

Fortran code to solve Heat Equations

Heat Equation for 1D

It has Four steps

1. Taking required inputs
2. Creating required Matrices
3. Using **TDMA** method to get Temperatures
4. Running the program

1. Taking required inputs

- Program takes inputs as follows - `TA` , `TB` (Temperature at ends `A` & `B`) , `n` , `thermalConductivity` , `Area`

2. Creating required Matrices

- We create three Matrices
 - `MatA` - to hold all the coefficients of equations
 - `MatB` - to hold all the constant values of equations
 - `Temperature` - to hold all the Temperatures we get after doing TDMA method
- Then we need to assign appropriate values to `MatA` and `MatB` , which we get from equations.

3. Using **TDMA** method to get Temperatures

- For doing TDMA we need **4** arrays - `a` , `b` , `c` (for the three diagonals) and `d` (for constant values from equations).
- We get the values for `a` , `b` , `c` , `d` from `MatA` and `MatB` .
- Then we do elimination process of **TDMA**.
- And lastly Backsubstitution process of **TDMA**
- Thus Node Temperature values are in `Temperature` array.

4. Running the program

- There is a makefile provided for running this program
 - Use `make project_1` to compile and run the program. It uses input values from `in_1.dat` file(provided). It then prints its output in `out_1.dat` .
 - Use `make clean` to remove old executables, output files.
 - If the program is to be given different input then edit `in_1.dat` file.
-

Heat Equation for 2D

It has Four steps

1. Taking required inputs
2. Creating required Matrices
3. Using **Gauss Seidel** method to get Temperatures
4. Running the program

1. Taking required inputs

- Program takes inputs for - `ny` and `nx` (number of nodes in y and x-direction).
- Then program control goes in `solve_using_gauss_seidal` subroutine.
- Next input to be given is - `Ly` and `Lx` , `topTemperature` and `thermalConductivity` and `Area` .

2. Creating required Matrices

- We need five 2D-arrays - `an` , `as` , `ae` , `aw` and `ap` to store all coefficients from equations. Assign them according to equations.
- Then put values from them to another `Matrix` which holds all coefficients from the equations.
- call `gauss_seidal` subroutine which uses `Matrix` to solve equations and get temperature in `Temperature` array.

3. Using **Gauss Seidel** method to get Temperatures

- Using `Matrix` and its size `n` we solve it using **Gauss Seidel** method and result is stored in `Temperature`.
- The Gauss-Seidel-Iteration is an iteration technique for solving a square system of n linear equations with unknown x .
- Let's suppose we have a matrix with ' n ' number of equations to calculate using Gauss-Seidel-Iteration method we have to choose $x^{(0)}$ the better guess, the quicker the algorithm will perform and $x^{(1)}, x^{(2)}, \dots$
- Let's suppose we have a matrix with ' n ' number of equations to calculate using Gauss-Seidel-Iteration method we have to choose $x^{(0)}$ the better guess, the quicker the algorithm will perform and $x^{(1)}, x^{(2)}, \dots$
- Let's suppose we have a matrix with ' n ' number of equations to calculate using Gauss-Seidel-Iteration method we have to choose $x^{(0)}$ the better guess, the quicker the algorithm will perform and $x^{(1)}, x^{(2)}, \dots, x^{(n)}$. and this process will go until the last answers approx. same the second last answer with less error percentage. Then we got the answer using Gauss-Seidel-Iteration method.

4. Running the program

- There is a makefile provided for running this program
 - Use `make project_2` to compile and run the program. It uses input values from `in_2.dat` file(provided). It then prints its output in `out_2.dat`.
 - Use `make clean` to remove old executables, output files.
 - If the program is to be given different input then edit `in_2.dat` file.
-

Heat Equation for 3D

It has Four steps

1. Taking required inputs
2. Creating required Matrices
3. Using **Gauss Seidel** method to get Temperatures
4. Running the program

1. Taking required inputs

- Program takes inputs for - `nz` , `ny` and `nx` (number of nodes in z,y and x-direction).
- Then program control goes in `solve_using_gauss_seidal` subroutine.
- Next input to be given is - `Lz` , `Ly` and `Lx` , `topTemperature` and `thermalConductivity` and `Area` .

2. Creating required Matrices

- We need seven 2D-arrays - `an` , `as` , `ae` , `aw` , `at` , `ab` and `ap` to store all coefficients from equations. Assign them according to equations.
- Then put values from them to another `Matrix` which holds all coefficients from the equations.
- call `gauss_seidal` subroutine which uses `Matrix` to solve equations and get temperature in `Temperature` array.

3. Using Gauss Seidel method to get Temperatures

- Using `Matrix` and its size `n` we solve it using **Gauss Seidel** method and result is stored in `Temperature` .
- The Gauss-Seidel-Iteration is an iteration technique for solving a square system of n linear equations with unknown x .
- Let's suppose we have a matrix with ' n ' number of equations to calculate using Gauss-Seidel-Iteration method we have to choose $x^{(0)}$ the better guess, the quicker the algorithm will perform and $x^{(1)}, x^{(2)}, \dots$
- Let's suppose we have a matrix with ' n ' number of equations to calculate using Gauss-Seidel-Iteration method we have to choose $x^{(0)}$ the better guess, the quicker the algorithm will perform and $x^{(1)}, x^{(2)}, \dots$
- Let's suppose we have a matrix with ' n ' number of equations to calculate using Gauss-Seidel-Iteration method we have to choose $x^{(0)}$ the better guess, the quicker the algorithm will perform and $x^{(1)}, x^{(2)}, \dots, x^{(n)}$. and this process will go until the last answers approx. same the second last answer with less error percentage. Then we got the answer using Gauss-Seidel-Iteration method.

4. Running the program

- There is a makefile provided for running this program
- Use `make project_2` to compile and run the program. It uses input values from `in_2.dat` file(provided). It then prints its output in `out_2.dat` .
- Use `make clean` to remove old executables, output files.
- If the program is to be given different input then edit `in_2.dat` file.