

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/235164287>

Finite Volume Algorithms for Heat Conduction

Article · May 2010

CITATIONS

4

READS

6,325

1 author:



[Douglas Vinson Nance](#)

Air Force Research Laboratory

27 PUBLICATIONS 74 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Computational Fluid Dynamics for Unsteady Flow [View project](#)



Computational Aeroacoustics Research [View project](#)

AFRL-RW-EG-TR-2010-049

FINITE VOLUME ALGORITHMS FOR HEAT CONDUCTION

Douglas V. Nance

Air Force Research Laboratory
Munitions Directorate
AFRL/RWPC
101 W. Eglin Blvd.
Eglin AFB FL 32542-6810



MAY 2010

TECHNICAL REPORT FOR PERIOD DECEMBER 2009 - MAY 2010

DISTRIBUTION A: Approved for public release; distribution unlimited. 96th ABW/PA Approval and Clearance # 96ABW-2010-0204, dated 5 April 2010.

AIR FORCE RESEARCH LABORATORY, MUNITIONS DIRECTORATE

■ Air Force Materiel Command ■ United States Air Force ■ Eglin Air Force Base

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

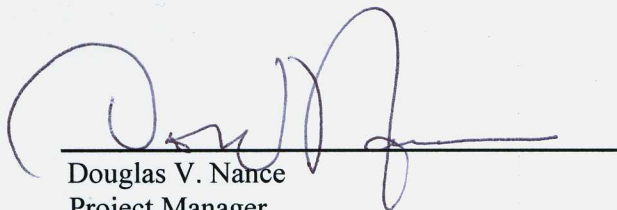
Qualified requestors may obtain copies of this report from the Defense Technical Information Center (DTIC) <<http://www.dtic.mil/dtic/index.html>>.

AFRL-RW-EG-TR-2010-049 HAS BEEN REVIEWED AND IS APPROVED FOR
PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:



Danny R. Brubaker, DR-IV
Technical Advisor
Lethality and Vulnerability Branch



Douglas V. Nance
Project Manager
Computational Mechanics Branch

This technical paper is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 11-05-2010		2. REPORT TYPE TECHNICAL		3. DATES COVERED (From - To) Dec 2009 - May 2010	
4. TITLE AND SUBTITLE Finite Volume Algorithms for Heat Conduction				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 62602F	
6. AUTHOR(S) Douglas V. Nance				5d. PROJECT NUMBER 2502	
				5e. TASK NUMBER 67	
				5f. WORK UNIT NUMBER 63	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Munitions Directorate AFRL/RWPC 101 W. Eglin Blvd. Eglin AFB, FL 32542-6810				8. PERFORMING ORGANIZATION REPORT NUMBER AFRL-RW-EG-TR-2010-049	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Munitions Directorate AFRL/RWPC 101 W. Eglin Blvd. Eglin AFB, FL 32542-6810				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL-RW-EG	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RW-EG-TR-2010-049	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION A: Approved for public release; distribution unlimited. 96 th ABW/PA Approval and Clearance # 96ABW-2010-0204, dated 5 April 2010.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Many modern computational fluid dynamics computer programs are developed by using the finite volume discretization method. It has an excellent numerical capability for capturing changes in conserved quantities such as mass, momentum and energy. In many cases, thermal energy is transferred from fluids to some adjacent solid mass. Heat accumulation in this solid matter is an important engineering issue. To capture this energy transfer, it is important to have heat conduction algorithms that function well with fluid dynamics codes. The present work tackles this problem by presenting an algorithm for solving the heat equation in finite volume form. Although this derivation is cast in two dimensions, it may be readily generalized to three dimensions. Example problems are solved involving heat conduction within a section of an annular ring. Along the boundaries we enforce both Dirichlet and Neumann boundary conditions. For code validation, our numerical solutions, based upon the Douglas-Rachford ADI time integration scheme, are compared with exact mathematical solutions.					
15. SUBJECT TERMS heat conduction, finite volume, generalized coordinate, ADI, Fourier series					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL	44	

TABLE OF CONTENTS

Section	Page
1.0 INTRODUCTION	4
1.1 Equation for Heat Conduction	5
1.2 A Map for the Remainder of this Report	6
2.0 METHODS, ASSUMPTIONS AND PROCEDURES	8
2.1 Finite Volume Discretization	8
2.2 Transformation Mathematics and Local Coordinates	10
2.3 Spatial Discretization of Heat Flux	13
2.4 Explicit Time Integration	16
2.5 Implicit Time Integration	18
2.6 Boundary Conditions	21
3.0 RESULTS AND DISCUSSION	24
3.1 Rectangular Asymmetric Heating Problem	24
3.2 Radial Asymmetric Heating of an Annular Section	25
3.3 Radial and Azimuthal Asymmetric Heating of an Annular Section	27
4.0 CONCLUSIONS	30
REFERENCES	31
APPENDIX A	32
APPENDIX B	33
APPENDIX C	35
APPENDIX D	39

LIST OF FIGURES

Figure		Page
1	Configuration of a Vertex Centered Quadrilateral 2D Finite Volume Cell	9
2	A Body Conforming Coordinate System for a Section of an Annular Ring	11
3	Grid Array for Vertex Centered Finite Volume Cell (i,j)	13
4	Section of an Annular Ring with Both Dirichlet and Neumann Boundary Conditions	21
5	Problem 1 Solutions Calculated by Using the Douglas-Rachford ADI Scheme for Curvilinear Coordinates. Numerical Solutions are Represented by Solid Lines. Exact Solutions are Represented by Symbols.	25
6	Problem 1 Solutions Calculated by Using the Douglas-Rachford ADI Scheme for Curvilinear Coordinates. Solutions are Compared Along Lines of Constant x at Time 1500.	25
7	Problem 2 Time Sequence of Radial Solution Plots. The Explicit (EXPL) and ADI (IMPL) Numerical Solutions are Compared Against the Exact (EXCT) Solution at Times 0.1, 0.2 and 1.0.	26
8	Problem 2 Analysis of Azimuthal Symmetry Occurring in Numerical Solutions. Plots are Shown for the Explicit (EXPL) Solution at Time 0.1 and for the ADI (IMPL) Solution at time 0.2. In Each Case, Numerical Solutions are Graphed Along the Annular Rays Located at 0° , 22.5° , 45° , 67.5° and 90° .	27
9	Problem 3 Comparisons of the Numerical ADI and Exact Solutions. Solutions are Plotted Along Radial Lines Located at 0° , 22.5° , 45° , 67.5° and 90° .	28
10	Problem 3 Color Contour Plots of the Temperature Field on an Annular Section. Plots of the ADI Numerical Solution are Shown at Times 0.1, 0.2, 0.3 and 0.5. All Solution Parameters are Dimensionless.	29
11	Boundary Condition Determinant Values for the Asymmetric Heating Problem Set on an Annular Ring Section Domain with Azimuthal Symmetry	37

SUMMARY

Many modern computational fluid dynamics computer programs are developed by using the finite volume discretization method. It has an excellent numerical capability for capturing changes in conserved quantities such as mass, momentum and energy. In many cases, thermal energy is transferred from fluids to some adjacent solid mass. Heat accumulation in this solid matter is an important engineering issue. To capture this energy transfer, it is important to have heat conduction algorithms that function well with fluid dynamics codes. The present work tackles this problem by presenting an algorithm for solving the heat equation in finite volume form. Although this derivation is cast in two dimensions, it may be readily generalized to three dimensions. Example problems are solved involving heat conduction within a section of an annular ring. Along the boundaries we enforce both Dirichlet and Neumann boundary conditions. For code validation, our numerical solutions, based upon the Douglas-Rachford ADI time integration scheme, are compared with exact mathematical solutions.

1.0 INTRODUCTION

The transfer of heat has been of great interest within the engineering and scientific communities for a very long time. Engineering systems are often susceptible to damage from extreme temperatures, either hot or cold, so in the course of operation, it is important to understand the temperature distribution through the material comprising the system. Consider the simple example of a piping section that transfers fluid through a conventionally fueled power plant. If it leads from the boiler to the turbo-electric generator, the piping becomes very hot. Since this piping is connected to nearby thermally conductive structures, we realize that the entire system has a complicated temperature distribution. Temperature gradients throughout the system create thermal stress fields in the piping material. Like its mechanical counterpart, thermal stress tends to distort the size and shape of a mechanical component.[1] Moreover, excessive temperature gradients can cause serious component damage. If the boiler tubing is suddenly subjected to a torrent of cold water upon a sudden loss of the heat source, the tubing may crack or shatter. The piping may also bend or change length breaking piping joints. These deleterious results are ultimately tied to the temperature distribution in the material. Since society relies upon the generation of energy and heat (either as a direct or indirect product), the importance of understanding heat transfer effects is made clear.

To provide clarity in the absence of rigor, we may differentiate between two of the major types of heat transfer. Convective heat transfer involves the large scale motion of material carrying thermal energy.[2] The piping system mentioned above carries high temperature fluid from a hot source to a cooler heat sink. The motion of the fluid in the pipe characterizes this transfer as being convective. Conductive heat transfer is not the same; instead, it relies upon tiny molecular motions.[3] If you hold an ice cube in your hands, heat conduction becomes apparent. The ice is, for all intent, solid and non-moving, so no large scale material motion occurs. Yet, your hand becomes cold where it is in contact with the ice, and the ice begins to melt into liquid. At the interface between skin and ice, molecules are in contact, and the warmer skin molecules transfer kinetic energy via collisions into the colder, more lethargic ice molecules. This process is governed by molecular diffusion, not by large scale material motion. The work that follows serves to model heat conduction in a careful way with a flexible set of numerical algorithms suitable for use in solving problems involving non-Cartesian geometries.

Standard numerical techniques for modeling heat conduction are really developed for Cartesian coordinates, but the vast majority of engineering problems have non-Cartesian geometries. A circular walled pipe is a good example. We can only apply Cartesian algorithms to this problem if we apply special computer coding techniques such as (i) using a system of interpolations at the boundary to remap the cylindrical problem geometry onto a background Cartesian grid, or (ii) representing the body as complicated interface on the Cartesian grid. This interface is effectively tracked on the Cartesian grid, and the physics of the problem can then be stored in reference to Cartesian grid. Approach (i) is quite old, and it is rife with error due to the need for an interpolation remapping scheme.[4] This scheme, in general, has no relevance to the physics of the problem. Near oddly shaped boundaries, the use of interpolation also requires a great deal of bookkeeping to track the indicial limits of the boundary on the grid. Approach (ii) is quite valuable, yet it is still a matter of contemporary research in numerical algorithms.[5] Modern interface tracking methods are very complicated, but they can apply to a wide class of

problems. In fact, you can treat multi-physics problems by using this approach. However, it is simply not necessary to resort to this much numerical machinery in order to develop a robust and accurate computational solver for heat conduction. Instead, the finite volume discretization technique may be applied in conjunction with a generalized coordinate transformation in order to map the governing equation onto a system of body-fitted coordinates. The transformed system is readily solvable by conventional discretization techniques. In fact, it is desirable that our time and space discretization equations look as much like their classical Cartesian counterparts as is possible. This widely applicable approach lies at the foundation of modern numerical disciplines like computational fluid dynamics (CFD) and computational electromagnetics (CEM). Like these disciplines, computational heat transfer relies upon the continuum mechanics interpretation of heat flow through governing partial differential equations.

1.1 Equation for Heat Conduction

Let u be the temperature at a point in space within a mass of material. Generally, temperature is measured on an absolute scale, but relative scales are also acceptable here. The equation representing the time dependent conduction of heat (without the presence of internal heat sources), as represented by the temperature, is written as follows.[6]

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0 \quad (1)$$

In (1), $u = u(x, y, z, t)$, where x , y and z are the Cartesian coordinates of the point; the time is denoted as t . The parameter α is the coefficient of thermal diffusivity defined as

$$\alpha = \frac{k}{\rho C_p} \quad (2)$$

where k is the coefficient of thermal conductivity for the material (taken as constant in this case), while ρ and C_p are the density and specific heat capacity of the material. For problems involving a spatially dependent thermal conductivity, (1) has a slightly different form, but our numerical algorithms function just as well for that case. Equation (1) has a number of names such as the heat equation, the diffusion equation and Fourier's equation. More importantly, it is classified as a linear, parabolic partial differential equation. This classification is used to detail a set of mathematical techniques that may be used to obtain "exact" solutions for (1). Later in this report, we apply some of these techniques to build validation solutions for our numerical algorithms. Equation (1) also contains the Laplace operator defined as

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

in three Cartesian dimensions. For problems cast in fewer dimensions, we simply delete from (3) the derivative for the missing coordinate(s). The Laplace operator can be expressed in generalized (non-Cartesian) coordinates, but its mathematical form is greatly complicated by this process. The heat equation (or diffusion operator) tends to add a great deal of smoothness to an

initial temperature distribution (or to the distribution of some other scalar quantity). Although (3) is relatively easy to discretize by using finite differences, its form in generalized coordinates is not. Later, we discuss the application of the finite volume discretization method, a technique we can use to overcome this difficulty.

The heat equation is normally solved as an initial-boundary value problem (IBVP) meaning that the equation must be accompanied by (i) a domain, (ii) a set of conditions specifying either the solution or its derivatives along the boundary and (iii) initial conditions specifying the solution at every point in the domain at the starting time for the problem. In one dimension, an example of a classical IBVP for the heat equation is shown below.

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} & u &= u(x, t) & 0 < x < 1, \quad t > 0 \\ u(0, t) &= 0 & u(1, t) &= 0 \\ u(x, 0) &= f(x) \end{aligned} \tag{4}$$

In (4), the thermal diffusivity is set equal to one. The time and space domains are clearly described in (4), and the boundary conditions are provided on the second line. Note that the boundary consists of exactly two space points for the problem. The initial conditions are delineated on the last line. For a piecewise continuous function $f(x)$, this problem is easily solved by separating variables and using Fourier series. The exact solution is

$$u(x, t) = \sum_{n=1}^{\infty} A_n \exp(-n^2 \pi^2 t) \sin(n \pi x) \tag{5}$$

where

$$A_n = 2 \int_0^1 f(x) \sin(n \pi x) dx \tag{6}$$

Exact solutions may be obtained for the heat equation in many cases, but only for problems cast in simple orthogonal coordinate systems (e.g., Cartesian, Cylindrical, Spherical). For other coordinate systems, we must resort to numerical solution techniques. This topic is the subject of the remainder of this report.

1.2 A Map for the Remainder of this Report

In Section 2, a discussion of the finite volume method is presented along with complete derivations required for discretizing the heat equation on non-Cartesian domains in two dimensions. Although this exposition is limited to two dimensions, the method is easily extended to three dimensions and for problems involving non-constant thermal conductivities. To assist in discretizing curvilinear bodies, a local coordinate transformation is introduced and used in transforming derivatives. With the spatial discretization scheme intact, we express finite volume interface quantities in a form suitable for use in alternating direction implicit (ADI) and explicit time-integration schemes. In fact, a finite volume interpretation of the Douglas-Rachford scheme is presented. Section 3 contains a series of test cases for the numerical algorithm. The basic numerical concept is demonstrated first for rectangular coordinates in solving the asymmetrical

heating problem. Although this test case is simple, it possesses real utility in testing the numerical scheme. Secondly, we solve the heat equation on a 90° sector of an annular ring undergoing asymmetrical heating in the radial direction. This problem is characterized by azimuthal (or angular) symmetry, yet it provides a thorough test of the time-marching scheme. For the final test case, we again employ the annular ring, but without azimuthal symmetry. Asymmetric heating is applied in both the radial and angular directions, particularly along the outer boundary of the ring sector. For each test case, the numerical solution is compared with an exact mathematical solution determined by Fourier series analysis. Good agreement with the exact solution is demonstrated for each test case, validating our numerical methods. Section 4 contains concluding remarks that pertain largely to the efficacy of this numerical approach to heat conduction. Cautionary remarks are made on the treatment of non-orthogonal curvilinear coordinate systems. Large numerical discretization errors can occur in these coordinate systems if the discretization is not conducted with exceptional care. Time dependent boundary conditions are also discussed since the temperature (or heat flux) is forced to change in time along the boundary of many engineering devices. Near the end of this report, appendices are included to describe the derivation of the exact solutions. These solutions are thoroughly discussed because of their importance in the validation process.

2.0 METHODS, ASSUMPTIONS AND PROCEDURES

The exposition that follows contains the mathematical machinery required to transform the heat equation from differential form into a set of algebraic equations that may easily be coded in a computer language. Diligence is applied in maintaining the derivation as rigorous as possible without sacrificing practicality. At the end of the day, a report that lacks implementation details is of little use to the practicing engineer or scientist. Naturally, we avoid mathematical proofs involving concepts such as convergence and stability, and this approach is reasonable since we use legacy numerical schemes (admittedly with some required and well documented changes) whenever possible. Details associated with the mechanics of our finite volume discretization we endeavor to explain in detail.

2.1 Finite Volume Discretization

Many scientists are familiar with the finite difference method for discretizing differential equations. In fact, the Newton divided difference is the numerical analog for a derivative.[7] Interestingly enough, the finite volume method relies upon integration via the Fundamental Theorem of Calculus for the simple one-dimensional example shown below. Suppose we wish to evaluate dy/dx at some point x_i . Let us integrate this derivative on the interval $[x_{i-1/2}, x_{i+1/2}]$.

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{dy}{dx} dx = y(x) \Big|_{x_{i-1/2}}^{x_{i+1/2}} = y(x_{i+1/2}) - y(x_{i-1/2}) \quad (7)$$

Also, we can envision the concept of an averaged value for dy/dx defined at x_i . In this sense, we have that

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{dy}{dx} dx = \overline{\frac{dy}{dx}} \Big|_{x_i} \cdot \Delta x \quad (8)$$

Hence,

$$\overline{\frac{dy}{dx}} \Big|_{x_i} = \frac{y(x_{i+1/2}) - y(x_{i-1/2})}{\Delta x} \quad (9)$$

where $\Delta x = x_{i+1/2} - x_{i-1/2}$. The expression (8) for the average derivative matches the Newton divided difference formula, so for uniform one-dimensional meshes, the finite volume and finite difference discretization procedures render the same results. It is instructive also to note that the finite volume method introduces the concept of “averaged” values. More importantly, the finite volume procedure has even greater utility in higher dimensions.

Consider an example cast in a plane as is shown in Figure 1. For simplicity, we have chosen not to show a complete body (or domain) geometry. Instead, we have outlined a single finite volume “cell” in dashed lines. Note that the sides of the cell are not aligned with any set of Cartesian coordinates. The cell is identified by indicial coordinates (i, j) . In the discussions that follow, the vertex centered finite volume method is applied. That is to say, the center of cell coincides with a grid point possessing the same indices. The cell centered finite volume is equally valid and invokes the so-called “northeast” cell numbering convention. According to this

convention, the grid point assigns its indices to the cell center to its northeast. The assigning grid point is circled in Figure 1. These indicial coordinates are associated with a system of local

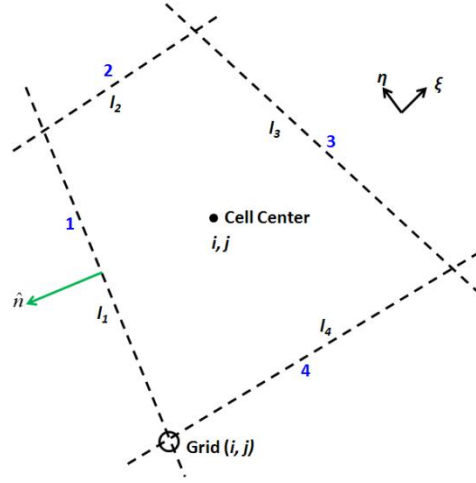


Figure 1: Configuration of a Vertex Centered Quadrilateral 2D Finite Volume Cell

coordinates (ξ, η) shown in the figure. The local coordinates can be identified with directional vectors $\hat{\xi}$ and $\hat{\eta}$. The directions of these two vectors change with each point in the grid unlike a Cartesian system.

$$\frac{\partial u}{\partial t} = \nabla^2 u = \nabla \cdot \nabla u \quad (10)$$

where

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}; \quad \nabla = \hat{i} \frac{\partial}{\partial x} + \hat{j} \frac{\partial}{\partial y} \quad (11)$$

∇ being the Gradient, a commonly used vector differential operator.

To begin the finite volume discretization of the 2D heat equation above, we invoke the Divergence Theorem, a higher dimensional version of the Fundamental Theorem of Calculus.[8] Integrate (10) over the interior volume of the cell. Note that in 2D, volume is interpreted as area, and it follows that the 2D analog of surface area (for the cell sides) is length. Accordingly, the quadrilateral cell shown in Figure 1 has four sides, labeled 1 through 4. Each cell side m (or interface) has length l_m , $m=1, \dots, 4$. We have that

$$\int_V \frac{\partial u}{\partial t} dv = \int_V \nabla \cdot \nabla u dv = \oint_{\partial V} \nabla u \cdot \hat{n} ds \quad (12)$$

The right most term in (12) is has been produced by application of the Divergence Theorem. The overall volume (area) integral has been transformed into an integral over the closed surface (side lengths) of the cell. The symbol \hat{n} represents the outer pointing unit normal vector defined at each point on the cell interface. Since the cell has four straight sides, \hat{n} takes on four distinct vector values. The left hand side of (12) can also be rewritten. For our purposes, the grid does not move, so the order of differentiation and integration can be interchanged, i.e.,

$$\int_V \frac{\partial u}{\partial t} dv = \frac{\partial}{\partial t} \int_V u dv \quad (13)$$

Recall that $u = u(x, y, t)$, so the process of integrating u over cell volume defines a *de facto* average \bar{u} in that

$$\int_V u \, dv = \bar{u} V \quad (14)$$

By substituting (13) and (14) into (12), we have that

$$\frac{d\bar{u}_{i,j}}{dt} = \frac{1}{V_{i,j}} \oint_{\partial ds} \nabla u_{i,j} \cdot \hat{n} \, ds \quad (15)$$

Note that because of the process of spatial integration, $\bar{u}_{i,j}$ does not depend on any particular point inside of the cell; in the cell, it is solely a function of time. Hence, a total time derivative has replaced the partial derivative. The symbol $V_{i,j}$ represents the total volume (or area, in this case) for the cell. Since the cell boundary is represented by a sequence of line segments, we can simplify the right hand side of (15). In fact, the integral over the closed boundary can be replaced by a summation of integrals along directed line segments, i.e.,

$$\frac{d\bar{u}_{i,j}}{dt} = \frac{1}{V_{i,j}} \sum_{m=1}^4 \int_m (\nabla u_{i,j} \cdot \hat{n})_m \, ds \quad (16)$$

At this point, the finite volume method applies an approximation. If the function u was known as a continuum in an open neighbor of the cell sides, the integration in (16) could be performed in an exact manner. Unfortunately, u is usually known only at a discrete number of points. Moreover, in order for (16) to form a proper numerical method, we must *reconstruct* ∇u from the locus of \bar{u} . This approximation is made at the center of each side of the cell. That is to say, ∇u is estimated at the center point of each cell side then multiplied by the side length. Likewise, the normal vector \hat{n} is calculated at the same point. Equation (16) then becomes

$$\frac{d\bar{u}_{i,j}}{dt} = \frac{1}{V_{i,j}} \sum_{m=1}^4 \nabla \bar{u}_{i,j}^m \cdot \hat{n}_m l_m \quad (17)$$

In (17), geometrical quantities such as cell volume, side normal vectors and lengths are easily calculated based upon the grid point locations while quantities based upon \bar{u} require a more considered approach. When a discretized form for $\nabla \bar{u}$ is used (a subject addressed later), one derivative remains in this expression. For that reason, (17) is referred to as a semi-discrete form. This form is very useful since it is amenable to a number of different temporal discretization schemes. Still, we must address differentiation on the curvilinear grid before we can completely formulate a discretized version of $\nabla u_{i,j}^m$.

2.2 Transformation Mathematics and Local Coordinates

The use of Cartesian discretization is simply impractical for most real world engineering problems. Even when applied along with a modern interface tracking method for oddly shaped bodies, interesting questions arise regarding the accuracy of derivatives computed near the boundary. For instance, if a thermal insulator is installed on the curved boundary of a material, physics dictates that we enforce boundary conditions on heat flux in the direction of the normal vector at every point on the boundary. A Cartesian approximation of this process enforces this

boundary condition by resolving the heat flux vector along two Cartesian coordinates. This approach works, but the absolute reliance upon fictitious coordinate systems and polygonal cut cells tends to leave the careful scientist with a feeling of unease. Instead, let us consider another approach to the problem.

Through the use of calculus, we can construct a transformation that maps the body and its boundary onto a Cartesian-like computational space. In this space, it is easy to compute differences along smooth coordinate lines that conform to the shape of the boundary. If the curvilinear coordinates are nearly orthogonal, then derivatives calculated with the use of these coordinates take on simple, easily discretizable forms. To support the 2D example shown in Figure 2, consider a transformation T from the (ξ, η) curvilinear coordinate system to the (x, y) Cartesian system, i.e.,

$$T : (\xi, \eta) \rightarrow (x, y) \quad (18)$$

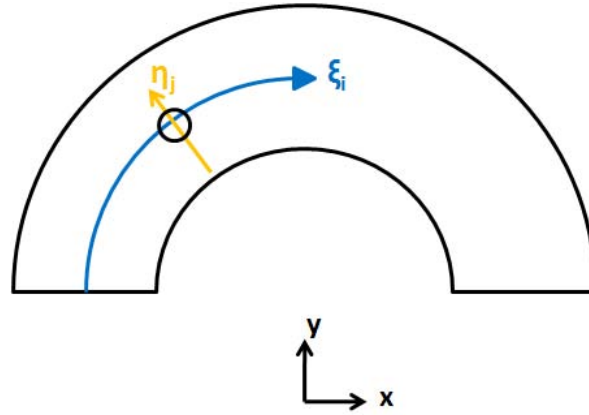


Figure 2: A Body Conforming Coordinate System for a Section of an Annular Ring

The Jacobian matrix for this transformation is written as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \quad (19)$$

The inverse function theorem guarantees that if $|\mathbf{J}| \approx 0$, then the inverse transformation T^{-1} exists such that $T^{-1} : (x, y) \rightarrow (\xi, \eta)$. This transformation has Jacobian matrix \mathbf{J}^{-1} given by

$$\mathbf{J}^{-1} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \frac{1}{|\mathbf{J}|} \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix} \quad (20)$$

where

$$|\mathbf{J}| = x_\xi y_\eta - x_\eta y_\xi \quad (21)$$

Note that each point (ξ, η) assigned within the grid (a simple polar grid, in this case) has a corresponding set of (x, y) coordinates. Moreover, the body conforming grid has real

advantages. Note that the ξ azimuthal coordinate curve is parameterized by the index i while the radial coordinate η is indexed by j . As a result, the computational grid point (ξ_i, η_j) directly coincides with the point $(x_{i,j}, y_{i,j})$ in the physical space. Using this framework, derivatives such as x_ξ and y_η are easily calculated at grid point (ξ_i, η_j) (or (i, j) in shorthand notation), i.e.,

$$x_\xi|_{i,j} \approx \frac{x(i+1, j) - x(i-1, j)}{2} ; \quad y_\eta|_{i,j} \approx \frac{y(i, j+1) - y(i, j-1)}{2} \quad (22)$$

As you may note, the divided differences in (22) contain no obvious measurement of the change in ξ or η . This fact is a testament to the power of this method. The mesh for the computational plane is uniform by definition. Therefore, since both ξ and η are elements of the natural numbers, $\Delta\xi = \Delta\eta = 1$. Since the denominators in (22) are respectively, $2\Delta\xi$ and $2\Delta\eta$, the estimates for these derivatives logically follow. Formulas such as those shown in (22), can then be used in (20) to produce derivatives such as ξ_x and η_y associated with T^{-1} , and the set of transformations has been completely described.

Our segue into coordinate transformations may have seemed to wander off topic, so it is now time to demonstrate its utility for the heat conduction problem. Recall that in (17) the finite volume scheme requires that $\nabla \bar{u}_{i,j}^m$ be evaluated. The gradient operator (11) is difficult to evaluate for the boundaries shown in Figure 2, so let us transform this operator to the body fitted (ξ, η) coordinate system. By using the chain rule, we obtain

$$\nabla u = \hat{i} \left(\frac{\partial \xi}{\partial x} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial u}{\partial \eta} \right) + \hat{j} \left(\frac{\partial \xi}{\partial y} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial u}{\partial \eta} \right) \quad (23)$$

or in our compact notation,

$$\nabla u = \hat{i} (\xi_x u_\xi + \eta_x u_\eta) + \hat{j} (\xi_y u_\xi + \eta_y u_\eta) \quad (24)$$

Although we omitted the averaging overbar in (23) and (24), the gradient is evaluated in terms of cell averaged values for u . Moreover, for the finite volume method, each cell is assumed to possess a single, “averaged” value of u . We assume that there is no internal variation for u inside of the cell. Other numerical methods do permit such variation, (e.g., the Discontinuous Galerkin Method [9]), but those methods are beyond the scope of this report. Furthermore, we just don’t need that much horsepower. Referring to Figure 1, it is important to realize that the gradient is to be evaluated at the center of each cell side (numbered 1 through 4). Also, according to (17), the normal vector for the cell side also participates. Since we have conveniently employed a cell configuration with rectilinear sides, the normal vector remains the same along the chosen side. Hence, we can evaluate $\nabla u \cdot \hat{n}$, an expression for heat flux, as follows.

$$\nabla u \cdot \hat{n} = \hat{n}_x (\xi_x u_\xi + \eta_x u_\eta) + \hat{n}_y (\xi_y u_\xi + \eta_y u_\eta) \quad (25)$$

where the unit normal vector \hat{n} is written as

$$\hat{n} = \hat{i} \hat{n}_x + \hat{j} \hat{n}_y \quad (26)$$

A more computationally practical form is

$$\nabla u \cdot \hat{n} = (\hat{n}_x \xi_x + \hat{n}_y \xi_y) u_\xi + (\hat{n}_x \eta_x + \hat{n}_y \eta_y) u_\eta \quad (27)$$

or

$$\nabla u \cdot \hat{n} = \theta_\xi u_\xi + \theta_\eta u_\eta \quad (28)$$

where

$$\theta_\xi = \hat{n}_x \xi_x + \hat{n}_y \xi_y; \quad \theta_\eta = \hat{n}_x \eta_x + \hat{n}_y \eta_y \quad (29)$$

The forms shown in (28) and (29) are quite useful when developing time-marching schemes for the heat equation. Again, these equations are evaluated at the center point for each cell side to assist in the heat flux discretization.

2.3 Spatial Discretization of Heat Flux

A purpose of the semi-discrete form is to separate the procedures for spatial and temporal discretization allowing a choice of discretization methods. In this section, we present the spatially discretized forms for (28) and (29), along with the cell side length, core features of (17). As a visual aid, Figure 3 is employed along with Figure 2.

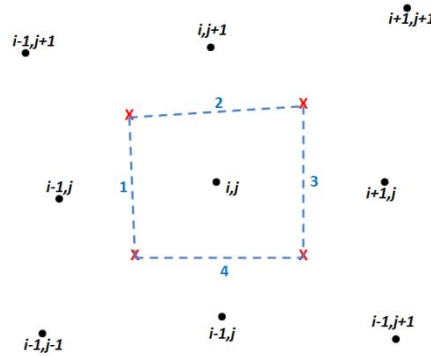


Figure 3: Grid Array for Vertex Centered Finite Volume Cell (i,j)

In Figure 3, the corners of the cell are indicated by red **X's**. The cell sides are drawn in as dashed lines, and grid node (i,j) is situated at the center of cell (i,j). The surrounding grid points needed for various discretization procedures are also included in this diagram. Cell sides 1 through 4 are labeled in blue. For purposes of bookkeeping, in the computational space, the cell corner points have half-node indices. That is to say, the upper left hand corner has grid indices $(i - \frac{1}{2}, j + \frac{1}{2})$; the remaining corners are indexed accordingly. Similarly, the center point for side 1 can be thought of as having grid indices $(i - \frac{1}{2}, j)$. For the vertex centered finite volume mesh, the center of cell (i,j) has the coordinates $(x_{i,j}, y_{i,j})$ or (ξ_i, η_j) . The heat flux term requires discretization on each side of the finite volume cell.

The heat flux discretization for side 1 is conducted as follows. At the center point of the side, we have that

$$(\nabla \bar{u} \cdot \hat{n})l|_{i-1/2,j} = \theta_{\xi} l|_{i-1/2,j} (\bar{u}_{i,j} - \bar{u}_{i-1,j}) + \theta_{\eta} l|_{i-1/2,j} \delta_{\eta} \bar{u}_{i,j} \quad (30)$$

The form of $\delta u_{i,j}$ depends on whether we are employing an explicit or an implicit temporal discretization scheme. In either case, the form of $\delta_{\eta} \bar{u}$ is simple, but it is to be discussed later in this report. For now, we are interested in the calculation of θ_{ξ} and θ_{η} . These equations are given by (29), but the transformation derivatives require the following formulas.

$$x_{\xi} = x_{i,j} - x_{i-1,j} \quad (31)$$

$$y_{\xi} = y_{i,j} - y_{i-1,j} \quad (32)$$

$$x_{\eta} = \frac{1}{4}(x_{i-1,j+1} - x_{i-1,j-1} + x_{i,j+1} - x_{i,j-1}) \quad (33)$$

$$y_{\eta} = \frac{1}{4}(y_{i-1,j+1} - y_{i-1,j-1} + y_{i,j+1} - y_{i,j-1}) \quad (34)$$

The side length and normal vector components may be calculated based upon the locations for the corner points for side 1. The tangent vector directed along side 1 is

$$\vec{\tau} = \vec{x}_{i-1/2,j+1/2} - \vec{x}_{i-1/2,j-1/2} = \hat{i} \tau_x + \hat{j} \tau_y \quad (35)$$

where the notation \vec{x} is simply shorthand representing the (x,y) coordinates of the corner point under consideration. More specifically,

$$\tau_x = x_{i-1/2,j+1/2} - x_{i-1/2,j-1/2} \quad (36)$$

$$\tau_y = y_{i-1/2,j+1/2} - y_{i-1/2,j-1/2} \quad (37)$$

Care must be exercised when calculating the normal vector. The Cartesian coordinate system (x,y,z) is right handed, so it important to ensure that the (ξ,η,z) is also right handed. Note that we have chosen z as our third generalized coordinate since the problem is confined to the plane. Having observed this caution, the normal vector is given by

$$\vec{n} = \vec{\tau} \times \hat{k} \quad (38)$$

where \hat{k} is the unit vector in the z direction. Hence, the unit normal vector components are

$$\hat{n}_x = \frac{-\tau_y}{l}; \quad \hat{n}_y = \frac{\tau_x}{l} \quad (39)$$

where the cell side length l is calculated by using the equation

$$l = \sqrt{\tau_x^2 + \tau_y^2} \quad (40)$$

Equations (31) through (34) are used in (20) and (21) permitting the computation of ξ_x , ξ_y , η_x and η_y . Finally, by using (29), θ_{ξ} and θ_{η} can be easily calculated for cell side 1.

Now consider side 2 as is shown in Figure 3. The heat flux term in (17) may be written as

$$(\nabla \bar{u} \cdot \hat{n})l|_{i,j+1/2} = \theta_{\xi} l|_{i,j+1/2} \delta_{\xi} \bar{u} + \theta_{\eta} l|_{i,j+1/2} (\bar{u}_{i,j+1} - \bar{u}_{i,j}) \quad (41)$$

The calculation of θ_{ξ} and θ_{η} is now centered on the midpoint of side 2. Transformation derivatives and other parameters can be evaluated with the help of the formulas below.

$$x_{\xi} = \frac{1}{4} (x_{i+1,j+1} - x_{i-1,j+1} + x_{i+1,j} - x_{i-1,j}) \quad (42)$$

$$y_{\xi} = \frac{1}{4} (y_{i+1,j+1} - y_{i-1,j+1} + y_{i+1,j} - y_{i-1,j}) \quad (43)$$

$$x_{\eta} = x_{i,j+1} - x_{i,j} \quad (44)$$

$$y_{\eta} = y_{i,j+1} - y_{i,j} \quad (45)$$

The tangent vector is

$$\vec{\tau} = \vec{x}_{i+1/2,j+1/2} - \vec{x}_{i-1/2,j+1/2} = \hat{i} \tau_x + \hat{j} \tau_y \quad (46)$$

The normal vector may be calculated as

$$\vec{n} = \hat{k} \times \vec{\tau} \quad (47)$$

and the unit vector is given by

$$\hat{n}_x = -\frac{\tau_y}{l}; \quad \hat{n}_y = \frac{\tau_x}{l} \quad (48)$$

where, as before, $l = \|\vec{\tau}\|$. The transformation parameters leading to θ_{ξ} and θ_{η} may be calculated by using the equations given in Section 2.2.

For side 3, the expression for the heat flux may be written as

$$(\nabla \bar{u}_{i,j} \cdot \hat{n})l|_{i+1/2,j} = \theta_{\xi} l|_{i+1/2,j} (\bar{u}_{i+1,j} - \bar{u}_{i,j}) + \theta_{\eta} l|_{i+1/2,j} \delta_{\eta} \bar{u} \quad (49)$$

The midpoint of side 3 is the center for the calculation of θ_{ξ} , θ_{η} and the transformation parameters. The following formulas are helpful in the calculations.

$$x_{\xi} = x_{i+1,j} - x_{i,j} \quad (50)$$

$$y_{\xi} = y_{i+1,j} - y_{i,j} \quad (51)$$

$$x_{\eta} = \frac{1}{4} (x_{i,j+1} - x_{i,j-1} + x_{i+1,j+1} - x_{i+1,j-1}) \quad (52)$$

$$y_{\eta} = \frac{1}{4} (y_{i,j+1} - y_{i,j-1} + y_{i+1,j+1} - y_{i+1,j-1}) \quad (53)$$

In our notational convention, the tangent vector for side 3 is calculated as

$$\vec{\tau} = \vec{x}_{i+1/2,j+1/2} - \vec{x}_{i+1/2,j-1/2} = \hat{i} \tau_x + \hat{j} \tau_y \quad (54)$$

A vector cross product provides the associated normal vector, i.e.,

$$\vec{n} = \vec{\tau} \times \hat{k} \quad (55)$$

Hence,

$$\hat{n}_x = \frac{\tau_y}{l}; \quad \hat{n}_y = -\frac{\tau_x}{l} \quad (56)$$

where $l = \|\vec{\tau}\|$. The remaining parameters occurring in (49) are calculated in the same manner as that applied for sides 1 and 2.

The heat flux for side 4 may be calculated in the same way. Naturally, caution must be taken when assigning the orientation of vectors. The formula for the heat flux is

$$(\nabla \bar{u}_{ij} \cdot \hat{n}) l \Big|_{i,j-1/2} = \theta_\xi l \Big|_{i,j-1/2} \delta_\xi \bar{u} + \theta_\eta l \Big|_{i,j-1/2} (\bar{u}_{i,j} - \bar{u}_{i,j-1}) \quad (57)$$

θ_ξ , θ_η and the transformation parameters may be with the assistance of the following formulas.

$$x_\xi = \frac{1}{4}(x_{i+1,j} - x_{i-1,j} + x_{i+1,j-1} - x_{i-1,j-1}) \quad (58)$$

$$y_\xi = \frac{1}{4}(y_{i+1,j} - y_{i-1,j} + y_{i+1,j-1} - y_{i-1,j-1}) \quad (59)$$

$$x_\eta = x_{i,j} - x_{i,j-1} \quad (60)$$

$$y_\eta = y_{i,j} - y_{i,j-1} \quad (61)$$

The tangent vector for side 4 may be calculated as follows.

$$\vec{\tau} = \vec{x}_{i+1/2,j-1/2} - \vec{x}_{i-1/2,j-1/2} = \hat{i} \tau_x + \hat{j} \tau_y \quad (62)$$

The associated normal vector is given by the cross product $\vec{n} = \vec{\tau} \times \hat{k}$; normalized, its components are

$$\hat{n}_x = \frac{\tau_y}{l}; \quad \hat{n}_y = -\frac{\tau_x}{l} \quad (63)$$

The above relations permit the calculation of the heat flux passing through each of the four sides of the quadrilateral finite volume cell illustrated in Figure 3. From the standpoint of geometry, the formulation for heat flux is complete. Still, we have not completely described the contribution of average cell temperature \bar{u} , through terms $\delta \bar{u}$, to the heat flux formulas (30), (41), (49) and (57). The structure of this term is dictated by the time integration scheme.

2.4 Explicit Time Integration

One should not make light of semi-discrete formulas such as (17). This form effectively decouples the temporal and spatial discretization schemes allowing a great deal of flexibility in algorithm selection. The time integration scheme advances the numerical solution for one instant in time to the next. When using an explicit time integration scheme, the numerical solution for \bar{u} at the new time level depends only on the values of \bar{u} at the previous time level. Explicit schemes can be very accurate, but they can also be very difficult to stabilize. That is to say, the truncation error may grow out of control if the time step is too large. For certain problems, explicit schemes can require prohibitively small time steps.[4] Implicit time integration algorithms are structurally different. For this type of procedure, the numerical solution depends

upon the values of \bar{u} at both the old and new time steps. An implicit scheme generally requires solving a system of linear equations. This process can become quite complex and tends to inject numerical viscosity into the integration procedure. On the other hand, explicit schemes, even those composed of multiple steps, are far easier to program and evaluate. However, they possess little numerical viscosity and are particularly unforgiving of programming errors. For these reasons, explicit schemes are ideal for debugging the “physics” represented by the discretized equations.

Our explicit time integration scheme is based upon the modified Euler method.[10] Equation (17) can be written in simple notation, i.e.,

$$\frac{d\bar{u}_{i,j}}{dt} = f(\bar{u}_{i\pm m, j\pm m}) \quad (64)$$

where m is an integer that is selected “cover” the extent of the difference stencil for $\bar{u}_{i,j}$. This equation can be placed in modified Euler form, a Runge-Kutta scheme consisting of two steps, as follows.

$$\begin{aligned} u_{i,j}^* &= u_{i,j}^n + \Delta t f(\bar{u}_{i\pm m, j\pm m}^n) \\ u_{i,j}^{n+1} &= \frac{1}{2}[\bar{u}_{i,j}^n + u_{i,j}^* + \Delta t f(u_{i\pm m, j\pm m}^*)] \end{aligned} \quad (65)$$

where n is the time step counter, i.e., \bar{u}^{n+1} represents the solution at the new time level. This explicit scheme is formally of second order in time, and especially for testing spatial integration schemes, functions quite well. Of course, if higher order temporal accuracy is needed, the classical fourth order Runge-Kutta scheme can be employed.[10] Approximately twice as many floating point operations are required for implementing the fourth order scheme than are required for implementing (65).

Numerically, the right hand side of (64) must be evaluated, in part, through the use of the heat flux equations (30), (41), (49) and (57). In these equations, the terms $\delta_\xi \bar{u}$ and $\delta_\eta \bar{u}$ require description. Equation (30) represents the heat flux through cell side 1 located at computational space coordinates $(i+1/2, j)$. This interface is oriented in the direction of η (or j) coordinate lines, so we cannot build an η directional divided difference at this location. Instead, it is necessary to average difference expressions evaluated at $(i-1, j)$ and (i, j) . Thus,

$$\delta_\eta \bar{u}\big|_{i+1/2, j} = \frac{1}{4}[\bar{u}_{i-1, j+1} - \bar{u}_{i-1, j-1} + \bar{u}_{i, j+1} - \bar{u}_{i, j-1}] \quad (66)$$

For cell side 3 located at computational space coordinates $(i+1/2, j)$, $\delta_\eta \bar{u}$ is evaluated for (41) in a similar manner, i.e.,

$$\delta_\eta \bar{u}\big|_{i+1/2, j} = \frac{1}{4}[\bar{u}_{i+1, j+1} - \bar{u}_{i+1, j-1} + \bar{u}_{i, j+1} - \bar{u}_{i, j-1}] \quad (67)$$

To compute the heat flux located at $(i, j+1/2)$ for side 2, (49) requires a detailed expression for $\delta_\xi \bar{u}$. Here, the direct computation of a divided difference along the ξ coordinate is not possible. For the explicit scheme, averaging can be used to obtain this term, i.e.,

$$\delta_\xi \bar{u} \Big|_{i,j+1/2} = \frac{1}{4} [\bar{u}_{i+1,j+1} - \bar{u}_{i-1,j+1} + \bar{u}_{i+1,j} - \bar{u}_{i-1,j}] \quad (68)$$

Along side 4, located at $(i, j-1/2)$, we employ a similar expression for the divided difference along the cell side.

$$\delta_\xi \bar{u} \Big|_{i,j-1/2} = \frac{1}{4} [\bar{u}_{i+1,j} - \bar{u}_{i-1,j} + \bar{u}_{i+1,j-1} - \bar{u}_{i-1,j-1}] \quad (69)$$

By substituting (66) through (69) into (30), (41), (49) and (57), the function f occurring in (65) can be easily evaluated in support of the explicit scheme. We have that

$$\begin{aligned} f(u_{i\pm m, j\pm m}) = & \theta_\xi \Big|_{i-1/2, j} (u_{i,j} - u_{i-1,j}) + \theta_\eta \Big|_{i-1/2, j} \cdot \frac{1}{4} [u_{i-1,j+1} - u_{i-1,j-1} + u_{i,j+1} - u_{i,j-1}] \\ & + \theta_\xi \Big|_{i, j+1/2} \cdot \frac{1}{4} [u_{i+1,j+1} - u_{i-1,j+1} + u_{i+1,j} - u_{i-1,j}] + \theta_\eta \Big|_{i, j+1/2} (u_{i,j+1} - u_{i,j}) \\ & + \theta_\xi \Big|_{i+1/2, j} (u_{i+1,j} - u_{i,j}) + \theta_\eta \Big|_{i+1/2, j} \cdot \frac{1}{4} [u_{i+1,j+1} - u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1}] \\ & + \theta_\xi \Big|_{i, j-1/2} \cdot \frac{1}{4} [u_{i+1,j} - u_{i-1,j} + u_{i+1,j-1} - u_{i-1,j-1}] + \theta_\eta \Big|_{i, j-1/2} (u_{i,j+1} - u_{i,j}) \end{aligned} \quad (70)$$

Consisting of (65) and the above formula, the explicit solver is easily programmed, yet the time step Δt must be limited to stabilize the scheme. Experience has shown that we can simply enforce the Von Neumann criteria [4] for the single step algorithm, i.e.,

$$\left| \frac{\Delta t}{\min(l_\xi, l_\eta)} \right| < \frac{1}{2} \quad (71)$$

In the context of this restriction, l_ξ and l_η are the lengths of the cell sides existing in the finite volume grid. It follows that Δt must shrink in proportion with the refinement of the grid in order to maintain the stability of the numerical solution.

2.5 Implicit Time Integration

In its most fundamental interpretation, implicit algorithms do not increase the accuracy of a numerical solution. Instead, all of the additional machinery needed by the implicit scheme simply permits the use of larger time steps by limiting the *growth* of numerical truncation and round-off errors. By itself, an implicit scheme can tolerate a great deal of error, yet it does not allow this error to grow. This concept is well illustrated by the work on model problems performed by Risher.[11] On the computer, implicit algorithms are more difficult to program because they usually involve solving systems of linear equations. Linear systems characterized by matrices possessing more than three diagonals are very difficult and computationally expensive to solve. Consider that a typical implicit solver for the 1D heat equation usually involves a tridiagonal matrix. For each additional dimension, we can count on adding at least two diagonals to the matrix. The first step in our implicit discretization procedure is to ensure that we

add only two diagonals per space dimension. To accomplish this goal, we refrain from using stencil averages for derivatives taken parallel to the cell sides. Why? Derivative averaging requires the use of two cell values exterior to the normal stencil. We can eliminate these averages by setting

$$\delta_\eta u|_{i-1/2,j} = \delta_\eta u|_{i+1/2,j} = \delta_\xi u|_{i,j-1/2} = \delta_\xi u|_{i,j+1/2} = 0$$

There is a disadvantage associated with not using stencil averaging. This technique does enhance scheme accuracy, and as is shown in the next section of this report, implicit schemes suffer this loss of precision. On the other hand, the enhanced stability provided by the implicit scheme does facilitate the use of larger time steps and produces steady state solutions more quickly.

Even if stencil averages are not used, without some simplification, a solution of the heat equation in three dimensions requires contending with a heptadiagonal matrix, not an inviting prospect. Instead, we employ a process called factorization to reduce the solution of a pentadiagonal matrix for the two-dimensional problem to solving a sequence of two tridiagonal matrix problems. The resulting method is referred to as an Alternating Direction Implicit (ADI) scheme.

To begin the derivation of our ADI scheme, we first write the combined heat flux term $f(\bar{u}_{i\pm m, j\pm m})$ as

$$f(\bar{u}_{i\pm m, j\pm m}) = (\Delta_\xi + \Delta_\eta) u_{i,j}^{n+1} \quad (72)$$

where,

$$V_{i,j} \Delta_\xi = \theta_\xi l|_{i+1/2,j} (i^+) + \theta_\xi l|_{i-1/2,j} - \theta_\xi l|_{i+1/2,j} - \theta_\xi l|_{i-1/2,j} (i^-) \quad (73)$$

$$V_{i,j} \Delta_\eta = \theta_\eta l|_{i,j+1/2} (j^+) + \theta_\eta l|_{i,j-1/2} - \theta_\eta l|_{i,j+1/2} - \theta_\eta l|_{i,j-1/2} (j^-) \quad (74)$$

The right side of (72) has been expressed in terms of the sum of two difference operators. No additional approximations have been made to obtain this form, but it does perform the important step of separating the directional difference terms. Equations (73) and (74) contain the formulas for these differences where θ_ξ and θ_η are obtained from (29) and the discussions in Section 2.3.

These expressions also contain the notation for the index shift operators (i^{+or-}) and (j^{+or-}) . For example, the (i^+) operator functions as

$$(i^+) \bar{u}_{i,j} = \bar{u}_{i+1,j} \quad (75)$$

With the use of (72), (64) can be discretized with the fully implicit form

$$\bar{u}_{i,j}^{n+1} - \bar{u}_{i,j}^n = \Delta t (\Delta_\xi + \Delta_\eta) \bar{u}_{i,j}^{n+1} \quad (76)$$

In operator form, (76) is written as

$$[1 - \Delta t (\Delta_\xi + \Delta_\eta)] \bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^n \quad (77)$$

Note that in (77), $\bar{u}_{i,j}$ at time level $n+1$ depends on other values of \bar{u} at other values on the space stencil existing at the same time level. It is therefore not possible to evaluate $\bar{u}_{i,j}^{n+1}$ knowing only the values $\bar{u}_{i,j}^n$. As a result, we must simultaneously solve for $\bar{u}_{i,j}^{n+1}$ at all values of i and j on the space grid since these values are *implicitly* interrelated. Although it induces some error within the numerical approximation, let us rewrite (77) as follows.

$$[1 - \Delta t \Delta_\xi - \Delta t \Delta_\eta + \Delta t^2 \Delta_\xi \Delta_\eta] \bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^n + \Delta t^2 \Delta_\xi \Delta_\eta \bar{u}_{i,j}^n + O(\Delta t^2) \quad (78)$$

The product $\Delta_\xi \Delta_\eta$ simply implies that the Δ_ξ and Δ_η operators are applied in sequence. We have also assumed that

$$\Delta t^2 \Delta_\xi \Delta_\eta \bar{u}_{i,j}^{n+1} \cong \Delta t^2 \Delta_\xi \Delta_\eta \bar{u}_{i,j}^n \quad (79)$$

Moreover, this term is $O(\Delta t^2)$, an error that is generally acceptable given that the other difference terms are $O(\Delta t)$. Using factorization, (78) can be replaced by

$$[1 - \Delta t \Delta_\xi][1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^n + \Delta t^2 \Delta_\xi \Delta_\eta \bar{u}_{i,j}^n + O(\Delta t^2) \quad (80)$$

where the $O(\Delta t^2)$ has been carried forward to represent the approximation error incurred in (78). In the course of deriving other implicit time integration schemes, (80) may be replaced by other formulas that are equally valid. Note that the current form is used for this specific derivation. We continue by writing

$$[1 - \Delta t \Delta_\xi][1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^n + \Delta t \Delta_\eta \bar{u}_{i,j}^n + \Delta t^2 \Delta_\xi \Delta_\eta \bar{u}_{i,j}^n - \Delta t \Delta_\eta \bar{u}_{i,j}^n \quad (81)$$

Simplifying (81) we obtain

$$[1 - \Delta t \Delta_\xi][1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = (1 + \Delta t \Delta_\eta) \bar{u}_{i,j}^n - (1 - \Delta t \Delta_\xi) \Delta t \Delta_\eta \bar{u}_{i,j}^n \quad (82)$$

Now let

$$(1 + \Delta t \Delta_\eta) \bar{u}_{i,j}^n = [1 - \Delta t \Delta_\xi] \bar{u}_{i,j}^* \quad (83)$$

Although not rigorously so, the $\bar{u}_{i,j}^*$ are accepted as existing at a time level intermediate to n and $n+1$. In truth, the $\bar{u}_{i,j}^*$ are intermediate values (but not in time) resulting at the end of step one of a two-step approximation. By substituting (83) into (82), we obtain

$$[1 - \Delta t \Delta_\xi][1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = (1 - \Delta t \Delta_\xi) \bar{u}_{i,j}^* - (1 - \Delta t \Delta_\xi) \Delta t \Delta_\eta \bar{u}_{i,j}^n \quad (84)$$

or

$$[1 - \Delta t \Delta_\xi][1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = [1 - \Delta t \Delta_\xi] (\bar{u}_{i,j}^* - \Delta t \Delta_\eta \bar{u}_{i,j}^n) \quad (85)$$

By equating the arguments for the common operator in (85),

$$[1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^* - \Delta t \Delta_\eta \bar{u}_{i,j}^n + O(\Delta t^2) \quad (86)$$

Equations (83) and (86) constitute the Douglas-Rachford ADI scheme [4] for curvilinear coordinate systems. This two-step time integration scheme is executed in the following order.

$$\text{Step 1: } [1 - \Delta t \Delta_\xi] \bar{u}_{i,j}^* = (1 + \Delta t \Delta_\eta) \bar{u}_{i,j}^n \quad (87)$$

$$\text{Step 2: } [1 - \Delta t \Delta_\eta] \bar{u}_{i,j}^{n+1} = \bar{u}_{i,j}^* - \Delta t \Delta_\eta \bar{u}_{i,j}^n \quad (88)$$

Note that the scheme implicitly sweeps first in the ξ direction (Step 1), and second in the η direction (Step 2). The alternating of implicit sweep directions is characteristic of ADI schemes. Again, it must be emphasized that the overall scheme is locally accurate to $O(\Delta t^2)$.

2.6 Boundary Conditions

No discussion of techniques for solving IBVPs is complete without describing how boundary conditions are implemented. This statement is particularly true for numerical methods since the programming of boundary condition algorithms can become complicated. For the problems solved later in this report, two types of boundary conditions are addressed: (i) the Dirichlet condition (\bar{u} is prescribed on the boundary at all times), and (ii) the Neumann condition ($\partial \bar{u} / \partial \hat{n}$ is prescribed at every point on the boundary at all times). For heat conduction problems, the Dirichlet condition sets the actual temperature at each point on the boundary. On the other hand, the Neumann condition sets the normal heat flux (or temperature gradient) at each point on the boundary. To illustrate the means by which these conditions are enforced, consider that Figure 4 contains the domain of concern. The “ i ” coordinate lines form arcs in this

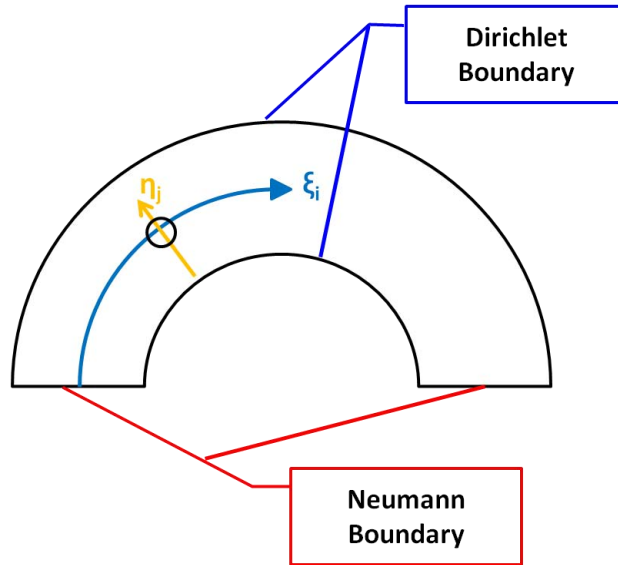


Figure 4: Section of an Annular Ring with Both Dirichlet and Neumann Boundary Conditions

figure. Assume that the attendant grid lines are indexed by $i = 1, \dots, IMAX$. In the radial direction lie the “ j ” coordinate lines indexed by $j = 1, \dots, JMAX$. Let us enforce a Dirichlet condition at the endpoints of the j coordinate lines. For the explicit time integration scheme this condition is implemented as follows. At the conclusion of each time step, make the following settings.

$$\bar{u}_{i,1} = u_{inner} ; \quad u_{i,1}^* = u_{inner} \quad (89)$$

$$\bar{u}_{i,JMAX} = u_{outer} ; \quad u_{i,JMAX}^* = u_{outer} \quad (90)$$

for $i=1, \dots, IMAX$. For the Douglas-Rachford ADI scheme, implementing this condition is more complicated. The Dirichlet condition is used for Step 2 and can be written as

$$\Delta t \theta_{\eta} \Big|_{i,j-1/2} \bar{u}_{i,j-1}^{n+1} + (1 - \Delta t \theta_{\eta} \Big|_{i,j-1/2} + \Delta t \theta_{\eta} \Big|_{j+1/2}) \bar{u}_{i,j}^{n+1} - \Delta t \theta_{\eta} \Big|_{i,j+1/2} \bar{u}_{i,j+1}^{n+1} = R_{i,j}^2 \quad (91)$$

for $i = 1, \dots, IMAX$, $j = 2, \dots, JMAX-1$. $R_{i,j}^2$ simply represents the known terms on the right hand side of (88). Specific equations involving boundary values arise when $j = 2$ and $j = JMAX-1$. The restrictions prescribed by (89) and (90) are still applicable for (91). That is to say, the term $\Delta t \theta_{\eta} \Big|_{i,j-1/2} \bar{u}_{i,j-1}^{n+1} = \Delta t \theta_{\eta} \Big|_{i,j-1/2} u_{inner}$ is known and hence moved to the right hand side of (91). The modified equation becomes

$$(1 - \Delta t \theta_{\eta} \Big|_{i,2-1/2} + \Delta t \theta_{\eta} \Big|_{2+1/2}) \bar{u}_{i,2}^{n+1} - \Delta t \theta_{\eta} \Big|_{i,2+1/2} \bar{u}_{i,3}^{n+1} = R_{i,2}^2 - \Delta t \theta_{\eta} \Big|_{i,2-1/2} \bar{u}_{inner} \quad (92)$$

In the same manner, an equation can be obtained for the case when $j = JMAX-1$, i.e.,

$$\begin{aligned} \Delta t \theta_{\eta} \Big|_{JMAX-1-1/2} \bar{u}_{i,JMAX-2}^{n+1} + (1 - \Delta t \theta_{\eta} \Big|_{JMAX-1-1/2} + \Delta t \theta_{\eta} \Big|_{JMAX-1+1/2}) \bar{u}_{i,JMAX-1}^{n+1} &= R_{i,JMAX-1}^2 \\ &+ \Delta t \theta_{\eta} \Big|_{JMAX-1+1/2} u_{outer} \end{aligned} \quad (93)$$

There are only two diagonal terms in each of equations (92) and (93). When these equations are combined with (91) (applied for $i = 2, \dots, IMAX-1$, $j = 3, \dots, JMAX-2$), a tridiagonal linear system is obtained for Step 2. This ADI step can be easily solved by using the Thomas Algorithm described in Appendix A.

The Neumann boundary condition is applied along the end rays of the annulus shown in Figure 4. For the problems solved in the next section, we designate this boundary as an insulator. That is to say, the heat flux across this boundary is zero. This condition is enforced by setting

$$\bar{u}_{1,j}^{n+1} = \bar{u}_{2,j}^{n+1} ; \quad \bar{u}_{1,j}^* = \bar{u}_{2,j}^* \quad (94)$$

$$\bar{u}_{IMAX,j}^{n+1} = \bar{u}_{IMAX-1,j}^{n+1} ; \quad \bar{u}_{IMAX,j}^* = \bar{u}_{IMAX-1,j}^* \quad (95)$$

for $j = 2, \dots, JMAX-1$. Step 1 of the ADI scheme requires this boundary condition; its difference equation is written as

$$\Delta t \theta_{\xi} \Big|_{i-1/2,j} \bar{u}_{i-1,j}^* + (1 - \Delta t \theta_{\xi} \Big|_{i-1/2,j} + \Delta t \theta_{\xi} \Big|_{i+1/2,j}) \bar{u}_{i,j}^* - \Delta t \theta_{\xi} \Big|_{i+1/2,j} \bar{u}_{i+1,j}^* = R_{i,j}^1 \quad (96)$$

for $i = 2, \dots, IMAX-1$, $j = 2, \dots, JMAX-1$. $R_{i,j}^1$ simply represents the known terms existing on the right hand side of the equation. Consider the case where $i = 2$; the specific difference equation is written as

$$\Delta t \theta_{\xi} \Big|_{2-1/2,j} \bar{u}_{1,j}^* + (1 - \Delta t \theta_{\xi} \Big|_{2-1/2,j} + \Delta t \theta_{\xi} \Big|_{2+1/2,j}) \bar{u}_{2,j}^* - \Delta t \theta_{\xi} \Big|_{2+1/2,j} \bar{u}_{3,j}^* = R_{2,j}^1 \quad (97)$$

To transform (97) into a proper boundary equation, we enforce the rightmost equation in (94). In particular, $\bar{u}_{2,j}^*$ takes the place of $\bar{u}_{1,j}^*$, but in this case, there is no migration of terms to the right hand side of the equation. After simplifying, we obtain

$$(1 + \Delta t \theta_{\xi} \Big|_{2+1/2}) \bar{u}_{2,j}^* - \Delta t \theta_{\xi} \Big|_{2+1/2} \bar{u}_{3,j}^* = R_{2,j}^1 \quad (97)$$

With the use of (94), a similar equation can be derived for the case where $i = IMAX-1$. To be brief, we simply state the result.

$$\Delta t \theta_{\eta} \Big|_{IMAX-1-1/2} \bar{u}_{IMAX-2,j}^* + (1 - \Delta t \theta_{\eta} \Big|_{IMAX-1-1/2}) \bar{u}_{IMAX-1,j}^* = R_{IMAX-1,j}^2 \quad (98)$$

The system formed by (96) for $i = 3, \dots, IMAX-2$ and $j = 2, \dots, JMAX-1$ is tridiagonal in structure and may be solved by the method described in Appendix A.

3.0 RESULTS AND DISCUSSION

Although the algorithms presented in Section 2 of this report are based upon archival methods, this effort's end product differs from many conventional heat transfer methods. This claim is particularly true for the ADI scheme developed for generalized coordinates. As a result, our algorithms require testing, or in more appropriate terms, validation. In the following discussions, the new scheme is tested for three heat conduction problems cast in two dimensions. Each problem tests a different aspect of the scheme's numerical performance. Moreover, each problem possesses an exact solution that can be compared one on one with the numerical solution.

3.1 Rectangular Asymmetric Heating Problem

The first test problem is designed to identify programming errors for the finite volume scheme. Its rectangular Cartesian geometry is extremely easy to grid, and the associated transformation terms and normal vector components are easily calculated on paper for comparison with the code calculations. The associated IBVP is presented below.

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad 0 < x < 100, \quad 0 < y < 100, \quad t > 0 \\ u &= u(x, y, t) \\ u(x, 0, t) &= 25; \quad u(x, 100, t) = 100 \\ \frac{\partial u(0, y, t)}{\partial t} &= 0; \quad \frac{\partial u(100, y, t)}{\partial t} = 0 \\ u(x, y, 0) &= 0\end{aligned}\tag{99}$$

The IBVP (99) is defined over a large domain with Dirichlet boundary conditions enforced at $y=0$ and $y=100$. Neumann (insulator) boundary conditions are imposed over the linear boundaries located at $x=0$ and $x=100$. For this problem, we have generated a uniform grid with $\Delta x = \Delta y = 1$. Note that unit spatial step sizes are frequently used for “benchmarking” numerical solvers. Remarkably, we can use a time step size of 1 unit and still maintain algorithm stability. The maximum solution time is set at 2500 units. Note that this problem is completely dimensionless. Since insulators are set at $x=0, 100$, it is evident that the solution for (99) varies only in the y direction. That is to say, the solution is the same at every value of x . As a result, the exact solution is given by the solution of the heat equation in one dimension with the same Dirichlet boundary conditions. The exact solution for this problem is derived in Appendix B. A time sequence of numerical solutions is shown in Figure 5. At each distinct time, the numerical solution is plotted against the corresponding exact solution. As one may surmise from the dimensions of the domain, the finite volume grid consists of mere 10,000 cells, not a large mesh by today's standards. As Figure 5 illustrates, there is excellent agreement between the numerical and exact solutions at each time level. Also, as is evidenced by Figure 6 at time 1500, the numerical solution exhibits the required uniformity in the x direction. The temperature contours (or contours of u) essentially overlap one another. This type of performance indicates that the finite volume scheme is operating properly. There is no indication of the presence of numerical

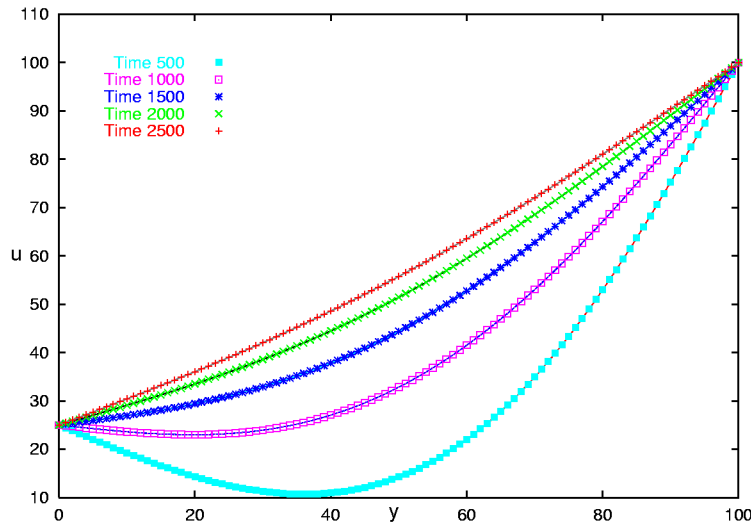


Figure 5: Problem 1 Solutions Calculated by Using the Douglas-Rachford ADI Scheme for Curvilinear Coordinates. Numerical Solutions are Represented by Solid Lines. Exact Solutions are Represented by Symbols.

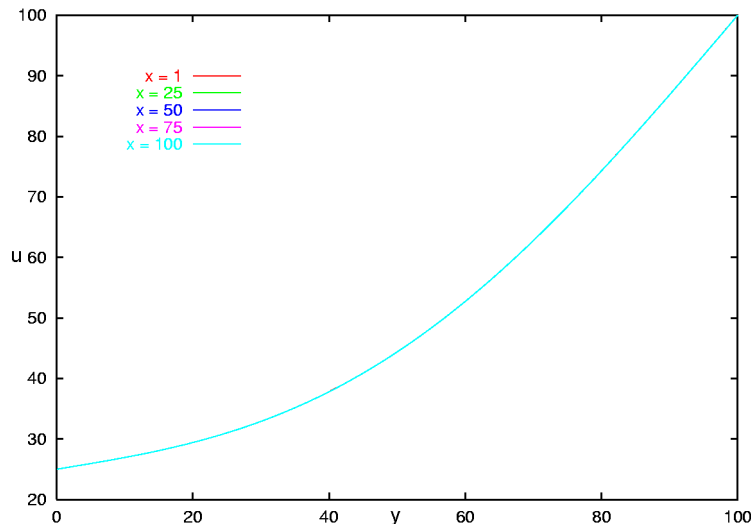


Figure 6: Problem 1 Solutions Calculated by Using the Douglas-Rachford ADI Scheme for Curvilinear Coordinates. Solutions are Compared Along Lines of Constant x at Time 1500.

“heat leaks” from the grid, so the programming seems to contain no flaws. Only minor differences appear in the curves near times 41, 82 and 97. For these reasons, we can conclude that the numerical scheme performs quite well for Problem 1.

3.2 Radial Asymmetric Heating of an Annular Section

The previous test case is designed to test the fundamental structure of the finite volume scheme. It also validates the basic time scheme, but only for the Cartesian grid where most numerical schemes excel. In this case, we have designed a test problem that will validate both the explicit and implicit schemes for a non-Cartesian geometry. In this case, the test case is defined on an annular sector as shown in Figure 4. The azimuthal extent for the problem is only 90°

instead of the pictured 180° , yet the boundary conditions are enforced as illustrated. The inner radius of the annulus is one while the outer radius is two. The IBVP is presented below.

$$\begin{aligned}
 \frac{\partial u}{\partial t} &= \nabla^2 u; \quad 1 < r < 2; \quad \frac{\pi}{4} < \theta < \frac{3\pi}{4}; \quad t > 0 \\
 u &= u(r, \theta, t) \\
 u(1, \theta, t) &= 25; \quad u(2, \theta, t) = 100 \\
 \frac{\partial u(r, \pi/4, t)}{\partial \theta} &= \frac{\partial u(r, 3\pi/4, t)}{\partial \theta} = 0 \\
 u(r, \theta, 0) &= 0
 \end{aligned} \tag{100}$$

An examination of the IBVP reveals that this problem is azimuthally symmetric. Still, the finite volume algorithm is programmed in a general manner. It computes the solution throughout the curved two-dimensional field. This problem constitutes an excellent test of the algorithm since we can easily compare solutions along annular radial lines to reveal any bias associated with the space scheme. As in the case of Problem 1, we utilize a 100×100 grid with uniform radial and angular increments. The results are illustrated in Figure 7, a plot of the explicit and ADI numerical solutions against the exact solution for times 0.1, 0.2 and 1.0. In each case, the explicit solution agrees extremely well with the exact solution, and the ADI solution also performs well. As was discussed earlier, the ADI scheme does suffer from a reduction in accuracy due to: (i) the lack of derivative stencil averaging, and (ii) the ADI factorization error. It is nonetheless important to realize that the ADI error is not cumulative. The error does not increase in time; also, the ADI solution converges to the steady state solution with accuracy equaling that of the explicit scheme.

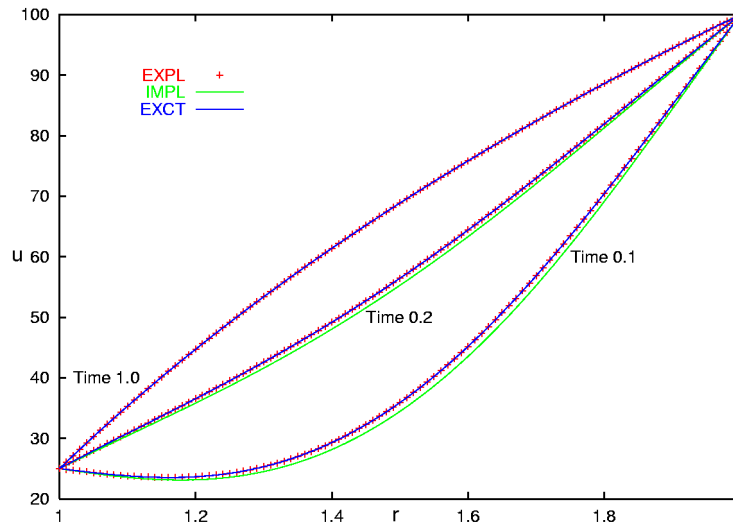


Figure 7: Problem 2 Time Sequence of Radial Solution Plots. The Explicit (EXPL) and ADI (IMPL) Numerical Solutions are Compared Against the Exact (EXCT) Solution at Times 0.1, 0.2 and 1.0.

The accuracy of the ADI solution improves as time advances; ultimately, it plots on top of the exact solution. A derivation of the exact solution for Problem 2 is provided in Appendix C. Figures 8 examines the level of azimuthal symmetry that is exhibited by the numerical solutions.

Based upon the boundary conditions of the IBVP, the exact solution is perfectly symmetric. Both explicit and implicit numerical solutions, possess a high level of symmetry. In Figure 8, we plot the explicit solution at time 0.1 cast along radii located at 45° , 67.5° , 90° , 112.5° and 135° . There is no discernible difference between the solutions along these rays. Figure 8 also contains a plot of the explicit solutions at time 0.2 along the same rays. In truth, there are differences between

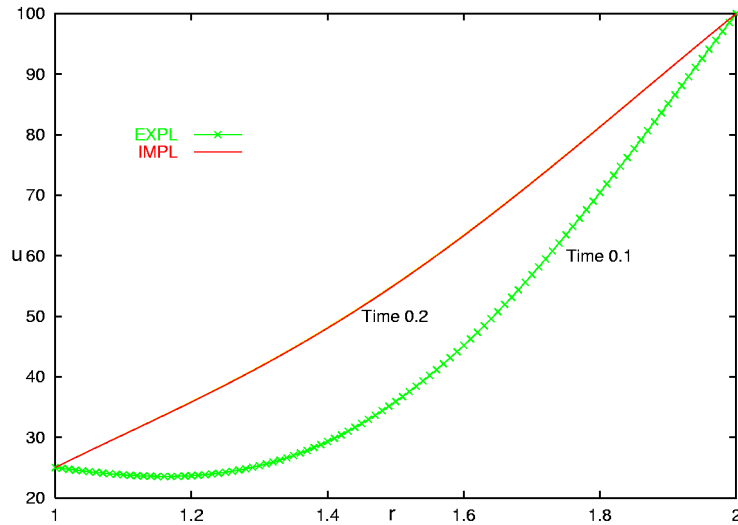


Figure 8: Problem 2 Analysis of Azimuthal Symmetry Occurring in Numerical Solutions. Plots are Shown for the Explicit (EXPL) Solution at Time 0.1 and for the ADI (IMPL) Solution at time 0.2. In Each Case, Numerical Solutions are Graphed Along the Annular Rays Located at 45° , 67.5° , 90° , 112.5° and 135° .

the ADI solutions, but the differences are very small and attributable to the errors inherent in the approximations used to create the ADI scheme. The ensuing errors are greater in magnitude for curvilinear grids than for Cartesian grids due to the effect of metric scaling. Still, both the explicit and implicit solutions exhibit a high degree of azimuthal symmetry and thus perform very well in this test.

3.3 Radially and Azimuthally Asymmetric Heating of an Annular Section

So far, we have successfully tested our finite volume numerical scheme for accuracy in time and for spatial accuracy on non-Cartesian grids. This final test is designed to determine whether or not the scheme can cope with asymmetric azimuthal boundary conditions. This test case is more difficult than either of its predecessors. In this case, the outer radial boundary condition is permitted to vary with θ while the inner boundary value is held fixed. The IBVP is described below. All problem parameters are dimensionless.

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \nabla^2 u; \quad 1 < r < 2; \quad \frac{\pi}{4} < \theta < \frac{3\pi}{4}; \quad t > 0 \\
u &= u(r, \theta, t) \\
u(1, \theta, t) &= 1; \quad u(2, \theta, t) = 2 + 2 \left(\frac{\theta - \pi/4}{\pi/2} \right) \\
\frac{\partial u(1, \theta, t)}{\partial \theta} &= \frac{\partial u(2, \theta, t)}{\partial \theta} = 0 \\
u(r, \theta, 0) &= 0
\end{aligned} \tag{101}$$

The exact, steady state solution for this problem is derived in Appendix D. The exact, time dependent solution has not been computed because of its complexity, and also in the prior test cases, the time scheme performs very well. Instead, this test case concentrates on validating the computer programming for steady state convergence. Based upon the results shown for the other test cases, the results show below are for the ADI scheme. The explicit method is expected to perform well in this problem, so those results are omitted. Figure 9 contains a sequence of plots cast along the radial lines (rays) located at θ equal to 45° , 67.5° , 90° , 112.5° and 135° . Along each ray, the numerical ADI solution is compared with the exact solution.

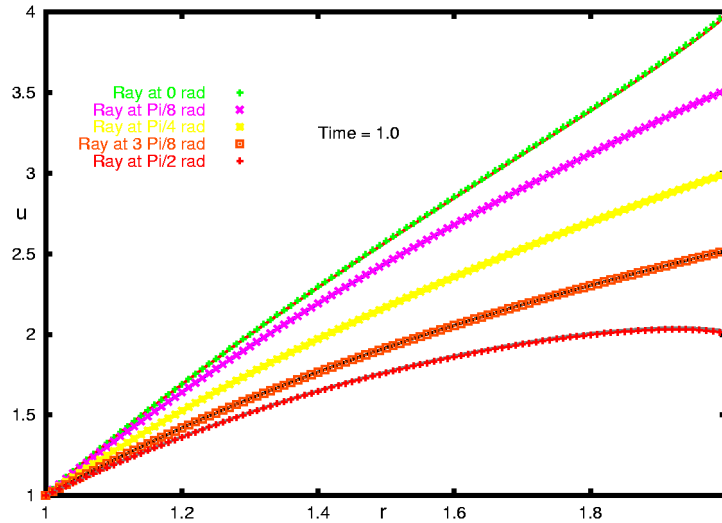


Figure 9: Problem 3 Comparisons of the Numerical ADI and Exact Solutions. Solutions are Plotted Along Radial Lines Located at 45° , 67.5° , 90° , 112.5° and 135° .

Along each ray, the numerical solution exhibits very good agreement with the exact solution. The poorest agreement is perhaps obtained nearest the hot outer radial boundary. In this region, the computer program is required to conform to the natural boundary temperature gradient across only one finite volume cell. It is expected that this performance may be improved by refining the grid near the outer boundary. Other aspects of the exact solution are captured quite well. It is also interesting to view color contour plots of a time sequence of numerical solutions. These plots are shown in Figure 10 for solution times 0.1, 0.2, 0.3 and 0.5. The snapshot at time 0.5 is included since it represents the nearly converged steady state solution. The heat conduction solution behaves, not surprisingly, like a classic diffusion problem. Near the hot boundary, the

temperature changes very quickly driven by high temperature gradients. The effect is shown in the first three plots in Figure 10. Notice how quickly the high temperature field propagates through the annular section. Observe also that the lower temperature gradient near the right side of the annulus causes a slower temperature response in time. Further consideration of this figure reveals that the temperature contours change very little between times 0.3 and 0.5. The reason for the “slower” propagation of the temperature field is caused by the physics of the heat equation. The time dependent term for solutions of this equation has an exponential dependency. That is to say, for a small time interval, this term causes large changes in the solution. Yet, for large time intervals, the time dependent term follows an asymptote that approaches zero, so at ever larger times, the time dependent term has less effect. Hence, the temperature field changes more slowly. A parallel effect, particularly significant at early times, is driven by the temperature gradient. High gradients motivate large local changes in the temperature field. As we discussed earlier, this effect is evident at times 0.1 and 0.2. Based upon this evidence, we conclude that our ADI solver is validated for two-dimensional heat conduction problems on curvilinear geometries.

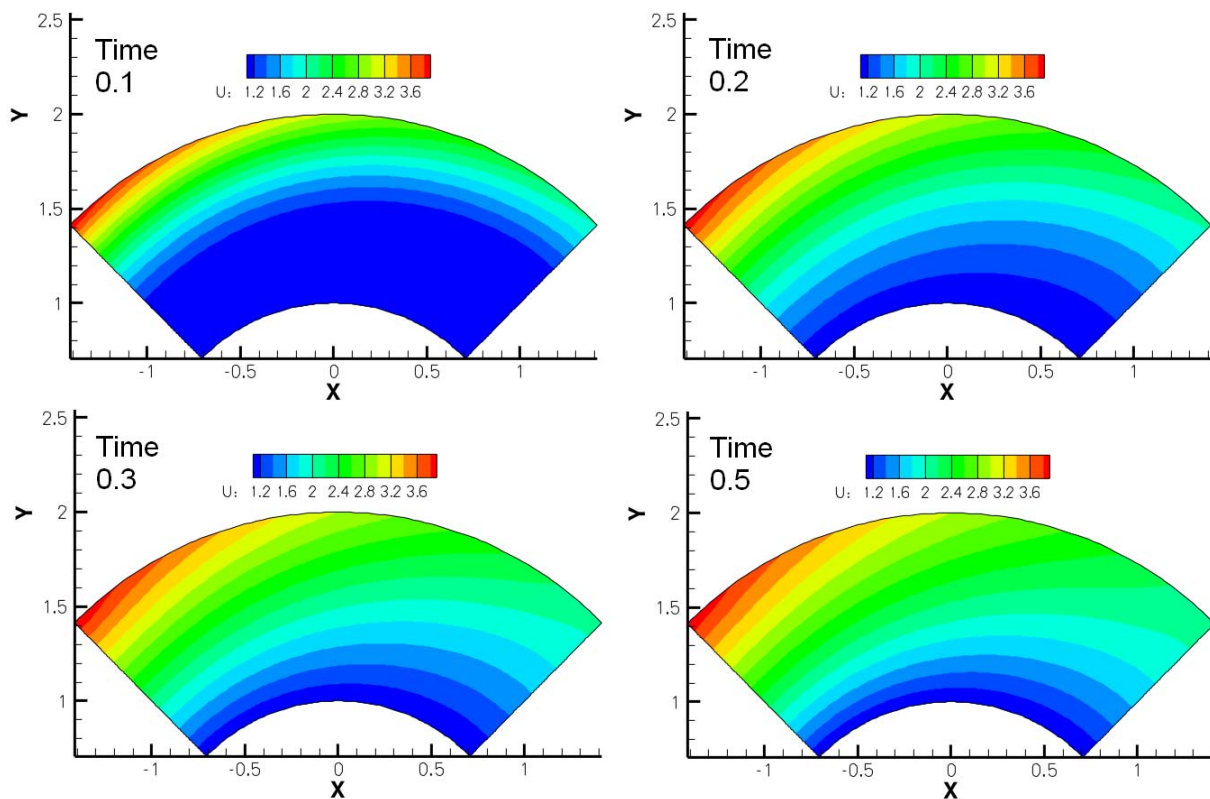


Figure 10: Problem 3 Color Contour Plots of the Temperature Field on an Annular Section. Plots of the ADI Numerical Solution are Shown at Times 0.1, 0.2, 0.3 and 0.5. All Solution Parameters are Dimensionless.

The finite volume ADI method produces solutions that capture the time dependent effects of the heat conduction equation. Also, this algorithm is stable and produces good spatial accuracy, especially considering the approximations necessary to implement the implicit scheme for curvilinear grids.

4.0 CONCLUSIONS

This report has described the numerical methodology behind a new implicit heat conduction equation solver based upon the finite volume method for curvilinear geometries. A principal strength of this method is that it permits the accurate enforcement of boundary conditions along curved boundaries without the need for complicated interpolation or interface tracking schemes. The application of the finite volume method allows an effective reduction of order for the spatial discretization scheme avoiding the need for discretizing the Laplace operator in generalized coordinates. Also, since it is unnecessary to differentiate basis vectors in the local coordinate system, the scheme remains highly accurate with far fewer geometric calculations. However, caution must be used when applying this scheme for problems on like topologies with cell shapes that change drastically within comparatively small regions of the grid. Without adequate mesh resolution in these locales, error can accumulate rapidly. A prototypical computer program with both explicit and implicit time integration schemes has been developed for curvilinear coordinates in two dimensions and validated against a set of exact solutions for an annular geometry. Our numerical solutions have proven to be very accurate when compared with the exact solutions. The explicit solver is highly accurate, and the parts of the ADI algorithm causing higher error have been identified. Based upon its simple finite volume, semi-discrete form, this algorithm is readily adaptable for problems set in three curvilinear dimensions. This algorithm has many potential applications for the analysis of engineering systems.

REFERENCES

1. Shames, Irving, Introduction to Solid Mechanics, Prentice Hall, Englewood Cliffs, N.J., 1975, pp.69-71.
2. Menzel, Donald, Mathematical Physics, Dover, New York, N.Y., 1961, pp. 224-231.
3. Kays, William and Crawford, M., Convective Heat and Mass Transfer, McGraw-Hill, New York, N.Y., 1980, pp. 1-3.
4. Mitchell, Andrew and Griffiths, D., The Finite Difference Method in Partial Differential Equations, Wiley, New York, N.Y., 1980, pp. 54-58.
5. Liu, T.G., Khoo, B.C. and Yeo, K.S., "Ghost Fluid Method for Strong Shock Impacting on Material Interface", *Journal of Computational Physics*, **190**, 2003, pp. 651-681.
6. Özisik, Necati, Boundary Value Problems of Heat Conduction, Dover, Mineola, N.Y., 1968, pp. 2-6.
7. Burden, Richard, Faires, J. Douglas and Reynolds, Albert, Numerical Analysis, 2nd Ed., Prindle, Weber & Schmidt, Boston, MA, 1981, pp. 125-128.
8. Hirsch, Charles, Numerical Computation of Internal and External Flows, Vol. 1, Wiley, New York, N.Y., 1988, pp. 237-264.
9. Cockburn, Bernardo, Karniadakis, George and Shu, Chi-Wang, "The Development of Discontinuous Methods", Discontinuous Galerkin Methods, Cockburn, B., Karniadakis, G. and Shu, C.-W., Eds., Springer, New York, N.Y., 2000, pp. 1-50.
10. Burden, Richard , Faires, J. Douglas and Reynolds, Albert, Numerical Analysis, 2nd Ed., Prindle, Weber & Schmidt, Boston, Massachusetts, 1981, pp. 304-317.
11. Risher, Gary, *One-Dimensional Numerical Analysis of the Heat Equation with Symmetric Cooling and Asymmetric Heating*, Technical Memorandum, Munitions Directorate, Air Force Research Laboratory, March 2010.
12. Weinberger, Hans, A First Course in Partial Differential Equations. John Wiley & Sons, New York, N.Y., 1965.

APPENDIX A

The Thomas Algorithm

The Thomas Algorithm is used to solve a system of linear equations that be written in the form

$$\mathbf{M}\vec{u} = \vec{R} \quad (102)$$

where \mathbf{M} is a tridiagonal matrix. This matrix has a form such as

$$\mathbf{M} = \begin{bmatrix} b_2 & c_2 & & & \\ a_3 & b_3 & c_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{I-2} & b_{I-2} & c_{I-2} \\ & & & a_{I-1} & b_{I-1} \end{bmatrix} \quad (103)$$

All entries other than the a_i , b_i and c_i are zero. The index structure, i.e., $i = 2, \dots, I-1$, is used to mimic the grid indices used in the ADI scheme discussed in Sections 2.5 and 2.6. Here, the maximum index I corresponds to either $IMAX$ or $JMAX$. The solution vector \vec{u} has the form

$$\vec{u} = [u_2, \dots, u_{I-1}]^T \quad (104)$$

while the vector of knowns \vec{R} has the form

$$\vec{R} = [R_2, \dots, R_{I-1}]^T \quad (105)$$

The Thomas Algorithm functions by applying Gaussian Elimination to this linear system. An equivalent result may also be obtained by the use of LU-Factorization. In either case, for a tridiagonal matrix \mathbf{M} that is (i) sufficiently well conditioned and (ii) free of zero entries on the diagonal, a solution for the linear system may be obtained in two steps.

Step 1 is referred to as forward elimination and is described by the following algorithm. Define $x_1 = y_1 = 0$ with $a_2 = c_{I-1} = 0$. For $i = 2, \dots, IMAX-1$, compute the following quantities:

$$x_i = \frac{R_i - a_i x_{i-1}}{a_i y_{i-1} + b_i}; \quad y_i = \frac{-c_i}{a_i y_{i-1} + b_i} \quad (106)$$

Clearly, x_i depends upon y_{i-1} , so the quantities must be calculated together in a loop. Step 2 of this algorithm is referred to as backward substitution and is described as follows. At the conclusion of forward elimination, it is immediately evident that

$$u_{I-1} = x_{I-1} \quad (107)$$

The rest of the solution is given by calculating for $i = I-2, \dots, 2$,

$$u_i = x_i + y_i u_{i+1} \quad (108)$$

APPENDIX B

Solution for Heat Equation with Asymmetric Heating in One Cartesian Dimension

In many cases, the heat equation is very easy to solve for one-dimensional Cartesian coordinates by using separation of variables, but for the asymmetric heating problem described below, this method requires modification. This IBVP is written as follows:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, \quad t > 0 \\ u &= u(x, t) \\ u(0, t) &= u_0 \neq 0; \quad u(L, t) = u_L \neq 0; \quad u_0 \neq u_L \text{ (in general)} \\ u(x, 0) &= 0\end{aligned}\tag{109}$$

The enforcement of non-zero boundary conditions presents problems for the separation of variables method. That is to say, if u is zero on a boundary, then this condition can be enforced by making the space dependent part of the solution zero at the same location. For a non-zero value, this technique will not work. Fortunately, we can work around this difficulty. Equation (109) involves a linear partial differential equation, so it accepts superposed solutions. As a result, we can write $u(x, t)$ as the sum of a purely space dependent function $u_s(x)$ and a time and space dependent function $\hat{u}(x, t)$, i.e.,

$$u(x, t) = u_s(x) + \hat{u}(x, t), \quad 0 \leq x \leq L, \quad t \geq 0\tag{110}$$

The function $u_s(x)$ is defined to satisfy the boundary conditions of (109); hence, $\hat{u}(x, t)$ must satisfy zero boundary conditions at $x=0, L$. Both $u_s(x)$ and $\hat{u}(x, t)$ satisfy the partial differential equation, so for $u_s(x)$, we have that

$$\frac{d^2 u_s(x)}{dx^2} = 0; \quad u_s(0) = u_0; \quad u_s(L) = u_L\tag{111}$$

The solution for (111) is easily calculated as

$$u_s(x) = \left(\frac{u_L - u_0}{L} \right) x + u_0\tag{112}$$

For $\hat{u}(x, t)$, we can now solve a different IBVP given by

$$\begin{aligned}\frac{\partial \hat{u}}{\partial t} &= \frac{\partial^2 \hat{u}}{\partial x^2}; \quad 0 < x < L; \quad t > 0 \\ \hat{u} &= \hat{u}(x, t) \\ \hat{u}(0, t) &= \hat{u}(L, t) = 0 \\ \hat{u}(x, 0) &= -u_s(x)\end{aligned}\tag{113}$$

Equation (113) is now in the conventional form for a heat conduction IBVP; moreover, it is solvable by using separation of variables with Fourier series. Since its solution is performed in many classical textbooks, we simply state the result, i.e.,

$$\hat{u}(x,t) = \sum_{n=1}^{\infty} C_n \exp\left[-\left(\frac{n\pi}{L}\right)^2 t\right] \sin\left(\frac{n\pi}{L} x\right) \quad (114)$$

where

$$C_n = -\frac{2}{L} \int_0^L u_s(x) \sin\left(\frac{n\pi}{L} x\right) dx \quad (115)$$

More specifically for (112),

$$C_n = \frac{2}{n\pi} (u_L \cos(n\pi) - u_0) \quad (116)$$

As is evident from (114), the function \hat{u} clearly decays exponentially, so superposed solution $u(x,t)$ approaches $u_s(x)$, the steady state solution, as $t \rightarrow \infty$.

APPENDIX C

Azimuthally Symmetric Solution for Heat Equation with Asymmetric Heating of an Annular Section

The geometry for this problem is shown in Section 2.6, Figure 4. The associated IBVP is described as

$$\begin{aligned}\frac{\partial u}{\partial t} &= \nabla^2 u; \quad a < r < b; \quad t > 0 \\ u &= u(r, t) \\ u(a, t) &= u_a; \quad u(b, t) = u_b \\ u(r, 0) &= 0\end{aligned}\tag{117}$$

Since this problem is azimuthally symmetric, the angular coordinate does not appear in (117). As was discussed in Appendix B, we represent $u(r, t)$ as the sum of superposed solutions $u_s(r)$ and $\hat{u}(r, t)$ with

$$u(r, t) = u_s(r) + \hat{u}(r, t)\tag{118}$$

The function $u_s(r)$ is the solution to the steady state boundary value problem

$$\begin{aligned}\frac{d^2 u_s(r)}{dr^2} + \frac{1}{r} \frac{du_s(r)}{dr} &= 0 \\ u_s(a) &= u_a; \quad u_s(b) = u_b\end{aligned}\tag{119}$$

Careful integration reveals that

$$u_s(r) = C_0 \ln(r) + C_1\tag{120}$$

where

$$C_0 = \frac{u_b - u_a}{\ln(b) - \ln(a)}; \quad C_1 = \frac{u_a \ln(b) - u_b \ln(a)}{\ln(b) - \ln(a)}\tag{121}$$

The time dependent function $\hat{u}(r, t)$ is a solution to the IBVP

$$\begin{aligned}\frac{\partial \hat{u}}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \hat{u}}{\partial r} \right); \quad a < r < b; \quad t > 0 \\ \hat{u} &= \hat{u}(r, t) \\ \hat{u}(a, t) &= \hat{u}(b, t) = 0 \\ \hat{u}(r, 0) &= -u_s(r)\end{aligned}\tag{122}$$

To solve (122), we apply the separation of variables method, i.e., let

$$\hat{u}(r, t) = R(r)T(t)\tag{123}$$

By substituting (123) into the operator in (122), we obtain the separable equation

$$\frac{1}{T} \frac{dT}{dt} = \frac{1}{R} \frac{d^2 R}{dr^2} + \frac{1}{rR} \frac{dR}{dr} \quad (124)$$

Experience shows that we can select a separation constant with the value $-\lambda^2$, and we obtain two ordinary differential equations that can be solved based upon the value of λ . Hence,

$$\frac{dT}{dt} + \lambda^2 T = 0 \quad (125)$$

$$r^2 \frac{d^2 R}{dr^2} + r \frac{dR}{dr} + (\lambda r)^2 R = 0 \quad (126)$$

Equation (126), along with the attendant boundary conditions, constitutes a Sturm-Liouville problem, so the only admissible values of λ are real and positive.[12] Of course, (126) is Bessel's zeroth order differential equation with parameter λ . The solution to (125) is very simple to derive, so we state the result.

$$T(t) = A \exp(-\lambda^2 t) \quad (127)$$

where A is a constant that must be obtained from applying the initial conditions. The solution of (126) is written as

$$R(r) = B J_0(\lambda r) + C Y_0(\lambda r) \quad (128)$$

where J_0 and Y_0 are the zeroth order Bessel and Neumann functions. B and C are, at this stage of the solution, arbitrary constants pending the application of boundary and initial conditions.

Based upon the structure of (122), the boundary conditions for this problem are reduced to solely space dependent forms, i.e.,

$$R(a) = R(b) = 0 \quad (129)$$

Therefore, by applying (128), we obtain the two simultaneous equations

$$B J_0(\lambda a) + C Y_0(\lambda a) = 0 \quad (130)$$

$$B J_0(\lambda b) + C Y_0(\lambda b) = 0 \quad (131)$$

The only variable parameters in (130) and (131) are B , C and λ . Clearly, B and C cannot be zero, or the trivial solution would result. Instead, let us rewrite (130) and (131) in matrix form where

$$\begin{bmatrix} J_0(\lambda a) & Y_0(\lambda a) \\ J_0(\lambda b) & Y_0(\lambda b) \end{bmatrix} \begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (132)$$

Since this system is homogeneous and linear, the only way that (132) can be satisfied is if

$$g(\lambda) = \begin{vmatrix} J_0(\lambda a) & Y_0(\lambda a) \\ J_0(\lambda b) & Y_0(\lambda b) \end{vmatrix} = 0 \quad (133)$$

The roots of (133) consist of a countably infinite set consisting of distinct values of λ . Figure contains a plot of the determinant $g(\lambda)$. You can see several roots of this equation indicated in the plot.

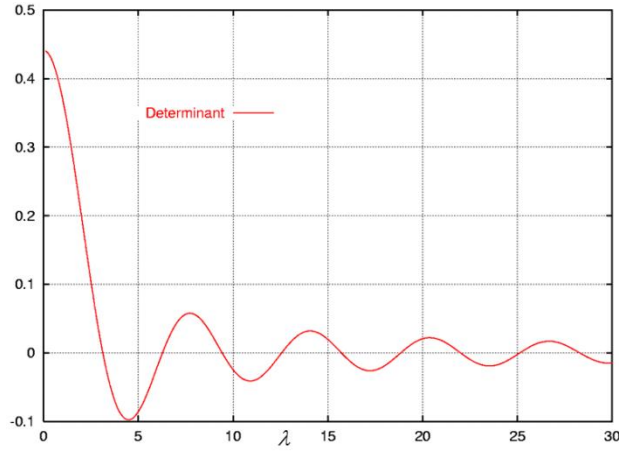


Figure 11: Boundary Condition Determinant Values for the Asymmetric heating Problem Set on an Annular Domain with Azimuthal Symmetry.

Each value of λ satisfying (133) defines an eigenfunction of (128). A more specific form of the eigenfunction (128) may be developed as follows. For the n^{th} root λ_n of (133), a rearrangement of (131) shows that

$$C = -B \frac{J_0(\lambda b)}{Y_0(\lambda b)} \quad (134)$$

The substitution of this relation into (128) renders the eigenfunction $R(\lambda_n r)$, i.e.,

$$R(\lambda_n r) = \frac{B}{J_0(\lambda_n b) Y_0^2(\lambda_n b)} \left[\frac{J_0(\lambda_n r)}{J_0(\lambda_n b)} - \frac{Y_0(\lambda_n r)}{Y_0(\lambda_n b)} \right] \quad (135)$$

Since B is just an arbitrary constant, we can write this eigenfunction as

$$R(\lambda_n r) = B \left[\frac{J_0(\lambda_n r)}{J_0(\lambda_n b)} - \frac{Y_0(\lambda_n r)}{Y_0(\lambda_n b)} \right] \quad (136)$$

Eigenfunctions (136) are now multiplied by the solution for $T(t)$ and expanded as Fourier series solving $\hat{u}(r, t)$. By using (118), (120) and (121), the general solution for (117) is

$$u(r, t) = \sum_{n=1}^{\infty} \tilde{B}_n \exp(-\lambda_n^2 t) \left[\frac{J_0(\lambda_n r)}{J_0(\lambda_n b)} - \frac{Y_0(\lambda_n r)}{Y_0(\lambda_n b)} \right] + u_s(r) \quad (137)$$

subject to the initial condition from (117)

$$u(r, 0) = 0 \quad (138)$$

Recall that Bessel functions (and hence, linear combinations, e.g., (136) of Bessel functions) are orthogonal under the inner product [12]

$$\langle R(\lambda_n), R(\lambda_m) \rangle = \int_a^b r R(\lambda_n) R(\lambda_m) dr = \delta_{mn} \int_a^b r R^2(\lambda_n) dr \quad (139)$$

Therefore, when we evaluate (137) at the initial condition (138) and form the inner integral $\langle u(r,0), R(\lambda_m) \rangle$, we can solve for the coefficients \tilde{B}_n , i.e.,

$$\tilde{B}_n = - \frac{\int_a^b r u_s(r) R(\lambda_n) dr}{\int_a^b r R^2(\lambda_n r) dr}, \quad n = 1, 2, \dots \quad (140)$$

In practice, integrals involving Bessel functions can be time consuming to evaluate, so for the results shown in this report, the integrals in (140) are numerically evaluated by using the trapezoidal rule over some 10,000 subintervals.[7] This process has been demonstrated to produce very accurate solutions.

APPENDIX D

Fully Asymmetric Heating Problem for an Annular Section

This problem is solved in order to provide a test for our finite volume heat conduction code that incorporates heating asymmetry in two separate directions. This problem is defined on a cylindrical geometry, so the unsteady solution involves complicated Bessel functions. Since the temporal performance of the computer program has been validated on two other problems, the exact solution that follows addresses a steady state temperature profile an annular section. The boundary conditions incorporate the desired asymmetry. The BVP is defined as follows.

$$\begin{aligned}
 \nabla^2 u &= 0, \quad a < r < b, \quad \theta_1 < \theta < \theta_2 \\
 u &= u(r, \theta) \\
 u(a, \theta) &= u_a(\theta); \quad u(b, \theta) = u_b(\theta) \\
 \frac{\partial u(r, \theta_1)}{\partial \theta} &= \frac{\partial u(r, \theta_2)}{\partial \theta} = 0
 \end{aligned} \tag{141}$$

The Laplace equation above can be expanded in cylindrical coordinates as

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = 0 \tag{142}$$

If we assume that $u(r, \theta) = R(r)\Theta(\theta)$, (142) is separable producing the two ordinary differential equations, i.e.,

$$r^2 \frac{d^2 R}{dr^2} + r \frac{dR}{dr} - \lambda^2 R = 0 \tag{143}$$

$$\frac{d^2 \Theta}{d\theta^2} + \lambda^2 \Theta = 0 \tag{144}$$

for a separation constant λ^2 . The process of solving (143) and (144) is complicated by the fact that zero is an admissible value for λ . As a result, each ordinary differential equation has two families of solutions, one for $\lambda = 0$, the other for $\lambda \neq 0$.

We begin by considering the radial equation (143). Suppose that $\lambda = 0$; then (143) becomes

$$r \frac{d^2 R}{dr^2} + \frac{dR}{dr} = 0 \tag{145}$$

This equation has the solution

$$R_0(r) = A \ln(r) + B \tag{146}$$

where A and B are arbitrary constants selected based on the boundary conditions. Suppose now that $\lambda \neq 0$. Equation (143) can be solved by assuming a solution of the form $R_\lambda(r) = r^\alpha$, where α is a parameter based upon λ . Substitution of r^α into (143) yields the expression

$$r^\alpha (\alpha(\alpha-1) + \alpha - \lambda^2) = 0 \quad (147)$$

For a non-trivial solution, r^α is non-zero, so

$$\alpha(\alpha-1) + \alpha - \lambda^2 = 0 \Rightarrow \alpha = \pm \lambda \quad (148)$$

Therefore, two possible solutions result for $\lambda \neq 0$, so their combination forms a solution for (143), i.e.,

$$R_\lambda(r) = Ar^\lambda + Br^{-\lambda} \quad (149)$$

Now consider the angular equation (144). The case where $\lambda = 0$ constitutes a perfectly acceptable solution, i.e.,

$$\frac{d^2\Theta}{d\theta^2} = 0 \Rightarrow \Theta_0(\theta) = C\theta + D \quad (150)$$

where C and D are arbitrary constants. On the other hand, suppose that $\lambda \neq 0$. The solution for (144) is well known for this circumstance. In fact,

$$\Theta(\theta) = C \cos(\lambda\theta) + D \sin(\lambda\theta) \quad (151)$$

where C and D are again, arbitrary constants. Using the angular solution forms given in (150) and (151), we may begin to evaluate some of the arbitrary constants. This process must be accomplished with care because arbitrary constants C and D have different values for the $\lambda = 0$ and $\lambda \neq 0$ solutions. Recall the angular boundary condition equations from (141).

$$\frac{\partial u(r, \theta_1)}{\partial \theta} = R \frac{d\Theta(\theta_1)}{d\theta} = 0; \quad R(r) \neq 0 \Rightarrow \frac{d\Theta(\theta_1)}{d\theta} = 0 \quad (152)$$

Similarly,

$$\frac{\partial u(r, \theta_2)}{\partial \theta} = R \frac{d\Theta(\theta_2)}{d\theta} = 0; \quad R(r) \neq 0 \Rightarrow \frac{d\Theta(\theta_2)}{d\theta} = 0 \quad (153)$$

By applying (152) to (150), we obtain

$$\frac{d\Theta(\theta_1)}{d\theta} = C = 0 \Rightarrow C = 0 \quad (154)$$

The application of (153) confirms this results; therefore

$$\Theta_0(\theta) = D \quad (155)$$

For the case where $\lambda \neq 0$, let us apply (152) to (151). We obtain

$$\frac{d\Theta(\theta_1)}{d\theta} = -\lambda C \sin(\lambda\theta_1) + \lambda D \cos(\lambda\theta_1) = 0 \quad (156)$$

Hence,

$$D = C \frac{\sin(\lambda \theta_1)}{\cos(\lambda \theta_1)} \quad (157)$$

Equation (151) can now be rewritten as

$$\Theta(\theta) = C[\cos(\lambda \theta)\cos(\lambda \theta_1) + \sin(\lambda \theta)\sin(\lambda \theta_1)] = C \cos \lambda(\theta - \theta_1) \quad (158)$$

In (158), the constant terms appearing in (157) have simply been rolled up into a single arbitrary constant C . Now we apply (153) to (158), and

$$\frac{d\Theta(\theta_2)}{d\theta} = -C \lambda \sin \lambda(\theta_2 - \theta_1); \quad C, \lambda \neq 0 \Rightarrow \sin \lambda(\theta_2 - \theta_1) = 0 \quad (159)$$

Equation (159) clearly allows the non-zero eigenvalues λ_n to be identified as

$$\lambda_n = \frac{n\pi}{\theta_2 - \theta_1}, \quad n = 1, 2, \dots \quad (160)$$

Finally, the angular eigenfunction for $\lambda_n \neq 0$ can be written as

$$\Theta_n(\theta) = C \cos \lambda_n(\theta - \theta_1), \quad n = 1, 2, \dots \quad (161)$$

The next step in the solution procedure involves building a general solution that satisfies the radial boundary conditions given in (141). Note that in general, the boundary values at $r = a$ and $r = b$ are non-trivial. In order to enforce non-trivial boundary values on both of these radial boundaries, we must use linearity. That is to say, $u(r, \theta)$ must be composed of two solutions $u_1(r, \theta)$ plus $u_2(r, \theta)$ such that

$$\begin{aligned} u(r, \theta) &= u_1(r, \theta) + u_2(r, \theta) \\ u_1(a, \theta) &= u_a(\theta); \quad u_1(b, \theta) = 0 \\ u_2(a, \theta) &= 0; \quad u_2(b, \theta) = u_b(\theta) \end{aligned} \quad (162)$$

If the components solutions u_1 and u_2 are structured in accordance with (162), the general solution $u(r, \theta)$ satisfies the required boundary conditions. Let us determine u_1 . For $\lambda = 0$, we apply (146) and (155) to obtain

$$u_1(b, \theta) = (A \ln(b) + B)D = 0; \quad D \neq 0 \Rightarrow B = -A \ln(b) \quad (163)$$

Hence, for $\lambda = 0$, we have that

$$u_1(r, \theta, \lambda = 0) = A_0 (\ln(r) - \ln(b)) \quad (164)$$

where A_0 represents a roll-up of arbitrary constants. The remaining part of u_1 is provided by an analysis of the case where $\lambda \neq 0$. Apply (149), (158) and (162), and we can write

$$u_1(r, \theta, \lambda \neq 0) = (A b^{\lambda_n} + B b^{-\lambda_n}) \cdot C \cos \lambda_n(\theta - \theta_1) = 0 \quad (165)$$

In general, the angular part of (165) is non-zero, so we conclude that

$$Ab^{\lambda_n} + Bb^{-\lambda_n} = 0 \Rightarrow B = -Ab^{2\lambda_n} \quad (166)$$

Therefore, with the use of another rolled up constant A_n ,

$$u_1(r, \theta, \lambda \neq 0) = A_n (r^{\lambda_n} b^{-\lambda_n} - r^{-\lambda_n} b^{\lambda_n}) \cos \lambda_n (\theta - \theta_1) \quad (167)$$

With the use of (160), (164) and (167), we can write the general solution for u_1 , i.e.,

$$u_1(r, \theta) = A_0 (\ln(r) - \ln(b)) + \sum_{n=1}^{\infty} A_n (r^{\lambda_n} b^{-\lambda_n} - r^{-\lambda_n} b^{\lambda_n}) \cos \lambda_n (\theta - \theta_1) \quad (168)$$

Expansion coefficients A_n must be determined based upon the boundary conditions enforced at $r = a$. For this purpose, we exploit the orthogonality of eigenfunctions. In particular, it can be shown that the set

$$\Xi = \{1, \cos \lambda_n (\theta - \theta_1), n = 1, 2, \dots\} \quad (169)$$

is orthogonal under the inner product defined by

$$\langle \Xi_n, \Xi_m \rangle = \int_{\theta_1}^{\theta_2} \Xi_n(\theta) \Xi_m(\theta) d\theta \quad (170)$$

By computing inner products carefully, it can be shown that

$$A_0 = \frac{1}{(\ln(a) - \ln(b))(\theta_2 - \theta_1)} \int_{\theta_1}^{\theta_2} u_a(\theta) d\theta \quad (171)$$

$$A_n = \frac{2}{(a^{\lambda_n} b^{-\lambda_n} - a^{-\lambda_n} b^{\lambda_n})(\theta_2 - \theta_1)} \int_{\theta_1}^{\theta_2} u_a(\theta) \cos n\pi \left(\frac{\theta - \theta_1}{\theta_2 - \theta_1} \right) d\theta \quad (172)$$

Equations (170) through (172) form a general solution for u_1 that satisfies the radial boundary conditions $u_1(a, \theta) = u_a(\theta)$ and $u_1(b, \theta) = 0$.

To complete the general solution for (141), we also must construct the component solution $u_2(r, \theta)$ that conforms to the boundary conditions given in (162). The same procedure used to obtain u_1 is applied in this case. Since the procedure is repetitive, only key results encountered in the course of the derivation are presented. By noting that $u_2(a, \theta) = 0$, the components of u_2 may be obtained as

$$u_2(r, \theta, \lambda = 0) = A_0 (\ln(r) - \ln(a)) \quad (173)$$

$$u_2(r, \theta, \lambda \neq 0) = \sum_{n=1}^{\infty} A_n (a^{-\lambda_n} r^{\lambda_n} - a^{\lambda_n} r^{-\lambda_n}) \cos \lambda_n (\theta - \theta_1) \quad (174)$$

Orthogonality, (169) and (170), can be used in conjunction with the boundary condition $u_2(b, \theta) = u_b(\theta)$ to determine coefficients A_n . With some careful calculation, we obtain

$$A_0 = \frac{1}{(\ln(b) - \ln(a))(\theta_2 - \theta_1)} \int_{\theta_1}^{\theta_2} u_b(\theta) d\theta \quad (175)$$

$$A_n = \frac{2}{(a^{-\lambda_n} b^{\lambda_n} - a^{\lambda_n} b^{-\lambda_n})(\theta_2 - \theta_1)} \int_{\theta_1}^{\theta_2} u_b(\theta) \cos n\pi \left(\frac{\theta - \theta_1}{\theta_2 - \theta_1} \right) d\theta \quad (176)$$

With the equations presented above, the general solution for (141) is easily calculated. A computer can be used to calculate $u(r, \theta)$ to any desired level of accuracy.

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

ADI	Alternating Direction Implicit
BVP	Boundary Value Problem
CEM	Computational Electromagnetics
CFD	Computational Fluid Dynamics
IBVP	Initial – Boundary Value Problem

DISTRIBUTION LIST
AFRL-RW-EG-TR-2010-049

DEFENSE TECHNICAL INFORMATION CENTER - 1 Electronic Copy (1 File & 1 Format)
ATTN: DTIC-OCA (ACQUISITION)
8725 JOHN J. KINGMAN ROAD, SUITE 0944
FT. BELVOIR VA 22060-6218

AFRL/RWOC-1 (STINFO Office) - 1 Hard (Color) Copy
AFRL/RW CA-N - STINFO Officer Provides Notice of Publication

IIT RESEARCH INSTITUTE/GACIAC - 1 Hard (Color) Copy
10 WEST 35TH STREET
CHICAGO IL 60616-3799