**Western University, Department of Mechanical and Materials Engineering**

**MME 9710: Advanced CFD**

*Assignment 1: Introduction to One Dimensional Diffusion Analysis*

**Submitted By:** Alex Kiar, 250731557

**Submitted To:** Dr. Straatman

Date: May 12th, 2017

Table of Contents

[1. Code Implementation 4](#_Toc482888167)

[1.1. Discretization Method 4](#_Toc482888168)

[1.1.1. Control Volume Formulation 4](#_Toc482888169)

[1.1.2. Discretization of the Heat Equation for 1D Conduction 4](#_Toc482888170)

[1.2. main.f 5](#_Toc482888171)

[1.3. difphi.f 5](#_Toc482888172)

[1.4. srct.f 6](#_Toc482888173)

[1.5. coeff.f 6](#_Toc482888174)

[1.6. bndct.f 7](#_Toc482888175)

[1.7. resid.f 7](#_Toc482888176)

[2. Question 1: 1D Conduction 8](#_Toc482888177)

[2.1. Code Implementation 8](#_Toc482888178)

[2.2. Results 9](#_Toc482888179)

[3. Question 2: 1D Convection 11](#_Toc482888180)

[3.1. Code Implementation 11](#_Toc482888181)

[3.2. Results 12](#_Toc482888182)

[4. Question 3: Internal Heat Generation 12](#_Toc482888183)

[5. Question 4: Linearization Techniques 12](#_Toc482888184)

[6. Appendix 1 13](#_Toc482888185)

# Code Implementation

Before solving the assignment questions, the Fortran code provided was updated to solve 1-dimensional heat transfer problems. Initially the code was built to solve 1D conduction only, before adding layers for convection and internal generation. Once those processes were verified, a linearization loop was built to include radiation solutions. The updated code can be found in Appendix 1. The following sections outline the changes made to each file and which physical phenomena those changes reflect.

## Discretization Method

### Control Volume Formulation

<

The discretization method implemented in the Fortran code is based on the Control-Volume (CV) formulation from the method of weighted residuals. In the CV formulation, the domain of interest is divided into a finite number of control volumes. The differential equation being solved is integrated over each control volume, and the result is solved using piecewise interpolation between the control volume interfaces. This method conserves quantities over each control volume and the entire domain regardless of the number of control volumes used.

This method is implemented by setting the end node value (IE) in the in.dat file. This marks the last node before the ending boundary condition. The difference between the end and beginning node value, plus one, indicates the number of control volumes used.

### Discretization of the Heat Equation for 1D Conduction

For a set of three control volumes West, P, and East (Figure 1), the energy balance excluding any sources reduces to:



Figure 1: Control Volume Set.

Where,

Using piecewise linear approximation of the flux through each control volume, and piecewise constant approximation for the temperature of a given control volume () as the temperature at the center of a control volume is assumed to be representative of the whole volume, the discretized energy balance for a given control volume becomes:

Where each coefficient is discussed in the following sections of code they are represented in.

## main.f

The main routine implements each element of the discretization process by calling the subroutines outlined below. This routine implements a linearization loop that continues iterating on the temperature field until the residuals converge. The main routine begins by initializing the temperature field, calculating an initial set of coefficients, and calculating the temperature field again with those coefficients. Following the first iteration, the routine updates the coefficients with the previous iteration’s temperature field, and calculates the average residual. If the average residual meets the convergence criteria, the main routine ends.

## difphi.f

This subroutine was built to calculate the diffusion coefficient (DE) on the East face of each control volume. A loop was built to calculate the East face DE for each node from IB-1 (the first boundary condition) to IE (the end node). The loop includes IB-1 because diffusion exists on the East face of the initial boundary condition. Diffusion does not exist on the end boundary condition, which is why the loop does not include IE+1. DE was calculated using the following equation:

Where is the thermal conductivity, is the cross-sectional area of the east face of the control volume, is the distance to the center of the East node, and is the distance to the center of the node of interest. Equation (1) results from linear piecewise interpolation of node point temperatures. This method is used to ensure that the discretized equation obeys the conservation law. For conservation to be true,

To satisfy this condition, the flux and temperature distribution at each integration point (e/w) must be the same for control volumes E and P. If a method of order higher than linear is used, the resulting distributions would not be equal, and conservation would not be maintained. If constant flux approximations were used, the resulting integration at each point would approach infinity, as the distribution would not be continuous, resulting in the same outcome.

As a result of the conservation law, the DE of the east face of one node must be equal to the DW of the following node. This allows only the DE of the east face to be calculated for each control volume.

After the loop finishes, the subroutine returns an array of DE values for each control volume.

## srct.f

This subroutine was used to determine the net source terms of each control volume. Source terms are created because of the linearization of each source of heat within a control volume. For each control volume, the total source is given by,

Assuming the control volume has the discrete equation given in (1) and ignoring radiation, the source term reduces to,

Where m is the current iteration of temperature, and and are given by,

Where is the fixed source coefficient, and is the linearized source coefficient. When radiation is not ignored, the source term must be linearized to ensure that piecewise linear flux problems are solved between each control volume. Various linearization techniques are available and give different results, as they break the non-linear problems down in different ways.

The subroutine implements a loop to calculate each source term and over each control volume. Three linearization techniques were included and tested in question 4. An internal generation term is also included in the code, adding to the source term on each control volume. The subroutine returns the two source terms to be used in the active coefficient calculation.

## coeff.f

This subroutine was used to calculate the active coefficients for each control volume. The active coefficients are the coefficients belonging to each temperature term given in equation (1). Each coefficient is given by,

A loop was implemented to calculate the coefficients for each control volume.

## bndct.f

This subroutine was used to assign boundary conditions to the active coefficient arrays. Three types of boundary conditions were derived for common scenarios. Each boundary condition was listed in this subroutine, and updated based on the type of problem being solved.

## resid.f

,,

This subroutine was used to calculate the residual of each node after a temperature field was solved. The residual describes how closely a linearized problem was solved, and measures the convergence of a solution over multiple iterations. The residual formula is given by,

The residual measures the difference between the updated coefficient calculation, and the previous iterations temperature. The residuals were implemented by comparing the average residual over all the control volumes to a convergence criteria. When the residual reached the criteria, the solution was accepted.

# Question 1: 1D Conduction

This problem analyzed a square fin experiencing pure conduction with prescribed temperature on each end of the fin. The temperature distribution across the fin was solved for. The properties of the problem can be found in Table 1, and the geometry of the problem can be found in Figure 2.

Table 1: Question 1 properties





Figure 2: Square fin geometry.

## Code Implementation

This problem was implemented by setting the geometry in makgrd.f, setting the boundary conditions in bndct.f, and varying the number of control volumes used in the discretization.

The half height, and half depth Y and Z, were set in makgrd.f to initialize the geometry of the problem. Dirichlet boundary conditions were present in the problem due to specified temperatures at each end of the fin. This boundary condition was set in bndct.f, and can be seen in Table 2. The boundary conditions were initialized at nodes IB – 1 and IE + 1 because the boundary condition control volume has zero volume, and the middle of the control volume is sent to the edges of the IB and IE nodes. The number of control volumes were varied from 1 to 64, doubling in each iteration, to determine the effect of meshing on the temperature distribution.



Table 2: Question 1 boundary conditions

## Results

For each number of control volumes, the average residual and heat flux between the first two nodes were calculated. The heat flux was used as the criterion to determine when the grid had converged. Table 3 summarizes the results of each control volume. The results of each control volume were plotted on a composite plot to visualize the grid convergence. Figure 2 plots the temperature distribution of each control volume iteration.

Table 3: Summary of Question 1 results



Both residual and grid convergence were tested to verify the solution of each number of control volumes. For each number of control volumes, the residuals converged after the first iteration. Since the problem was completely linear, none of the active coefficients were a function of temperature. This caused the active coefficients to remain constant after the first solution, which caused a constant solution over each iteration of the linearization loop. The residuals of 32 and 64 control volumes did not reach the convergence criteria. However, the residuals were still small, and converged after the first iteration as expected, resulting in a reasonable solution.

To measure grid convergence, the heat flux between the IB – 1 and IB nodes was calculated. The heat flux remained relatively constant regardless of the number of control volumes used, as the change in heat flux between each number of control volumes was always below two percent. This indicates that the temperature distribution was completely grid independent, and is reliable over all sizes of control volumes.

One control volume was tested to determine the extreme behavior of the solver. Since the heat flux remained constant, one control volume was a reasonable number to use for this problem. This is a result of the geometry and expected temperature distribution of the problem. Since the problem was completely linear, both in temperature distribution (no source terms) and in geometry (constant cross section), the temperature of the fin is directly proportional to its x distance from the first node. This is responsible for making all control volume sizes reasonable approximations of the temperature distribution.

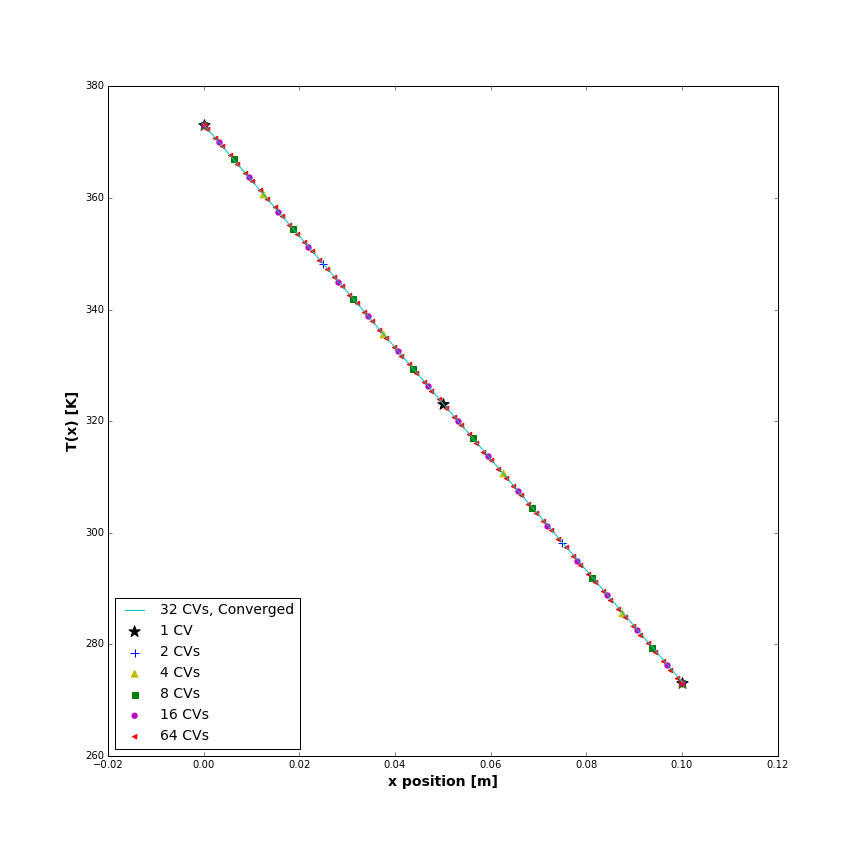


Figure 2: Temperature distribution of square fin with pure conduction

# Question 2: 1D Convection

This problem analyzed a square fin experiencing both conduction and convection with prescribed temperatures on each end of the fin. The temperature distribution across the fin was solved for and compared to the analytic solution. The properties of the problem can be found in Table 4, and the geometry of this problem was the same as problem 1, and can be found in Figure 3.

Table 4: Problem 2 properties



## Code Implementation

<

This problem was verified using the same process as problem 1, with the addition of source term calculations for each control volume to account for convection.

The same geometry from problem 1 was kept in makgrd.f. The Dirichlet boundary condition were present in the problem due to specified temperatures at each end of the fin. This boundary condition was set in bndct.f, and can be seen in Table 5. The boundary conditions were initialized at nodes IB – 1 and IE + 1. The number of control volumes were varied from 1 to 64, doubling in each iteration, to determine the effect of meshing on the temperature distribution.

The source term calculation was built into the code by calculating and in each control volume as discussed in section 1.4. This accounted for the convection heat removal over the surface of each control volume.

Table 5: Problem 2 boundary conditions



## Results

For each number of control volumes, the average residual, heat flux between the first two nodes, and the temperature gradient between the first two nodes were calculated. The heat flux and temperature gradient were each used as the criterion to determine when grid convergence had been reached. Table \_\_ summarizes the results of each control volume. The results of each control volume were plotted on a composite plot to visualize the grid convergence. Figure \_\_ plots the temperature distribution of each control volume iteration.

When grid convergence was achieved, the converged control volume size was compared to the analytic solution.

# Question 3: Internal Heat Generation

# Question 4: Linearization Techniques

# Appendix 1