow diagram or steps with pseudo code. Make sure to outline how you will:
 - Implement the boundary conditions
 - Initialise the solution domain
 - Structure your code to iterate and update the solution
 - Check for convergence

(10%)

2. Solve the steady-state temperature field for the given boundary conditions using both the G-S and SOR methods. Choose the number of divisions in the domain to be $n_{\text{div}} = 2^n$ for $n = 2 : 9$ in both the x and z directions. (Note: this means the number of points is $n_{\text{pts}} = n_{\text{div}} + 1$.) For each method and each domain discretisation, compute the number of iterations taken to reach the steady-state solution within a tolerance of $\epsilon = 1 \times 10^{-6}$. Present a figure showing the number of iterations vs grid size for the two methods. (15%)
3. Using the SOR method, produce three heat-maps showing the solution resolution improvement at 3 different grid sizes tested above (choose a coarse, medium and fine grid to illustrate the changes). (5%)

Tip: In order to produce a heat-map in MATLAB, use the `surf()` function with edge colour set to zero. The syntax for this is: `surf(X,Y,T,'EdgeColor','none')` where: **X** is a matrix of x positions; **Y** is a matrix of y positions; and **T** is your 2D temperature solution. Use the command `view(0,-90)` to project the surface into the x,y plane, with the colour gradient representing the temperature in that cell. The command `colormap()` allows you to change the colour map, see <https://au.mathworks.com/help/matlab/ref/colormap.html> for more options.

4. Compute the L^1 , L^2 and L^∞ error norms and the order of the error for each respective norm for the SOR method (you don't need to do this for the G-S method). For this error norm analysis, use the same number of points tested in Part 2.2. The error you compute should be based on a comparison with the analytic solution obtained in Section 1. Present your results in a table, as in the lecture slides. State the number of divisions in the domain required to achieve asymptotic convergence. (10%)

The L^p norm for a two-dimensional solution can be computed using the expression:

$$L^p = \left(\Delta x \Delta z \sum_{i=1}^{nx} \sum_{j=1}^{nz} |T_{i,j} - T(x_i, z_j)|^p \right)^{1/p} \quad (11)$$

Note when computing the norm in MATLAB, we need to supply the `norm()` function with a vector input. You can use the following code to compute the p-norms for your solution:

```
% Create a variable for the difference between solutions:
diff      = T_numeric(2:end-1,2:end-1) - T_analytic(2:end-1,2:end-1);
% "Unwrap" the 2D array to a 1D row vector;
diffVec   = reshape(diff',1,numel(diff));
% Compute Lp norms as usual;
L1norm    = norm(diffVec,1)*(dx*dz);
L2norm    = norm(diffVec,2)*sqrt(dx*dz);
LInfnorm  = norm(diffVec,Inf);
```

3 Poisson Equation for Pressure-Density Coupling (20%)

In this section you will extend the applicability of your SOR solver from Part 2 to the Poisson Equation. Ultimately, we want to compute the density field, however the density field depends on both the temperature (which you computed in Part 2) and hydrostatic pressure. The hydrostatic pressure field can be calculated by solving a Poisson equation:

$$\frac{\partial^2 P}{\partial^2 x} + \frac{\partial^2 P}{\partial^2 z} = \frac{1}{\rho} \left(\frac{\partial P}{\partial z} \frac{\partial \rho}{\partial z} + \frac{\partial P}{\partial x} \frac{\partial \rho}{\partial x} \right) = \phi(x, z) \quad (12)$$

where ρ is the density and P is the pressure. Given the pressure field from the above equation, and temperature field from the previous section, the density field can be computed using the Ideal Gas law:

$$\rho = \frac{P}{R_{\text{specific}} T} \quad (13)$$

where $R_{\text{specific}} = 287.058 \text{ J kg}^{-1} \text{ K}^{-1}$ is the specific gas constant, and T is the temperature (specified in Kelvin for this equation).

To solve the Poisson equation, the SOR method may be adapted to include the source term $\phi(x, z)$ as follows:

$$P_{i,j}^{n+1} = P_{i,j}^n + 1.85 \left(\phi_{i,j} + \frac{1}{4} \left(P_{i+1,j}^n + P_{i-1,j}^{n+1} + P_{i,j+1}^n + P_{i,j-1}^{n+1} \right) - P_{i,j}^n \right) \quad (14)$$

You can take the boundary conditions of the pressure field to be:

$$P(x, z = 0) = 101325 \text{ Pa} \quad (15)$$

$$\frac{\partial P(x = 0, z)}{\partial x} = \frac{\partial P(x = L, z)}{\partial x} = 0 \quad (16)$$

$$\frac{\partial P(x, z = D)}{\partial z} = -\rho(x, z = D)g \quad (17)$$

where gravitational acceleration $g = 9.81 \text{ m/s}^2$. These boundary conditions can be discretised as follows:

$$P_{i,j=1}^n = 101325 \quad (18)$$

$$P_{i=1,j}^n = P_{i=2,j}^n \quad (19)$$

$$P_{i=nx,j}^n = P_{i=nx-1,j}^n \quad (20)$$

$$P_{i,j=nz}^n = P_{i,j=nz-1}^n + \rho_{i,j=nz-1}^n g \Delta z \quad (21)$$

Note that the boundary conditions for the density field must be computed using the Ideal Gas Law.

Conceptually, the sequence of steps you will follow is:

- i To start your simulation, assume the pressure distribution is $P = 101325\text{Pa}$ everywhere.
- ii Use the initial pressure field together with the computed temperature field to obtain an initial guess for the density field.
- iii With the initial guess for pressure and density, you can now solve for the new value of pressure after application of the SOR method:
 - Compute the derivatives for the source term.
 - Using your modified SOR solver, compute the new pressure field $P_{i,j}^{n+1}$.
- iv Use the new pressure field to update the density field at the computed points (Equation 13).
- v Update the boundary point values of pressure based on the new pressure field.
- vi Update the density at the boundary points based on the new pressure at the boundary.
- vii Repeat steps (iii) to (vi) until the residuals are below a specified tolerance.

Note that a useful check that your solver is giving sensible results is to consider that for a constant density field, the total pressure difference from top to bottom of the domain should be equal to $\rho g D$.

Using this information, complete the following tasks:

1. State the discretisation you have used to calculate the source term ϕ . (5%)
2. Using the information gained in Section 2, and your own grid convergence study for this problem, determine an appropriate grid size and residual convergence criteria which gives a grid converged solution. Present two line plots of density and pressure across the mid-height of the domain ($z=12.5\text{ m}$) for at least three grid resolutions to support your choice. (10%)
3. Using your solver, compute the density and pressure fields for the domain. Produce heat-maps showing:
 - (a) Initial guess for the density field;
 - (b) Final estimate for the density field (i.e. once pressure field has reached convergence);
 - (c) Final estimate for the pressure field.(5%)

Assignment Submission Information

Your assignment is to be presented as a concise report in PDF format. Note that 10% of the assignment marks are awarded based on the presentation, clarity and functionality of your MATLAB code. A further 10% of the assignment marks are allocated based on overall structure, clarity and presentation of the report. **An electronic copy of the report should be submitted to Turnitin by the due date. Late submissions will incur a penalty of 5% per day late.**

IMPORTANT:

- Marks will be deducted for handwritten answers and screenshots of equations and/or figures. See Tutorial 2 for detailed notes on how to insert figures into your assignment.
- The report should not exceed 10 pages; additional pages will not be marked, so aim to be concise with your findings.
- Any figure/table you include in your report must be numbered and must be referred to and discussed in the main text of your report.
- You are strongly encouraged to read through the "Example Assignment" we have made available on Canvas **here**. This demonstrates the expectations with respect to assignment layout, explanations and figures/tables.

MATLAB Submission

- You will submit your MATLAB Code independently from your assignment submission, **not** as an appendix. Instructions on how to do this will be included in the Assignment 2 Information page on Canvas.
- You are also required to submit a script, which should solve Section 2.2 for $n_{div} = 64$ using the SOR method, and return a heat-map of the temperature solution. This code will be run by the marker and checked as a part of your MATLAB mark.