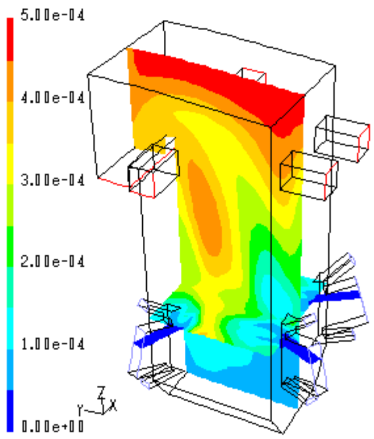
SOLUTION FOR 2-DIMENSIONAL STEADY STATE HEAT CONDUCTION EQUATION THROUGH CONJUGATE GRADIENT METHOD



CFD PROJECT

DONE BY
JAGANNATH SURESH 111117041
JOHN S GEORGE 111117046
SOORAJ MENON VK 111117086

QUESTION

**2_D steady heat conduction is given by $\frac{\partial^2}{\partial x^2}T + \frac{\partial^2}{\partial y^2}T + 100 = 0$. Find the solution of this equation in the form $T(x, y)$ for a plate with boundary conditions

$T(x,0) = 400\,K, T(x,1) = 300K, T(0,y) = 500\,K$ and $\frac{\partial}{\partial x}T = 0$ at $(2, y)$.

Discretize the equation using the finite difference method and solve using conjugate gradient method, Show the temperature contours.**

INTRODUCTION

In numerical analysis, finite-difference methods (FDM) are discretizations used for solving differential equations by approximating them with difference equations that finite differences approximate the derivatives.

FDMs convert linear ordinary differential equations (ODE) or nonlinear partial differential equations (PDE) into a system of equations that can be solved by matrix algebra techniques. In this assignment, the Conjugate Gradient Method is used to solve these equations.

BOUNDARY CONDITIONS & 2D STEADY STATE HEAT TRANSFER EQUATION WITH HEAT GENERATION:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\dot{q}}{k} = 0$$

$T(x,0) = 400\,K, T(x,1) = 300K, T(0,y) = 500\,K$ and $\frac{\partial}{\partial x}T = 0$ at $(2, y)$.

q = heat generation per unit volume.(J/m^3)

k = thermal conductivity.(W/(m · **K**)).

DISCRETIZATION USING FDM

The PDE was discretized using FDM as follows:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} + 100 = 0$$

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = \frac{-100}{n^2}$$ For internal grid points

$$2T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = \frac{-100}{n^2}$$ For adiabatic boundary conditions

CONJUGATE GRADIENT METHOD

In mathematics, the conjugate gradient method is an algorithm for the numerical solution of particular systems of linear equations. The conjugate gradient method is often implemented as an iterative algorithm, applicable to sparse systems that are too large to be handled by a direct implementation. Large sparse systems often arise when numerically solving partial differential equations or optimization problems.
The algorithm is as follows:

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$

if $\mathbf{r}_0$ is sufficiently small, then return $\mathbf{x}_0$ as the result

$\mathbf{p}_0 := \mathbf{r}_0$

$k := 0$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}$$

$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$

$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$

if $\mathbf{r}_{k+1}$ is sufficiently small, then exit loop

$$\beta_k := \frac{\mathbf{r}_{k+1}^\mathsf{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}$$

$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$

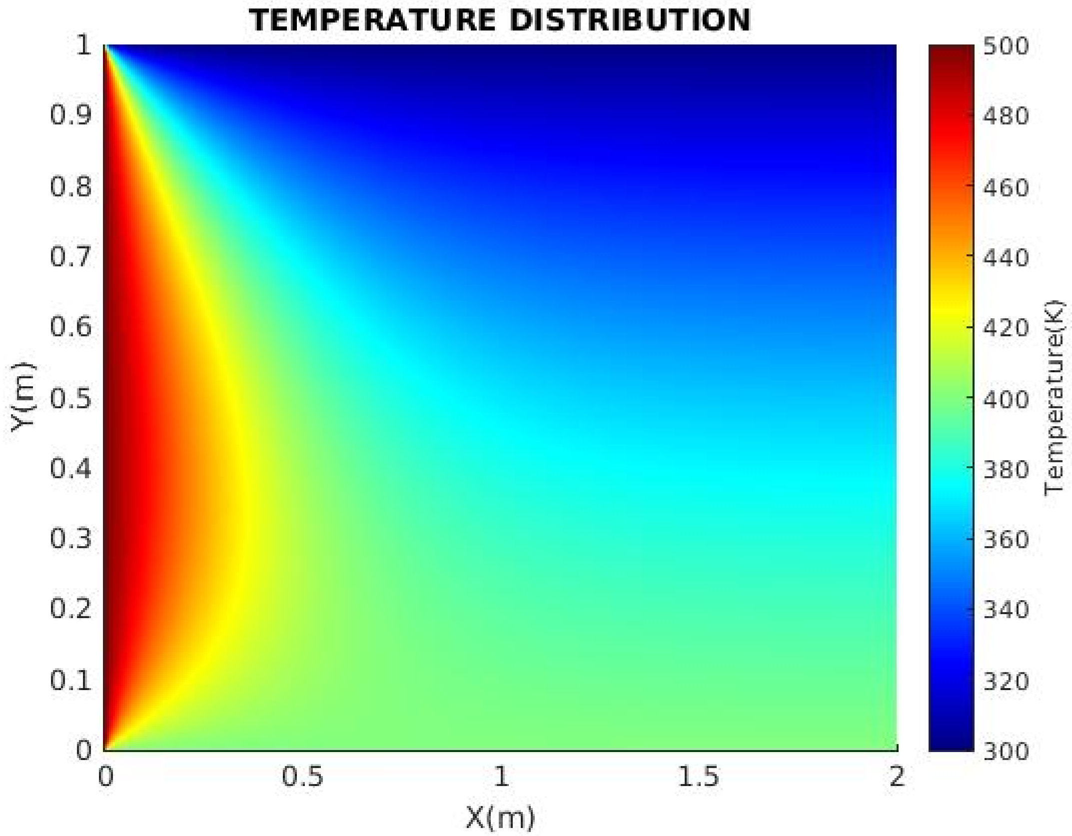
$k := k + 1$

end repeat

return $\mathbf{x}_{k+1}$ as the result

RESULT & CONCLUSIONS

The equation was solved for 13041(81 x 161) nodes in the domain to obtain the below result. However, the number of nodes can be varied in the code for adjusting the mesh refinement. Matlab was exclusively used for coding the problem. In conjugate gradient descent, the iterations were done until the square root of the sum of squares of error at each node was less than 1e-10.



The link to the Matlab code is provided below:

https://drive.google.com/file/d/1eeWjQV6rBZjmtxQ7QxMVQYV3E9Mk_LDM/view?usp=sharing