

Assignment Report

MECH764

Prof. Fadl Moukalled

Due Date: 24/4/2018

Ramy Tanios

201404444

Introduction

In this report, I will be presenting the results of the finite volume discretization method applied to the two dimensional transient heat conduction equation

$$\frac{\partial \phi}{\partial t} - \kappa \nabla^2 \phi = 0,$$

where κ is the thermal diffusivity of the plate and ϕ is temperature. The transient term will be discretized using the **Backward Euler** scheme and the **Crank-Nicolson** scheme.

Problem Statement

In this problem, the thermal diffusivity κ is constant such that $\kappa = 1 \text{ m.s}^{-1}$. The computational domain studied is a square plate with length $L = 1\text{m}$ and height $H = 1\text{m}$. The boundary conditions are such that the north boundary is insulated, i.e $\forall x \in [0, 1], \forall t > 0, \nabla \phi(x, H, t) = 0$ (von Neumann BC) and the other boundaries are held at a constant temperature of 0°C (Dirichlet BC), for $t > 0$. At $t = 0$, the plate has a uniform temperature distribution of 100°C .

Analytical Solution

The analytical solution to this problem with the specified boundary conditions is:

$$\phi(x, y, t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1600}{(2n-1)(2m-1)\pi^2} \sin\left(\frac{(2n-1)\pi x}{L}\right) \sin\left(\frac{(2m-1)\pi y}{2H}\right) e^{-\kappa\pi^2 t \left(\frac{(2m-1)^2}{4H^2} + \frac{(2n-1)^2}{L^2}\right)}$$

Numerical Implementation

The method is implemented using fortran90. The main program *main.f90* calls the following subroutines: *allocateArrays.f90*, *getAnalyticalSolution.f90*, *transfiniteInterpolation.f90*, *geometricQuantities.f90*, *surfaceVectors.f90*, *getFields.f90*, *getDiffusiveCoeff.f90*, *getConvectiveCoeff.f90* and the solver *iterativeSolver.f90*.

Most importantly, the subroutine *getAnalyticalSolution* generate the analytical solution on the 2D grid. Since the solution is of the form of an infinite sum, the series is truncated when the absolute difference between two consecutive terms is less than a specified tolerance, mainly 10^{-8} . Also, the subroutine *getDiffusiveCoeff.f90* adds the contribution of the diffusion term to the coefficients a_E, a_W, a_S, a_N, a_C and b_C . In addition, the subroutine *modifyCoefficientsForTransientSolving.f90* adds the transient term contribution to the source term b_C and the coefficients a_C . Finally, *iterativeSolver.f90* solves for the variable ϕ using the **TDMA** method, applied to lines of constant indexing i and j in order to get a tridiagonal matrix for the system of equations $\mathbf{A}\phi = \mathbf{b}$.

Also, using the command *call system("python plot2D.py")*, the main program will call a python script that uses the *matplotlib* library for plotting the results.

In the module *fields.mod*, the user can change the velocity field function, the diffusivity field

distribution and may add a source term to the equation to be solved. Finally, the boundary conditions of the problem are initialized in the module *boundaryConditions.mod* . In that module, the user can change and set the desired boundary conditions. Note that all the variables in that module are declared as one dimensional arrays of size 4, where the first element corresponds the boundary condition at the north boundary, the second for the east boundary, the third for the south boundary and finally the fourth element for the east boundary of the domain.

Analytical And Finite Volume Solution Results

In the following section, I will present a comparison between the backward Euler and the Crank-Nicolson methods for the discretization of the transient term, as well as 2D plots for the variation of the temperature distribution ϕ across the plate for different times. The time step is set to $\Delta t = 0.001$ sec. The following plot represents a comparison between backward Euler and Crank-Nicolson scheme. The value plotted is the percentage difference between the analytical solution ϕ_{Exact} and the finite volume solution ϕ_{FVM} , i.e

$$P.E = 100 \left| \frac{\phi_{Exact} - \phi_{FVM}}{\phi_{Exact}} \right|.$$

It is clear that the Crank-Nicolson method which is second order provides a more accurate

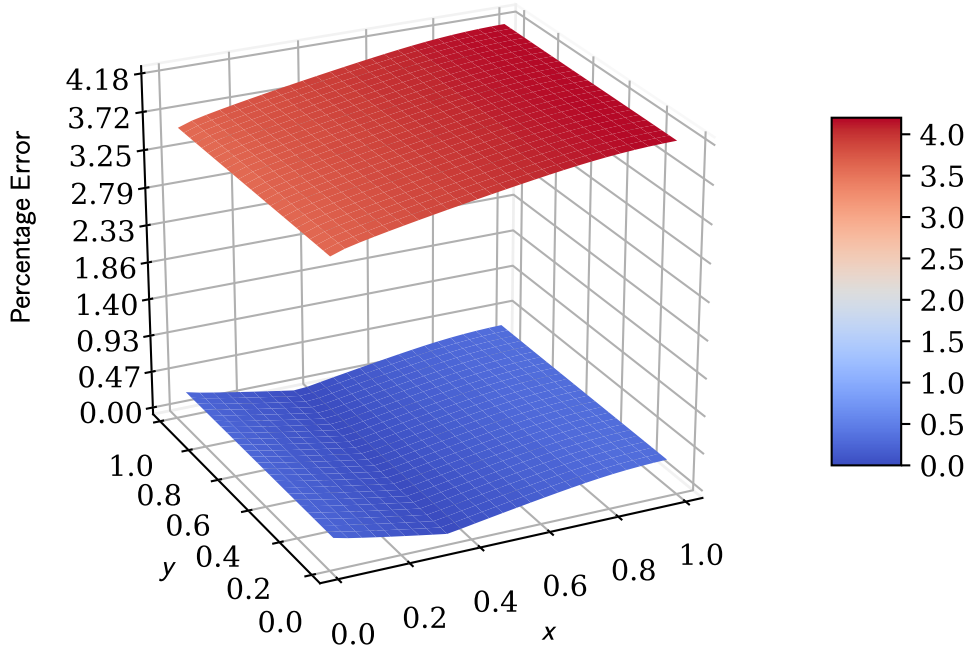


Figure 1: The percentage error generated on the computational domain for the backward Euler method(upper surface) and the Crank-Nicolson method(lower surface) .

solution to the problem in comparison to the backward Euler which is first order. Finally, the

following are plots of the ϕ distribution for different timings. Since the initial temperature of

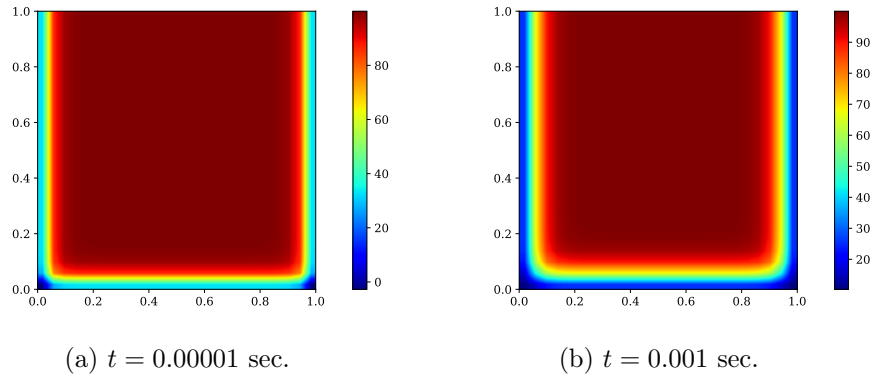


Figure 2: The temperature distribution over the plate for $t = 0.00001$ sec and $t = 0.001$ sec.

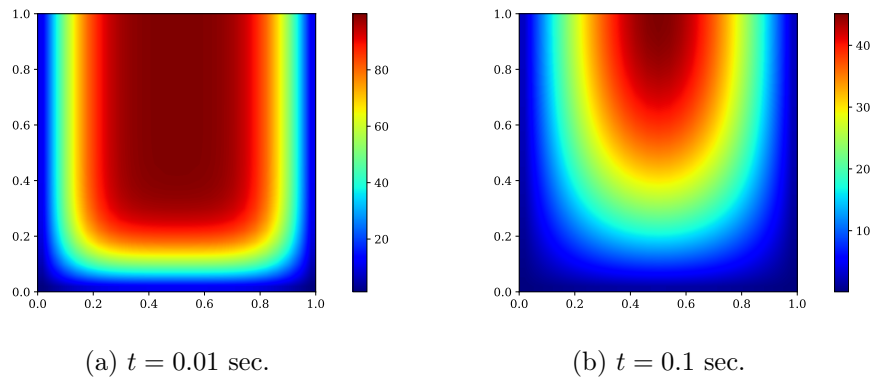


Figure 3: The temperature distribution over the plate for $t = 0.01$ sec and $t = 0.1$ sec.

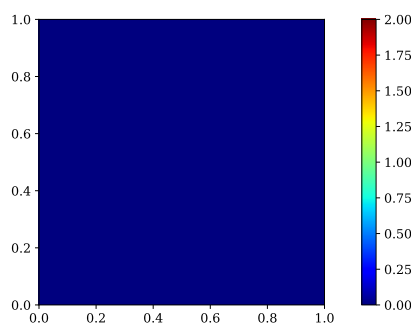


Figure 4: The steady state temperature distribution.

the plate is 100°C and $\nabla\phi(x, H, t) = 0$, the east, west and south boundaries start cooling because of the Dirichlet boundary conditions while part of the north face remains at 100°C . With time, the temperature over the plate starts decreasing in a symmetric way ($x = 0.5$ is the axis of symmetry). As $t \rightarrow \infty$, all the points of the plate will be at 0°C .