SANDIA REPORT

SAND2005-1485
Unlimited Release
Printed March 2005

Visualization of Instrumental Verification Information Details (VIVID): Code Development, Description, and Usage

Donald L. Potter, Frederick G. Blottner, Amalia R. Black, Christopher J. Roy, and Bruce L. Bainbridge

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND 2005-1485
Unlimited Release
Printed March 2005

Visualization of Instrumental Verification Information Details (VIVID): Code Development, Description, and Usage

D. L. Potter, F. G. Blottner, C. J. Roy
Aerosciences and Compressible Fluid Mechanics

A. R. Black
Validation and Uncertainty Quantification

B. L. Bainbridge
Thermal and Reactive Processes

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0835

Abstract

The formulation, implementation and usage of a numerical solution verification code is described. This code uses the Richardson extrapolation procedure to estimate the order of accuracy and error of a computational program solution. It evaluates multiple solutions performed in numerical grid convergence studies to verify a numerical algorithm implementation. Analyses are performed on both structured and unstructured grid codes. Finite volume and finite element discretization programs are examined. Two and three-dimensional solutions are evaluated. Steady state and transient solution analysis capabilities are present in the verification code. Multiple input data bases are accepted. Benchmark options are included to allow for minimal solution validation capability as well as verification.

Acknowledgment

The authors thank Ben Blackwell and Bill Oberkampf for their review of the fully assembled draft of this report. The first author thanks his co-authors for their direct contributions for their sections. In addition, they are thanked for their comments and review of all other sections.

Contents

Figures 7
Tables 9
Nomenclature 10
1. Introduction 13
2. Program Structure 15
2.1 Input Data Extraction 16
2.2 Reference Data Production 16
2.3 Order of Accuracy and Error Analysis 17
3. Richardson Extrapolation and Order of Accuracy 18
3.1 One Dimensional Extrapolation 18
3.2 Two Dimensional Extrapolation 19
3.2.1 Simultaneous Equal Refinement in Both Coordinates 19
3.2.2 Selective Directional Refinement 20
3.2.3 Single Directional Refinement 20
3.3 Order of Accuracy 22
4. Code Development and Functionality 25
4.1 Main Program: VIVID 25
4.1.1 Driver Routine Structure 25
4.1.2 Command Line Arguments 25
4.1.3 Interactive Interrogation and Constitutive Program Execution 26
4.2 EXTRACT Subprogram 26
4.3 ORDER Subprogram 26
4.4 BENCHMARK Subprogram 27
4.4.1 Flat Duct Incompressible Flow 28
4.4.2 Two-Dimensional Heat Conduction 29
5. Sample Analyses & Results 31
5.1 MP-Salsa Cavity Flow Unstructured Grid Convergence Analysis 32
5.2 CFD-ACE Flat Duct Structured Grid Convergence Analysis 43
5.3 SACCARA Compressible Structured Grid Convergence Analysis 48
5.4 Premo Compressible Unstructured Grid Convergence Analysis 51
5.5 Fuego Flat Duct Unstructured Grid Convergence Analysis 53
5.6 Calore Conduction Unstructured Grid Convergence Analysis 58
5.7 FCV-FAM Conduction Structured Grid Convergence Analysis 62
5.8 FCV Transient Unstructured Grid Convergence Analysis 65
5.9 Calore 3-Dimensional Conduction Unstructured Grid Convergence Analysis 67
6. Conclusions and Future Work 70
References 72
Appendix A. Program Structure Details 74
A.1 Input Data Extraction 74
A.2 Reference Data Production 74
A.3 Order of Accuracy and Error Analysis 75

Appendix B. Code Development & Functionality 77
B.1 Main Program: VIVID 77
B.1.1 Driver Routine Structure 77
B.1.2 Command Line Arguments 78
B.1.3 Interactive Interrogation & Constitutive Program Execution 78
B.2 EXTRACT Subprogram 79
B.3 ORDER Subprogram 81
B.4 BENCHMARK Subprogram 83
B.4.1 Flat Duct Incompressible Flow 83
B.4.2 Two-Dimensional Heat Conduction 84
B.5 Interactive Screen Session for VIVID 84
Appendix C. Sample Control Input Files 93
Appendix D. Dimension and Conduction Files 106
Appendix E. FCV-FAM Code Description 107
Distribution 109

Figures

Figure 2.1 Schematic of VIVID Order Evaluation....15
Figure 4.1 Two-Dimensional Conduction Boundary Conditions....30
Figure 5.1.1 MP-Salsa Horizontal Velocity Contours....33
Figure 5.1.2 Horizontal Velocity Solution Order....34
Figure 5.1.3 Horizontal Velocity Solution Order With revised Scale....34
Figure 5.1.4 Horizontal Velocity Solution Order With Crude Mesh....35
Figure 5.1.5 Error Behavior vs. Mesh Refinement at Three Points....37
Figure 5.1.6 Absolute % Error of Horizontal Velocity for Fine Mesh....39
Figure 5.1.7 Percent Error of Horizontal Velocity for Fine Mesh....39
Figure 5.1.8 Percent Error of Horizontal Velocity for Coarse Mesh....40
Figure 5.1.9 Discretization Error of Horizontal Velocity for Fine Mesh....40
Figure 5.1.10 Discretization Error of Horizontal Velocity for Coarse Mesh....41
Figure 5.1.11 Location of Zero Percent Error of Horizontal Velocity for Fine Mesh....41
Figure 5.1.12 FCV/Fuego Temperature Error vs. Extrapolated....42
Figure 5.2.1 CFD-ACE Flat Duct Axial Velocity Contours....43
Figure 5.2.2 CFD-ACE Structured Grid Velocity Order....44
Figure 5.2.3 CFD-ACE Structured Grid Pressure Order....46
Figure 5.2.4 CFD-ACE Fine Grid Axial Velocity Error....46
Figure 5.2.5 CFD-ACE Medium Grid Axial Velocity Error....47
Figure 5.3.1 SACCARA Structured Grid Mach Number Contours....48
Figure 5.3.2 SACCARA Structured Grid u-Velocity Error Norms....48
Figure 5.3.3 SACCARA Structured Grid u-Velocity Order....49
Figure 5.3.4 SACCARA Structured Grid u-Velocity Error....50
Figure 5.4.1 Premo Unstructured Grid Density Contours....51
Figure 5.4.2 Premo Unstructured Grid Density Error Norms....51
Figure 5.4.3 Premo Unstructured Grid Density Order....52
Figure 5.4.4 Premo Unstructured Grid Density Error....52
Figure 5.5.1 Fuego Unstructured Grid Velocity Contours....53
Figure 5.5.2 Fuego Unstructured Grid Axial Velocity Order....54
Figure 5.5.3 Fuego Unstructured Grid Axial Velocity Error....55
Figure 5.5.4 Axial Velocity Norms for Coarse, Medium and Fine Mesh Solutions....55
Figure 5.5.5 Fuego Unstructured Grid Pressure Order....56
Figure 5.5.6 Fuego Unstructured Grid Pressure Error....56
Figure 5.5.7 Pressure Norms for Coarse, Medium and Fine Mesh Solutions....57
Figure 5.6.1 Calore Unstructured Grid Temperature Contours....59
Figure 5.6.2 Calore Temperature Order from Extrapolated....59
Figure 5.6.3 Calore Temperature Error vs. Extrapolated....60
Figure 5.6.4 Calore Temperature Error vs. Analytical....60
Figure 5.6.5 Calore Temperature Order from Analytical....61
Figure 5.7.1 FCV-FAM Structured Grid Temperature Contours....62
Figure 5.7.2 FCV-FAM Temperature Order from Extrapolated....63
Figure 5.7.3 FCV-FAM Temperature Error vs. Extrapolated....63
Figure 5.7.4 FCV-FAM Temperature Error vs. Analytical....64

Figure 5.7.5 FCV-FAM Temperature Order from Analytical 64
Figure 5.8.1 FCV Transient Velocity Contours 65
Figure 5.8.2 FCV Transient Velocity Order 65
Figure 5.8.3 FCV Transient Velocity Error vs. Extrapolated 66
Figure 5.9.1 Calore 3-D Example Boundary Conditions 67
Figure 5.9.2 Calore 3-D Unstructured Grid Temperature Solution 67
Figure 5.9.3 Calore 3-D Unstructured Temperature Order 68
Figure 5.9.4 Calore 3-D Cross Section Temperature Order 68
Figure 5.9.5 Calore 3-D Unstructured Temperature Error 69
Figure 5.9.6 Calore 3-D Cross Section Temperature Error 69

Tables

Table 5.1 Summary of VIVID Sample Problems 31
Table 5.2 Solution Scheme Order Behavior at Three Locations 36
Table 5.3 Solution of v_x at the Mesh Point $N = 216$ 36
Table 5.4 Solution of v_x at the Mesh Point $N = 217$ 36
Table 5.5 Solution of v_x at the Mesh Point $N = 218$ 37

Nomenclature

b	Duct half spacing in the flat duct analytical formulation
D_h	Hydraulic diameter
dp/dx	Pressure gradient in the axial direction in the flat duct analytical formulation
E	Numerical solution discretization error
F_e	Exact solution of governing equation in Richardson extrapolation formulation
f_n	Computational solution on mesh n in Richardson extrapolation formulation
H	Cross section height in 2-D analytical heat conduction formulation
h	Convective heat transfer coefficient
k	Thermal conductivity
L	Cross section width in 2-D analytical heat conduction formulation
L_{hy}	Hydrodynamic entrance length in analytical flat duct formulation
p	Order of accuracy
$\dot{q}$	Heat flux at a surface in 2-D analytical heat conduction formulation
q'''	Uniform internal heat generation in 2-D analytical heat conduction formulation
Re	Reynolds number
r	Grid refinement ratio = δ_{n+1}/δ_n
U	Dimensionless velocity in the flat duct analytical formulation
T	Temperature in 2-D analytical heat conduction formulation
T_o	Constant temperature boundary condition in 2-D analytical heat conduction formulation
T_∞	Convective sink temperature in 2-D analytical heat conduction formulation
t	Time
u	Local axial velocity in the flat duct analytical formulation
u_m	Mean inlet axial velocity in the flat duct analytical formulation
v	Local vertical velocity in the flat duct analytical formulation
x	Axial distance in two-dimensional problem
x^+	Dimensionless axial distance in the flat duct analytical formulation
y	Vertical distance from duct centerline in flat duct analytical formulation

Greek Letters

α	Thermal diffusivity in 2-D analytical heat conduction formulation
α, β	Spatial coefficient in discretization error expansion for Richardson extrapolation
Δ	Grid spacing in a numerical solution
δ	Grid spacing in a numerical solution

Superscripts

p	Order of accuracy

Subscripts

h	Hydraulic diameter variable subscript
hy	Hydrodynamic entrance length variable subscript
i	Inviscid region in flat duct analytical formulation
n	Mesh number as refinement is performed (n =1 is the finest mesh)
v	Viscous region in flat duct analytical formulation
x	X coordinate direction in mesh
y	Y coordinate direction in mesh
1, 2, 3	Generally refers to successively refined grid solutions

(Page Left Blank)

1. Introduction

Design procedures for advanced weapon systems have changed dramatically over the years. A fundamental part of that change has been the move away from an experimental/test based design to one that relies more heavily on computational procedures. In part, this has been driven by the desire to reduce costs incurred during multiple full scale application tests of a prospective system. As the shift has occurred, greater demands have been placed on the computational tools and the people using them. Simulation models must have more fidelity and be applied over a wider range of conditions. As such, it is imperative that the software tools developed and employed by technical personnel have undergone rigorous verification and validation procedures. Paraphrasing from Blottner[1], *verification* means *solving the equations right*, while *validation* means *solving the right equations*. The current work is concerned with the former. Specifically, verification is the process by which software developers or analysts ensure that a code is *accurately solving the governing equations* of interest. A significant effort is being made at Sandia by the SIERRA code development personnel as part of the Advanced Strategic Computing Initiative (ASCI) Applications program to develop a formal software verification procedure.[2] As a part of this effort, development of the Visualization of Instrumental Verification Information Details (VIVID) code has been pursued. This report documents progress in the development, usage, and testing of the VIVID software.

VIVID is a Fortran 77 code which runs on the SRN LAN employed by the 9100 Engineering Sciences Center (Unix platforms only - such as sass2889). It is not currently operational on personal linux workstations. VIVID consists of a number of subprograms that extract dependent variable information from numerical solutions, estimate solution error and order of accuracy, and execute benchmark routines for comparison to the numerical results. The subprograms by name are EXTRACT, ORDER, and BENCHMARK. The subprograms executed by VIVID are referred to from this point as constitutive programs. Each constitutive program was written specifically for the ASCI project to further the development of a formal verification procedure.

Order of accuracy relates the error reduction characteristics in a numerical solution for a specified amount of grid refinement. For example, if the separation distance between grid points is reduced by a factor of two, the error in a second order numerical scheme would be reduced by a factor of four. Following is a brief description of the solution procedure to determine the numerical order of accuracy of a computational program. Initially, the domain geometry, numerical solutions, and user defined run options are supplied to the program. Either two or three numerical solutions completed for the same problem with different grid refinements may be provided as input for verification analysis. Per user instructions, dependent variables over a segment of the solution are extracted for evaluation. At the users discretion, any or all parts of the solution may be examined for any or all dependent variables. Output

extracted from the numerical solutions is then used as input in an order of accuracy analysis. The input data is sorted to match corresponding grid points between the different grid refinement solutions. Assuming an analytical, numerical, or data base reference is available, a benchmark dependent variable value is determined at each solution point for comparison to the numerical method under consideration. Richardson extrapolation[3] is applied at the coarse grid point solution coordinates to determine order of accuracy and an estimate of the numerical solution error over the analysis domain defined by the user. At completion of an evaluation, an output file is produced for visualization.

Verification work of this nature at Sandia has its roots in theory and analyses forwarded by Blottner, Oberkampf, Walker[4,5,6] and others. At the time of their pioneering work, they did not have the luxury of the computing resources available today. Evaluations were necessarily limited to global variables for relatively simple geometries. With the implementation of the current code development approach, VIVID provides a powerful tool for detailed local analyses of numerical solutions on a domain scale never before achieved.

2. Program Structure

VIVID is a driver program that coordinates the subprograms to evaluate the order of accuracy and estimate the error of a numerical solution. It has been primarily developed on solutions generated by computational fluid dynamics programs, but is class independent. For example, it may be applied to solid mechanics problems. A schematic of the program, as it functions in an order of accuracy evaluation, is presented by Figure 2.1. In practice, the independent subprograms ORDER, EXTRACT,

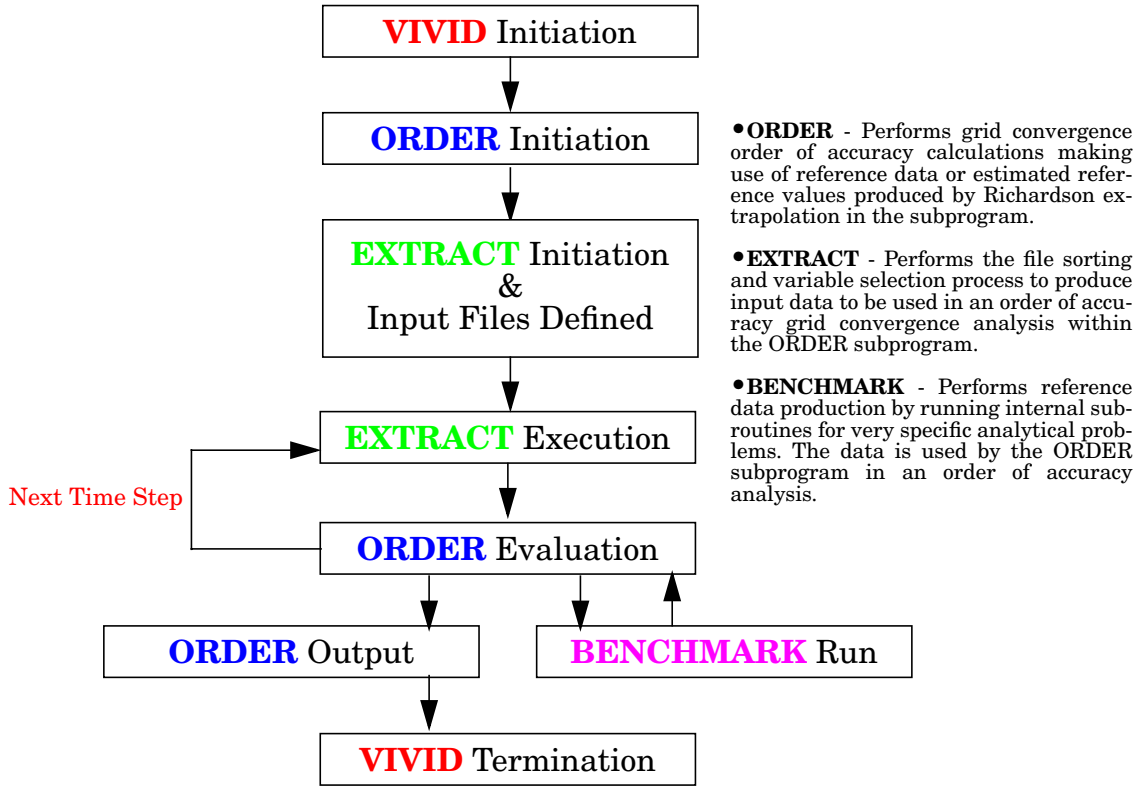

Figure 2.1 Schematic of VIVID Order Evaluation

and BENCHMARK are incorporated as subroutines and executed as options of the main program. Program dimensions are expediently handled by an include file. It should be noted that both EXTRACT and BENCHMARK can be run in an independent mode in VIVID separate from ORDER. Subprograms employed by VIVID are briefly introduced here in the subsections and discussed in detail in Appendix A. Program discussions are of a general nature with most of the mathematical details left to the references.

Following is a brief description of the VIVID procedure assuming a numerical order of accuracy evaluation is being pursued. More detailed description of the subprograms comprising VIVID are presented in Section 4. Initially, the ORDER program is executed by the driver routine to establish basic run parameters. ORDER

then initiates a run of the EXTRACT subprogram. Depending on the evaluation, two or three numerical solutions completed for the same problem with different grid refinements are supplied to EXTRACT as input. At the discretion of the user, any or all parts of the solution may be examined for any or all dependent variables. Output files are generated by EXTRACT at the completion of its run for use as input to ORDER as control is returned to this program. At this point, further run parameters are defined including the options controlling execution of BENCHMARK. ORDER sorts the input data and matches corresponding nodal points between the different solutions. Assuming an analytical, numerical, or data base reference is available, the appropriate BENCHMARK option is invoked from ORDER at each solution point for comparison to the numerical method under consideration. Richardson extrapolation is applied at the coarse grid point solution coordinates by ORDER to determine order of accuracy and an estimate of the numerical solution error over the analysis domain defined by the user. At completion of the ORDER evaluation, an output file in TECPLOT[7] format is produced for visualization.

2.1 Input Data Extraction

Input data parsing for VIVID is handled by a program written specifically for the ASCI/Fuego verification process. Numerical solution data to be evaluated by the VIVID processor is initially input to the **EXTRACT** subprogram. Both EXODUS and TECPLOT data base formats can be used for the multiple input solutions required in a grid convergence study. The program name comes directly from its functional purpose. It literally sorts through the input files and ***extracts*** the parts of each solution defined by the user to be evaluated in future analyses. Independent and dependant variables are defined, collected, and output to files for subsequent use by the ORDER subprogram. This program can be run in an independent mode or as part of an order of accuracy analysis.

2.2 Reference Data Production

As with the input data extraction process, reference baseline data generated by VIVID is produced by a program written specifically for the ASCI/Fuego verification process. Reference solutions used for comparative analysis purposes by the VIVID processor are created by the **BENCHMARK** subprogram. The routine name comes from the procedure's functional purpose. It produces ***benchmark*** solution information to be used in grid convergence analyses performed by the user. The reference solutions are produced by subroutines that solve very specific simple analytical problems. Just as with the EXTRACT procedure, this program can be run in an independent mode or as part of an order of accuracy analysis.

2.3 Order of Accuracy and Error Analyses

The analysis functions, including order of accuracy and error estimation, conducted on the numerical solutions input to VIVID are performed by a third program written specifically for the ASCI/Fuego verification process. Numerical solution data evaluated by the VIVID processor is analyzed by the **ORDER** subprogram. Like the two previous subprograms discussed in this section, the current procedure name also comes directly from its functional purpose. The routine evaluates the observed ***order*** of accuracy for the numerical solution over the region defined by the user. As stated in the first section, order of accuracy relates the amount of error reduction in a numerical solution for a specified amount of grid refinement. Stated mathematically this means that the ratio of the error between two solutions is equal to the grid refinement factor raised to a power corresponding to the order of accuracy of the numerical method (see Eq. 2.1).[8] In Equation 2.1, E, r, and p are the solution

$$E_2/E_1 = r^p \qquad \textit{(r=mesh 1 element size/mesh 2 element size)} \qquad \textbf{(2.1)}$$

discretization error at the grid points, grid refinement ratio, and order of accuracy, respectively. Subscript one represents the refined grid solution. The refinement ratio r is the ratio of the physical size of the elements in the same location in solution one versus solution two. To be strictly applicable, this equation makes the assumption that the refined grid contains nodal points at all of the same locations as the coarse mesh. Order of accuracy analyses performed by this subroutine can be done in concert with the VIVID resident EXTRACT and BENCHMARK programs or independent of them if the appropriate columnar files are directly generated by a numerical method. Analyses can be performed with input of either two or three numerical solution files.

3. Richardson Extrapolation & Order of Accuracy

For VIVID to achieve its primary purpose, numerical solution verification, it must be able to estimate the error within a series of refined grid solutions in a grid convergence study. This implies that it must estimate the exact solution based on the series of numerical solutions and from that determine the order of accuracy and error of the numerical method. Richardson extrapolation, as noted in Sections 1 and 2.3, can be used to estimate the exact solution based on numerical results. Following is a series of mathematical developments demonstrating the extrapolation concept.

3.1 One-Dimensional Extrapolation (p assumed known)

For a steady state one-dimensional problem, the exact solution can be estimated from the numerical solution (i.e. f_n) and the discretization error by Equation 3.1.[6]

$$F_e = f_n + \alpha\Delta^p + higher\text{–}order\text{–}terms \qquad \textbf{(3.1)}$$

Nominally, there are three unknowns in this equation if you ignore the higher order terms. They are F_e (the estimated exact solution), α (the error coefficient), and p (the order of accuracy). Three equations (i.e. three numerical solutions with successively refined grids) would be required to discern all unknowns in a rigorous fashion. However, by assuming that p is known and is equal to the formal order of accuracy, the exact solution can be estimated from only two numerical solutions. The numerical solution (f_n) in Equation 3.1 was obtained from a solution algorithm of formal order p with a grid spacing of Δ. By substituting two different grid size values (δ and $r\delta$) into Equation 3.1, while ignoring the higher order terms, the coefficient α is eliminated and an expression for the estimated exact solution to the problem can be derived. A typical value for r (the grid spacing ratio) is two. It is assumed to be a constant within and between grids in a convergence analysis. Equations 3.2 and 3.3 result when the grid spacing values are inserted into 3.1. Subtracting 3.2 from 3.3

$$F_e = f_1 + \alpha\delta^p \quad \text{(fine grid solution)} \qquad \textbf{(3.2)}$$

$$F_e = f_2 + \alpha(r\delta)^p \quad \text{(coarse grid solution)} \qquad \textbf{(3.3)}$$

and solving for α produces Equation 3.4. Substituting Equation 3.4 for the error co-

$$\alpha = (f_1 - f_2)/(r^p - 1)\delta^p \qquad \textbf{(3.4)}$$

efficient in 3.2 produces an extrapolated estimate for the exact solution to the differential equation approximated by the original difference equation (see Eq. 3.5). The

$$F_e = f_1 + (f_1 - f_2)/(r^p - 1) \qquad \textbf{(3.5)}$$

grid spacing refinement ratio ($r=\delta_2/\delta_1$), used in the derivation, is arbitrary (greater than 1) and is not confined to an integer value. An estimate of the error for the ***fine grid*** solution is obtained in Equation 3.6 by subtracting the solution (f_1) from the estimated exact value (F_e).

$$E_1 = (f_1 - f_2)/(r^p - 1) \qquad \textbf{(3.6)}$$

3.2 Two-Dimensional Extrapolation

The one-dimensional extrapolation analysis for steady state problems is readily extended to two-dimensional problems. Three types of extrapolation will be examined. The specific extrapolation process is dependent on the nature of the grid refinements used in the multiple numeric solutions.

3.2.1 Simultaneous Equal Refinement in Both Coordinates (p assumed)

Two numerical solutions are required for the following approach. Order of accuracy (p) is assumed to be known to reduce the number of unknowns in the estimated exact value equations to two. It further assumes p is the same in both the x and y directions. Grid spacing for the fine mesh solution is δ_x by δ_y. Spacing for the coarse mesh solution is $r\delta_x$ by $r\delta_y$ where r is a number greater than 1 (typically 2). By ignoring the higher order terms, an estimate of the two-dimensional exact solution is expressed by Equation 3.7. A derivation procedure similar to that of Section 3.1 is

$$F_e = f_1 + \alpha\delta_x^p + \beta\delta_y^p \qquad \text{(fine grid solution)} \qquad \textbf{(3.7)}$$

followed. Initially, a second independent equation for the exact solution estimate is obtained by substituting in the refined grid separation variables that are multiples of the original solution grid spacing into 3.7 (see Eq. 3.8). Simultaneous solution of

$$F_e = f_2 + \alpha(r\delta_x)^p + \beta(r\delta_y)^p \qquad \text{(coarse grid solution)} \qquad \textbf{(3.8)}$$

these two relations leads to an expression relating the error estimate terms to the numerical solutions. Equation 3.9 shows the combined result of the x and y error terms. When 3.9 is substituted back into 3.7, and the fine mesh numerical solution is

$$\alpha\delta_x^p + \beta\delta_y^p = (f_1 - f_2)/(r^p - 1) \qquad \textbf{(3.9)}$$

subtracted from the result, the error estimate for the fine grid is expressed in Equation 3.10. It should be noted that this is exactly equal to the right side of Equation

$$E_1 = (f_1 - f_2)/(r^p - 1) \qquad \textbf{(3.10)}$$

3.9. It should be further noted that the two-dimensional error estimate is identical in form to the one-dimensional solution shown in Equation 3.6. From this, it can be implied that Equation 3.5 then also represents the estimated exact solution for a grid convergence study of this variety.

$$F_e = f_1 + (f_1 - f_2)/(r^p - 1) \tag{3.5}$$

3.2.2 Selective Directional Refinement (p assumed)

Although the error in the fine grid solution is estimated by Equation 3.10, no information can be discerned about the specific error contributions resulting from grid refinements separately in the x and y directions. Pursuit of the individual error contributions from the separate directions requires solution of the fourth unknown variable introduced in Equation 3.7. This unknown is β (the error coefficient in the y direction). When combined with the three unknown variables defined at the beginning of Section 3.1, a system of four equations would be required for complete rigorous solution. However, by repeating the assumption of p as the formal order of accuracy, coordinate dependent error information is available with a three solution approach. With selective choice of the three grid refinements, the x and y solution error terms ($\alpha\delta_x^p$ and $\beta\delta_y^p$ respectively) can be evaluated individually. Equation 3.7 is still the starting point. Three grid refinement solutions of the following nature are required: 1 -> δ_x by δ_y (fine mesh solution), 2 -> $r\delta_x$ by δ_y, and 3 -> δ_x by $r\delta_y$. Substituting the refinements for one and two into 3.7 and solving for the x error term yields 3.11. A similar procedure using refinements one and three results in isolation

$$\alpha\delta_x^p = (f_1 - f_2)/(r^p - 1) \tag{3.11}$$

of the y error term (see Eq. 3.12). By substituting the results of the x and y error

$$\beta\delta_y^p = (f_1 - f_3)/(r^p - 1) \tag{3.12}$$

terms into 3.7, the estimated exact solution can be derived making use of all three solutions (see Eq. 3.13). The advantage of this development is that grid refinements

$$F_e = f_1 + (f_1 - f_2)/(r^p - 1) + (f_1 - f_3)/(r^p - 1) \tag{3.13}$$

can now be undertaken in selective directions determining their relative importance.

3.2.3 Single Directional Refinement

Single directional refinement for a two-dimensional grid is an extension of the one-dimensional approach. Three solutions are employed so that p can be solved for directly rather than assuming it equal to the solution algorithm formal order of accuracy. As noted in the previous section, four unknowns exist in the equation that estimates the exact value for a two-dimensional grid convergence error analysis. From Equation 3.7, repeated below for convenience, the four unknown variables are the es-

timated exact value, order of accuracy, and the two directional error coefficients.

$$F_e = f_1 + \alpha\delta_x^p + \beta\delta_y^p \qquad \text{(fine grid solution)} \qquad \textbf{(3.7)}$$

However, since only one of the grid spacing coordinates is being refined, the error coefficient for the transverse direction is conveniently eliminated in the same step as F_e such that the order may be solved for rather than assumed. The grids are consecutively refined between solutions by the same grid spacing constant. For the initial development, the grid in a single direction (the x coordinate in this illustration) will be refined. Grid refinements of the following form are required: 1 -> δ_x by δ_y, 2 -> $r\delta_x$ by δ_y, and 3 -> $r^2\delta_x$ by δ_y . As to be expected, substituting grid refinements one and two into Equation 3.7 results in the same expression for the x error term as in 3.11. Following this, the substitution of refinements one and three into 3.7 produces

$$\alpha\delta_x^p = (f_1 - f_2)/(r^p - 1) \qquad \textbf{(3.11)}$$

a second expression for the x error term (see Eq. 3.14). As was alluded to above and will be expanded upon in the Section 3.3, combination of Equations 3.11 and 3.14 allows for direct solution of the order of accuracy variable. Note that Equation 3.14 dif-

$$\alpha\delta_x^p = (f_1 - f_3)/(r^{2p} - 1) \qquad \textbf{(3.14)}$$

fers only by the power on r when compared to the y error term (Eq. 3.12) in the previous development. The estimated exact solution can be given by 3.15a. Alter-

$$F_e = f_1 + [(f_1 - f_2)/(r^p - 1)] + \beta\delta_y^p \qquad \textbf{(3.15a)}$$

nately, Equation 3.15b makes use of the second estimate for the x error term (Equation 3.14) to estimate the exact value. A development of this type, making use of

$$F_e = f_1 + [(f_1 - f_3)/(r^{2p} - 1)] + \beta\delta_y^p \qquad \textbf{(3.15b)}$$

refinement in the y direction, would lead to similar results with the y coefficient error term replaced in the equations above by the x error coefficient quantity (i.e. $\alpha\delta_x^p$). It should be noted that this type of refinement analysis while able to discern order, is unable to yield a complete error estimate of the solution or estimate the exact value due to a lack of error information in the transverse direction. To perform these functions, it would be necessary to either assume p is equal to the formal order of accuracy or assume that the errors in the transverse direction were negligible. If it were assumed that the grid refinement in the y direction was already sufficient to produce higher order error (approximately zero) in the transverse direction, Equation 3.15a reduces to the one-dimensional solution (i.e. Equation 3.5).

Assuming negligible errors in the non-refined grid direction will lead to erroneous results if the existing grid is insufficient. To avoid this problem, a simple safeguard is to refine equally in both directions. Making use of the single directional refinement work allows direct extension of the simultaneous equal refinement method of Section 3.2.1 to encompass a three solution approach. Equations 3.7 through 3.10 apply directly in this case. The grids for solutions one and two are as follows: 1 -> δ_x by δ_y, 2 -> $r\delta_x$ by $r\delta_y$. The combined x and y error term for this case is a repeat of Equation 3.9. If the third grid is assumed to be $r^2\delta_x$ by $r^2\delta_y$, substitution of this

$$\alpha\delta_x^p + \beta\delta_y^p = (f_1 - f_2)/(r^p - 1) \qquad \textbf{(3.9)}$$

into 3.7 and subsequent simultaneous solution of the resulting equations produces a second relation for the combined error term (see Eq. 3.16). The right side of this error

$$\alpha\delta_x^p + \beta\delta_y^p = (f_1 - f_3)/(r^{2p} - 1) \qquad \textbf{(3.16)}$$

equation is of identical form to that in Equation 3.14. Examination of the two error terms produced here show that the form of the estimates is identical to those produced for the single directional refinement analysis of the previous paragraph. However, in this case, the right hand sides represent the complete error estimate rather than for a single coordinate direction. Equations 3.5 and 3.15c shown below are both capable of estimating the exact solution for the dual direction refinement approach just presented. VIVID uses the form shown in 3.5 for grid convergence studies of the

$$F_e = f_1 + (f_1 - f_2)/(r^p - 1) \qquad \text{(fine grid solution)} \qquad \textbf{(3.5)}$$

$$F_e = f_1 + (f_1 - f_3)/(r^{2p} - 1) \qquad \textbf{(3.15c)}$$

variety just discussed as well as that of Sections 3.2.1. It should be noted that this form is not appropriate for estimating the exact solution with single direction refinement of the variety discussed at the beginning of this section (3.2.3) unless the error in the transverse direction has already been deemed to be essentially zero.

3.3 Order of Accuracy

To this point, the developments have focussed on estimating the error of a series of numerical simulations with the critical assumption that the order of accuracy is known. Order of accuracy will now be addressed. Roache[8] Equation 2.1 of Section 2.3 states the physical significance of the mathematical concept of solution order of accuracy where subscript one represents the finer grid solution. As stated previously

in Equation 2.1 (which is repeated here for the convenience of the reader), the ratio

$$E_2/E_1 = r^p$$

of the error between two solutions is equal to the grid refinement factor raised to a power corresponding to the order of accuracy of the numerical method. This implies that the order of accuracy could be determined directly from estimated error of a group of numerical solutions. However, it requires that the number of unknowns in the group of solutions is equal to the number of independent equations available in the grid convergence study. When the three unknowns are estimated exact value, error, and order, it is possible to solve for the order directly if there are three successively refined grid solutions used as input.

The following derivation applies to grid convergence studies that have undergone equal grid refinement in all coordinate directions. The results can be used to evaluate the order of accuracy of two or three-dimensional simulations. It does not apply to the selective directional grid refinements of the variety discussed in Section 3.2.2. Refer to equations 3.11 and 3.14 in Section 3.2.3. Two independent expressions are given for the error term described in that derivation. A similar thing occurs in Equations 3.9 and 3.16. Either of these two situations gives rise to the opportunity to develop an expression for order of accuracy without having to first calculate the actual error terms. The derivation is begun in Equation 3.17 by equating the right hand sides of Equations 3.11 and 3.14. Collecting like terms results in Equation

$$(f_1 - f_2)/(r^p - 1) = (f_1 - f_3)/(r^{2p} - 1) \qquad \textbf{(3.17)}$$

3.18. The numerator on the left hand side is the difference between two perfect

$$(r^{2p} - 1)/(r^p - 1) = (f_1 - f_3)/(f_1 - f_2) \qquad \textbf{(3.18)}$$

squares and can be factored. After cancelling out the common term in the numerator

$$[(r^p - 1)(r^p + 1)]/(r^p - 1) = (f_1 - f_3)/(f_1 - f_2) \qquad \textbf{(3.18)}$$

and denominator, an expression relating the grid refinement ratio and the order of accuracy to the numerical solutions is obtained in Equation 3.19. After simplification

$$r^p = [(f_1 - f_3)/(f_1 - f_2)] - 1 \qquad \textbf{(3.19)}$$

on the right hand and reversing the minus signs between the terms in the numerator and denominator, this reduces to Equation 3.20. Taking the log of 3.20 and solv-

$$r^p = (f_3 - f_2)/(f_2 - f_1) \qquad \textbf{(3.20)}$$

ing for p gives the desired relation (see Equ. 3.21).

$$p = \log[(f_3 - f_2)/(f_2 - f_1)]/\log(r) \tag{3.21}$$

This agrees with the form of the equation derived by Roache. As stated at the beginning of the derivation, Equation 3.21 does not apply to the selective directional grid refinements. For the selective approach, it would be appropriate to calculate the error terms directly from the estimated exact value defined by Equation 3.13, then determine the order of accuracy from the definition provided by Equation 2.1. Such an option is not yet provided for within VIVID.

It should be obvious that the approach of Equation 3.21 can not be used for two solution grid convergence studies. In this case, the choice is automatically limited to the order definition relation (i.e. Equ. 2.1). However, an evaluation of this variety is superfluous in VIVID since the answer will be preordained as the numerical method formal order value supplied by the user. This is true since the assumed formal order was used to generate the estimated exact soution value. In turn, the estimated exact value is then used to calculate the solution order. They are not independent in a two solution study. Two solution studies only yield independent beneficial information if a benchmark value is available to replace the estimated exact value in order calculations.

4. Code Development and Functionality

The Visualization of Instrumental Verification Information Details program is the result of a sizable code development effort. It included the writing of the constitutive driver program (a.k.a. VIVID) as well as the three subprograms required for various options. The driver routine is strictly a controlling program with no indigenous analysis capabilities. Its organization and function are relatively simple. Its primary purpose is to initiate the run option desired by the user. A synopsis of the functionality features included in the driver program and the three original subprograms written for VIVID is given in the following sections. Further details are provided in Appendix B. The programs are discussed in the order in which they would be initially executed in an order of accuracy analysis.

4.1 Main Program: VIVID

VIVID is a Fortran 77 code which runs on the SRN LAN employed by the 9100 Engineering Sciences Center (Unix platforms only - such as sass2889). It is an interactive procedure that is invoked by typing *vivid* from a command line window. It may be run in two modes: interactive or control file driven. Following is a discussion of the primary routines structure and functional attributes. Included in the discussion are its use of command line arguments and interrogative features.

4.1.1 Driver Routine Structure

Functionality of the main program can be categorized into three parts: command line parser, interactive parser, and program option control. The command line parser determines the disposition of the control file. If one exists, control file mode is initiated. If there is not a control file, a new control file is opened and the interactive mode is initiated. All interactive instructions made by the user in the program queries are written to the new file. Following determination of the control file disposition, the screen interactive parser of the driver program is executed. Its function is to define the subprogram execution option (i.e. ORDER, EXTRACT, or BENCHMARK). Next, the program option control portion of the main routine is executed. This section initiates the desired subprogram via a Fortran subroutine call. Following completion of the selected subprogram option, termination control is exercised.

4.1.2 Command Line Arguments

VIVID has the capability to interpret command line arguments. This is limited to one argument on the command line invocation. The argument following the *vivid* syntax is the name of the control option input file. If no argument is found, all required input file names and control parameters are defined in the pursuant interactive interrogative screen sessions conducted by the main program and the subprogram codes it executes. A VIVID run may use anywhere from one to three of its constitutive programs during execution. The precise nature of the run is determined by the control option file or any required interactive screen session.

4.1.3 Interactive Interrogation & Constitutive Program Execution

Assuming an input control file run is not invoked, VIVID uses a screen interactive parser to establish the execution control parameters. Most of these parameters are defined in the subprograms. Subprogram interactive interrogations are discussed beginning in Section 4.2. A structural summary of the EXTRACT, ORDER, and BENCHMARK subroutines is outlined in Sections 4.2, 4.3, and 4.4, respectively. Structural subprogram details are provided in Appendix B. The driver program establishes two fundamental variables. The first is declaration of the subprogram option. This determines whether to invoke ORDER, EXTRACT, or BENCHMARK. In addition, the main program controls the run repeat option. Whether interactive or control file mode has been chosen, execution parameters containing solution variables and extent are defined within the solution input files.

4.2 EXTRACT Subprogram

EXTRACT was written to isolate and output specific independent and dependent variables of interest for further analysis. It can be ran independently or as part of an order of accuracy evaluation. In the independent mode, files are sorted serially with repetition terminated at the user's discretion. In the case of error and order accuracy analyses, two or three numerical solution files are accepted serially as input, processed per the user's instructions, and output as column formatted files for use by the ORDER subroutine. Data can be input either as TECPLOT or EXODUS data base files/

The program begins with an interactive interrogation section that establishes an initial set of control variables for the file to be extracted. Domain, grid, input file type, and the all dependent variable option flags are defined in this set. Following definition of the initial group of flags, the data extraction section of the routine's interactive parser is encountered. This section of the subprogram is repeated for each file evaluated and defines the input and output file names. File name selections are immediately followed by specification of a group of variables that define the extraction extent of a given file. This includes extraction zones, extraction variables, extraction time planes, and extraction nodes. Following extraction extent definition, files are immediately read by calls to subroutines specifically written to be compatible with the given input file type. Multiple output files may be produced by this subprogram including columnar files for ORDER evaluation as well as TECPLOT files.

4.3 ORDER Subprogram

ORDER was written to estimate the order of accuracy and error of numerical solution methods. To calculate order independently, three numerical solutions are required for the procedure. Two solution error estimate options are available if the order is assumed equal to the algorithm formal order of accuracy or if a benchmark reference is available.

The program begins with an interactive interrogation section that establishes an initial set of control variables for the files being evaluated. Domain, grid, number of input files, two-dimensionality, and interpolation option flags are defined first. Following definition of the initial group of flags, control is passed from ORDER to the EXTRACT subprogram. The input information to be evaluated in ORDER is sorted out by EXTRACT and passed back to it in the form of columnar input files. An option exists to bypass the extraction run and operate on a series of preexisting ORDER input files. Immediately upon regaining control from the extraction subroutine, input and output file names are requested. **Input file names must be entered in order from coarse grid to fine grid.** Each solution must be entered in an independent file. All input solutions can **NOT** be contained in one file. If this is a domain run (i.e. the entire computational geometry), a TECPLOT PREPLOT format output file is also produced for multi-dimensional visualization purposes.

Input and output data definitions are immediately followed by specification (through interactive queries or control file reads) of a group of variables that define the analysis options governing the order evaluation. This includes grid refinement factor, numerical solution formal order of accuracy, error calculation options, and BENCHMARK solution flags. Unless otherwise specified, error is determined for the finest grid solution. The interactive parser section concludes after defining the analysis control variables. At this point, input files are immediately read and the information is saved in storage arrays for ease of grid point matching between solutions. Variable storage considerations require tracking degrees of freedom for three spatial directions, variable number, zone number, and file number. Following read completion, the multi-dimensional variable storage arrays are processed by a grid point coordinate matching subroutine. After the grids have been matched, error and order of accuracy analyses are completed. These analyses are based on the detailed equations presented in Section 3. The analysis results in the production of order of accuracy and error variables for all dependent variables analyzed. Results are written to a TECPLOT PREPLOT block format file (*order_tec.o*) for multi-dimensional visualization. As well, a column formatted output file with details of the extrapolation analysis is written to file *order.o*. This file contains specific extracted, extrapolated, and benchmark values at all domain points analyzed for a dependent variable. The order and error variables are also included in this file.

4.4 BENCHMARK Subprogram

BENCHMARK produces reference data for comparison to numerical program solutions. It can be run independently or as part of an order of accuracy analysis. In the independent mode, it is accessed repetitively directly from the main routine with repetition terminated at the user's discretion. In the case of order accuracy evaluations, it is accessed by the ORDER subprogram and executed at each point for which an accuracy analysis is being performed. Column formatted output files are generated for the reference solutions.

The BENCHMARK subprogram control routine is similar to the main routine of VIVID. It is a driver program with no indigenous analysis capabilities. It acts as an analysis traffic director based on the user's directions. Currently, it contains two analytical subroutines for verification purposes. However, it is written in a highly modular fashion for ease of expansion to include future verification subroutines. The current reference routines available for comparison are the two-dimensional flat duct incompressible flow program and the two-dimensional heat conduction program. Multiple sub-options are available for both current analytical programs. Development work is proceeding on four other reference routines: Couette flow, inviscid cross flow cylinder, three-dimensional heat conduction, and manufactured solutions. The FLAT_DUCT and CONDUCTION_2D subroutines are discussed in detail in the following sections.

4.4.1 Flat Duct Incompressible Flow

FLAT_DUCT is an analytical 2-D planer laminar flow subroutine based on information contained in the convective heat transfer handbook by Kakac, et. al.[9] This procedure contains relations applicable to incompressible constant property low speed flow. Two types of inlet boundary condition problems are addressed in the routine. Problem types are categorized by inlet velocity profile. Solutions are given for a fully developed and a constant velocity inlet profile. Axial velocity, vertical velocity, and pressure are available from both analytical solutions.

Solution for the <u>fully developed inflow velocity profile is exact</u> and is based on the relation defined in Equation 4.1. In this equation, u, y, and b are the axial veloci-

$$\frac{u}{u_m} = \frac{3}{2}\left[1 - \left(\frac{y}{b}\right)^2\right] \qquad \textbf{(4.1)}$$

ty, vertical distance from the duct centerline, and duct half spacing, respectively. The range of y is from $-b$ at the bottom of the duct to b at the top of the duct. The subscript m refers to the mean inlet flow velocity and the origin for the coordinate system is assumed to be at the duct centerline. Half duct spacing (b) is equal to the one fourth of the hydraulic diameter (i.e. $D_h = 4b$). For this type of inlet condition, pressure is related to the mean axial velocity through Equation 4.2. Vertical velocity in

$$u_m = -\frac{1}{3}\left(\frac{dp}{dx}\right)b^2 \qquad \textbf{(4.2)}$$

fully developed flow is identically zero throughout the flowfield.

Analytical solution to hydrodynamically developing flow (for constant inflow velocity) is approximate in the developing region and should be used for comparison purposes only. It can not be considered as an exact reference for determining numerical solution error. However, it must be solved to get to the region of the analytical solution that is exact (as such, it is included in this document). The inherent value of this analytical solution comes at the point that the flow has reached fully developed status in the duct. At this point, the solution is exact and is a direct reference for ver-

ification of the numerical method. The developing region is considered to consist of two separate flow fields, the viscous boundary layer and the inviscid core region. These two merge vertically at a value of y coincident to the boundary layer thickness. Further, the developing region asymptotically approaches a fully developed flowfield in the duct axial direction at the hydrodynamic entrance length defined by Equation 4.3. In this relation, Re refers to the Reynolds number based on the hydraulic diame-

$$L_{hy} = D_h\left(0.011Re + \frac{0.315}{1 - 0.0175Re}\right) \tag{4.3}$$

ter. Independent analytical solutions exist for both the axial and vertical velocities in each of the two distinct flow fields of the developing region. All four velocity expressions are quite complex and will not be repeated here. However, it is noted that all expressions are closed functional forms that include a dimensionless velocity term (U) that is in turn an implicit function of x^+ (defined below in Equ. 4.8). Expressions 4.4 through 4.7 define the function variables for each velocity term. The first two

$$u_i = f(U, u_m) \tag{4.4}$$

$$v_i = f(U, y, u_m) \tag{4.5}$$

equations apply to the inviscid core region, while the next two are for the viscous developing boundary layer. The x^+ variable is defined in Equation 4.8. Making use of

$$u_v = f(U, y, u_m, \delta) \tag{4.6}$$

$$v_v = f(U, y, u_m, \delta) \tag{4.7}$$

$$x^{+} = \frac{x}{D_h Re} = f(U) \tag{4.8}$$

this definition allows the subroutine to solve for the dimensionless velocity term in an iterative fashion. Once it is determined, all four of the absolute velocities can be evaluated. Local pressure drop is also calculated as a complex function of the dimensionless velocity in the developing region. When the axial location exceeds the hydraulic entrance length, velocity and pressure calculations revert to the fully developed relations in Equations 4.1 and 4.2.

4.4.2 Two-Dimensional Heat Conduction

As in the flat duct subroutine, the heat conduction program begins with an interactive parser. In the heat conduction interactive parser, boundary conditions and material property variables are defined. In a variance from the flat duct approach, the user is allowed the option of defining these terms in an input file. An example of this file is shown in Appendix D. Initially, in the case of an ORDER run, the program displays the dependent variable being evaluated. The user is provided the option to

evaluate this quantity analytically. As above, the user may choose to bypass the routine at this point. If processing is continued, an analytical solution of the two-dimensional heat conduction equation is executed.

CONDUCTION_2D is an analytical subroutine developed by Blackwell and Dowding.[10,11] This procedure contains relations applicable to two-dimensional heat conduction in an infinitely long rod of rectangular cross section dimensions L and H. The routine successfully addresses both the steady state and transient conduction problems. Boundary conditions for the problem solved by the routine are illustrated in Figure 4.1. They consist of constant heat flux at the left vertical surface, constant

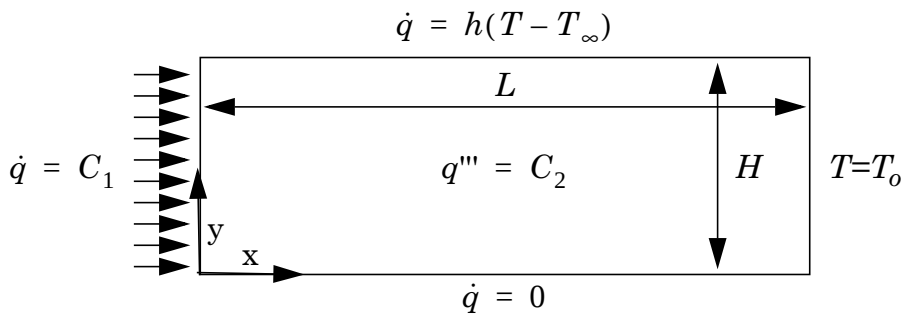

Figure 4.1 Two-Dimensional Conduction Boundary Conditions

temperature at the right vertical surface, adiabatic at the bottom horizontal surface, and convective heat removal at the top horizontal surface. The convective sink temperature (T_∞) and the convective coefficient (h) are assumed constant. In addition, constant uniform internal heat generation can be included. Material thermal properties within the domain are considered constant with temperature.

Solution to the two-dimensional heat conduction problem is exact and is based on the partial differential equations shown in 4.9 and 4.10 for the steady state and transient evaluations, respectively. In these equations, T, q''', and k are the tempera-

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{q'''}{k} = 0 \qquad \textbf{(4.9)}$$

ture, heat generation, and thermal conductivity, respectively. The terms a and t in

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{q'''}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \qquad \textbf{(4.10)}$$

Equation 4.10 refer to the thermal diffusivity and time. Blackwell and Dowding used the principle of superposition to break the problem of interest down into a series of smaller more manageable problems. The solution to each of the smaller pieces is an eigenvalue problem. Further details of the analytical developments for the steady state and transient analyses are contained in references 10 and 11 respectively. Resulting solution of the eigenvalues gives the temperature field of interest.

5. Sample Analyses & Results

VIVID's analysis capabilities are demonstrated in the following sections. Both structured and unstructured grid convergence studies are presented. Evaluations using data produced by a number of numerical programs are illustrated. Programs employed in the analyses included CFD-ACE,[12] MP-SALSA,[13] Calore,[14] Premo,[15] SACCARA[16,17] and Fuego.[18] These solutions and analyses were generated by multiple staff members and contributed work is noted by specific authors in major section titles. A summary of the sample analyses presented in this section is given in Table 5.1. Order of accuracy and error evaluations were performed for all sample numeri-

Table 5.1: Summary of VIVID Sample Problems

#	Code/ Formulation	Solution Originator	Mechanics	Grid Type	Spatial Geometry	Temporal
1	MP-Salsa/ Finite Element	F. G. Blottner M. A. Payne	Incompressible Fluid	Unstructured	2D Thermal Cavity	Steady State
2	CFD-ACE/ Finite Volume	Potter & Black	Incompressible Fluid	Structured	2D Flat Duct	Steady State
3	SACCARA/ Finite Volume	C. J. Roy	Compressible Fluid	Structured	2D Inviscid Nozzle	Steady State
4	Premo/ Finite Volume	C. J. Roy	Compressible Fluid	Unstructured	2D Manufactured	Steady State
5	Fuego/ Finite Volume	A. R. Black	Incompressible Fluid	Unstructured	2D Flat Duct	Steady State
6	Calore/ Finite Element	B. Blackwell	Heat Conduction	Unstructured	2D Infinite Rod	Steady State
7	FCV-FAM/ Finite Volume	D. L. Potter	Heat Conduction	Structured	2D Infinite Rod	Steady State
8	FCV/Fuego Finite Volume	M. A. Payne	Incompressible Fluid	Unstructured	2D Duct	Transient
9	Calore/ Finite Element	B. Bainbridge	Heat Conduction	Unstructured	3D Cube	Steady State

cal solutions. ***All graphical illustrations of order and error are presented on the coarsest mesh used in each grid convergence study***. Problems ranged from an incompressible flat duct flow to a hypersonic compressible reentry vehicle. Various dependent variables are illustrated through the different sections. Two and three dimensional samples are shown including the unique transient option analysis treating time as the third dimension. ***These examples are not meant to be conveyed as rigorous order of accuracy studies for the various codes employed***. ***The point is to show the range of application over which the VIVID tool can be exercised***. It should be further noted that the VIVID results for the first sample problem are analyzed in detail to illustrate difficulties in determining the order and percent error of numerical solutions. Subsequent sample problems rely on this discussion and refer to it, but are presented at a reduced level of detail.

5.1 MP-Salsa Cavity Unstructured Grid Convergence Analysis (Fred Blottner)

The first sample problem is concerned with testing Richardson extrapolation methodology with the VIVID code for a well established benchmark problem of thermally driven flow in a square cavity. This problem has been carefully investigated by de Vahl Davis,[19] de Vahl Davis and Jones,[20] and Hortmann, Peric and Scheuerer[21] for a range of Rayleigh numbers. The present solutions to this problem have been obtained by McWherter-Payne with the finite element code MP-Salsa with the results documented in a Sandia Memo by McWherter-Payne, Lopez and Cochran.[22]

Richardson extrapolation requires the order of the numerical scheme to be known or additional numerical solutions are required to determine the order. Experience with determination of the order throughout a solution domain has shown that unusual behavior can occur. The order predictions are very sensitive to the accuracy of the numerical solutions being used. If the numerical scheme is second-order, at certain locations the order can be significantly higher or lower than two. Also the solution error and the order of the numerical scheme is determined with the Richardson extrapolation method from the solution on three different meshes. The present study for a Rayleigh number of 1000 uses numerical solutions with a mesh refinement factor of $r = 2$. The number of mesh cells and the applicable mesh names used in this document are as follows:

Crude Mesh: 20 x 20	400 elements
Coarse Mesh: 40 x 40	1600 elements
Medium Mesh: 80 x 80	6400 elements
Fine Mesh: 160 x 160	25,600 elements

The VIVID code has been used to determine the solution order and error of one of the dependent variables (horizontal velocity) in the thermally driven cavity flow problem. The present results are compared to the previously obtained results presented in the Reference [20}. In addition, further analysis of the results are presented.

Problem Description:

The benchmark problem is the thermally driven flow in a two-dimensional square cavity: with height H and width W equal to 0.062178 m. The left wall is hot with temperature $T_{hot} = T_{ref} + \Delta T/2$ and the right wall is cold with temperature $T_{cold} = T_{ref} - \Delta T/2$ where the reference temperature $T_{ref} = 303.0K$ and $\Delta T = T_{hot} - T_{cold} \approx 0.04630K$. The top and bottom walls have zero heat flux. The tangential and normal gas velocity is zero at the four walls. The governing equations are the Oberbeck-Boussinesq form of the Navier-Stokes equations. The Rayleigh Number is one thousand for the case considered in this study and is defined as

$$Ra = \rho_{ref}^2 c_p g \beta \Delta T L_0^3 / k\mu = 1000$$

The specified values of the parameters in this simulation are as follows:

$$\rho_{ref} = 1.165 \ kg/m^3 \qquad c_p = 1005 \ J/kg\text{–}K \qquad g = 9.81 \ m/s^2$$

$$L_0 = H = W = 0.062178 \ m \qquad k = 0.02637 \ W/m\text{–}K$$

$$\mu = 1.864 \times 10^{-5} \text{ kg/m-s}$$

The calculated values of the parameters in this simulation are

$$Pr = c_p \mu / k \approx 0.7104 \qquad \beta = 1/T_{ref} \approx 0.0033003$$

$$\Delta T = (k \mu Ra)/\rho_{ref}^2 c_p g \beta L_0^3 \approx 0.04630$$

The steady flow in the cavity is clockwise and the horizontal or x-component of velocity $v_x = VX$ is given in Figure 5.1.1. The velocity component is zero across the

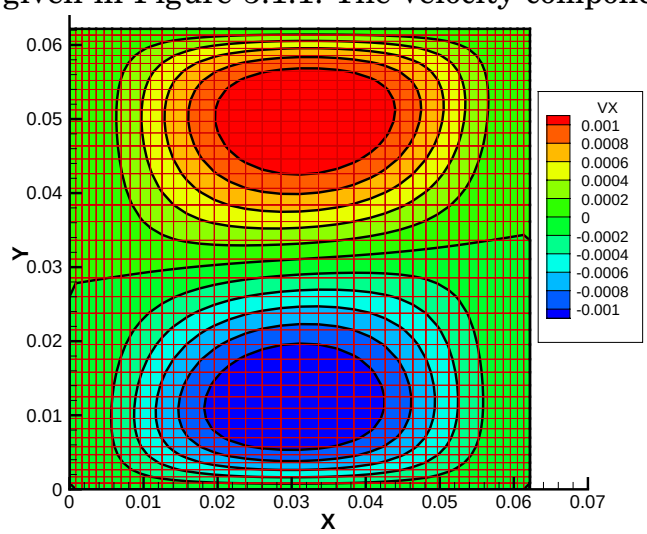

Figure 5.1.1 - MP-Salsa Horizontal Velocity Contours

cavity near the center of the cavity. The 40 x 40 mesh is also shown in this figure.

Thermally Driven Cavity Flow Solution Analysis:

Order of Numerical Scheme

The analysis presented in this report has carefully evaluated the solution behavior of only the horizontal velocity, one of several dependent variables in this problem. See Reference [20] for a complete evaluation of all of the dependent variables. The spatial order of the numerical scheme p, as determined from the numerical solution of one of the dependent variables f with three refined meshes, was computed from the equation below. This is essentially a repeat of Equation 3.21 as derived in

$$p_{cmf} = \log[(f_{coarse} - f_{medium})/(f_{medium} - f_{fine})]/\log[r] \qquad \textbf{(3.21)}$$

the theory discussion in Section 3.3. Subscript cmf indicates that the coarse, medium, and fine meshes are used. The parameter r is the mesh refinement factor and $r = 2$ for the present problem. The order of the finite element code MP-Salsa is sec-

ond order as shown previously[22] and the present results will indicate the same behavior as the mesh is refined. The present analysis of the solution with the VIVID code for the order of the numerical scheme p, as determined from the solution of the horizontal velocity v_x with coarse, medium, and fine meshes, is presented in Figure 5.1.2. These results are in close agreement with the results presented in Figure 5 of

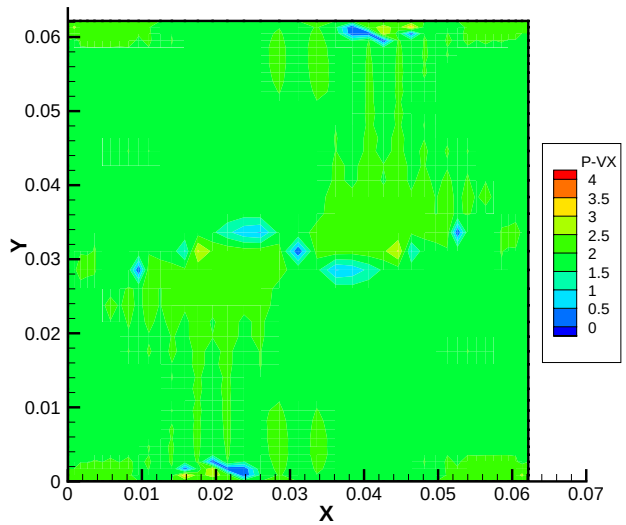

Figure 5.1.2 - Horizontal Velocity Solution Order

Reference [20]. There are points across the center of the cavity where the solution order is erratic. In addition, there is a region near the bottom and upper walls where erratic behavior also occurs. However the symmetry in the solution is apparent. A replot on the data is given in Figure 5.1.3 where the order is restricted to $1.7 < p < 2.3$. This figure shows the regions where the order is not near the expected

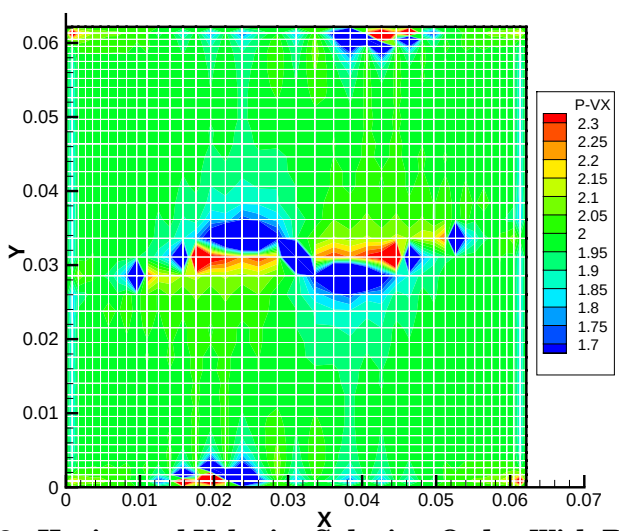

Figure 5.1.3 - Horizontal Velocity Solution Order With Refined Scale

value of 2. The order of the numerical scheme using the crude, coarse, and medium meshes is presented in Figure 5.1.4. These two figures indicate that as the mesh is refined, the region of non-second-order behavior decreases as the solution approaches the asymptotic range.

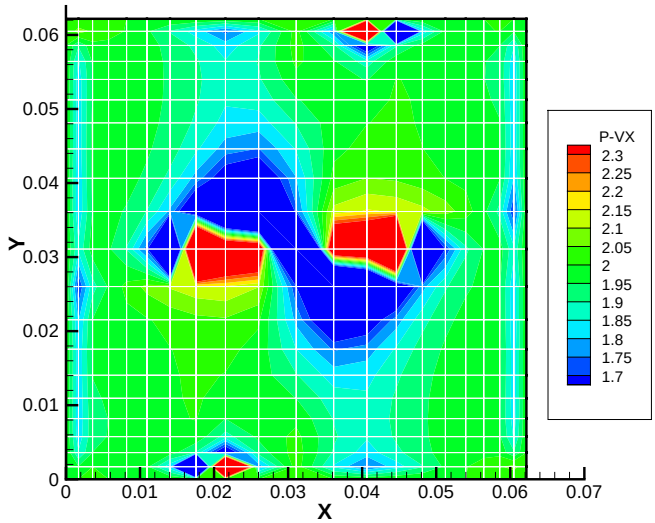

Figure 5.1.4 - Horizontal Velocity Solution Order With Crude Mesh

It is not known how much of this non-second-order behavior can be removed with further mesh refinement. *Some of the erratic behavior of p is probably due to the solutions not being in the asymptotic range.* If the solutions are in the asymptotic range, then the ***discretization error*** e at any spatial location on the three meshes are related as follows for a *second-order* numerical scheme:

$$e_{coarse} = (f_{coarse} - F) = 16(f_{fine} - F) = 16e_{fine} \qquad e_{medium} = 4e_{fine}$$

$$p = \log(12/3)/\log 2 = 2$$

where F is the exact solution. When most of the solutions at the various mesh points indicate a second-order scheme, perhaps the regions of poor behavior of p can be an indication the asymptotic range has not been obtained in these regions. For the solutions with the coarse, medium, and fine meshes, there are 1521 interior mesh points and 160 points on the wall where the boundary condition specifies zero velocity. Therefore the solution is exact at the wall for the velocity components for all of the meshes and the order of the scheme can not be determined at the wall mesh points. This is clearly shown when Equation 3.21 is written in terms of the *discretization error* as the numerical scheme order equation becomes indeterminate since $e_{coarse} = 0$, $e_{medium} = 0$, and $e_{fine} = 0$ (see Equ. 5.1). For the present case, the order

$$p_{cmf} = \log[(e_{coarse} - e_{medium})/(e_{medium} - e_{fine})]/\log[r] \qquad \textbf{(5.1)}$$

predictions satisfy $1.7 < p < 2.3$ for all the points except 40 mesh points (2.6%).

The behavior of the solution for the horizontal velocity has been investigated at three mesh points. The location of the mesh points and the order of the numerical scheme as determined from the solutions are given in Table 5.2. The subscript ccm indicates that the crude, coarse, and medium meshes are used while the subscript cmf indicates that the coarse, medium, and fine meshes are used. The numerical so-

lutions, solution percent errors, and discretization error, $E|_{\Delta}$, at the three mesh points are given in Table 5.3, Table 5.4, and Table 5.5. The behavior of the solution

Table 5.2: Solution Scheme Order Behavior at Three Locations

Mesh Point Number	x	y	p_{ccm}	p_{cmf}
216	0.01094	0.03109	1.727	1.939
217	0.01405	0.03109	0.5104	1.764
218	0.01756	0.03109	3.477	3.074

Table 5.3: Solution of v_x at the Mesh Point $N = 216$

$p_{cmf} = 1.939$ $x = 0.010943$ $y = 0.03109$			
Mesh	$v_x \times 10^5$	E, % error relative to F_{RE}	$E_{\Delta} \times 10^7$
20 x 20	6.34738681	2.619	16.20
40 x 40	6.23224259	0.7578	4.688
80 x 80	6.19745915	0.1955	1.209
160 x 160	6.18839040	0.04887	0.3023
F_{RE}	6.18536748		

Table 5.4: Solution of v_x at the Mesh Point $N = 217$

$p_{cmf} = 1.764$ $x = 0.01405$ $y = 0.03109$			
Mesh	$v_x \times 10^5$	E, % error relative to F_{RE}	$E_{\Delta} \times 10^7$
20 x 20	6.91596506	0.6127	4.212
40 x 40	6.89466906	0.3029	2.082
80 x 80	6.87971915	0.08539	0.5869
160 x 160	6.87531719	0.02135	0.1467
F_{RE}	6.87384987		

Table 5.5: Solution of v_x at the Mesh Point $N = 218$

$p_{cmf} = 3.074$ $x = 0.01756$ $y = 0.03109$			
Mesh	$v_x \times 10^5$	E, % error relative to F_{RE}	$E_\Delta \times 10^8$
20 x 20	6.47975903	- 1.399	- 91.97
40 x 40	6.56306220	- 0.1318	- 8.665
80 x 80	6.57054261	- 0.01802	- 1.185
160 x 160	6.57143100	- 0.004506	- 0.2961
F_{RE}	6.57172713		

error for the horizontal velocity for the three mesh points with a second-order scheme assumed $(p = 2)$ is given in Figure 5.1.5. It appears that the solutions at

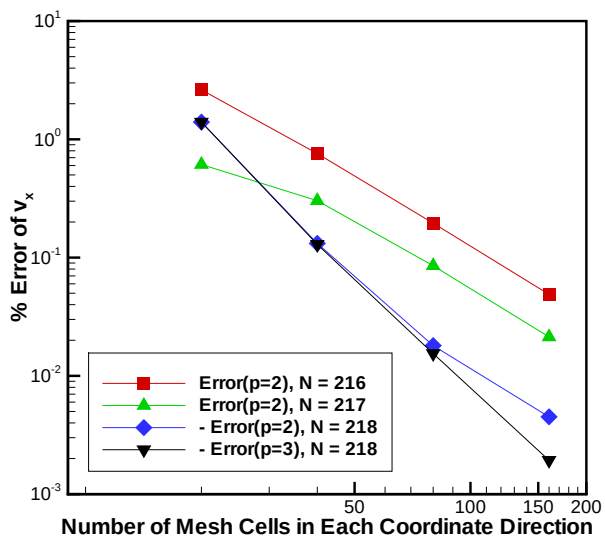

Figure 5.1.5 - Error Behavior vs. Mesh Refinement at Three Points

mesh points 216 and 217 are approaching second-order behavior as the mesh is refined. At mesh point 216, the solution is in the asymptotic range; while at mesh point 217, the solution is still approaching the asymptotic range. The solution at mesh point 218 is approaching third-order behavior. If third-order behavior is assumed, the Richardson extrapolation for the exact solution gives

$$F \approx F_{RE} = f_{fine} - (f_{medium} - f_{fine})/7 \qquad \text{when } r = 2$$

The solution error at mesh point 218 has been determined assuming second-order and third-order schemes and these results are also given in Figure 5.1.5.

Another issue can also confuse the evaluation of the order of a numerical scheme and produce an *enhanced scheme order* at a limited number of points in the

solution region. This type of behavior appears to be occurring at mesh point 218. The numerical solution for a two-dimensional problem is related to the exact solution F as follows for a second-order numerical scheme (see Blottner [1]):

$$f = F + a_2\Delta x^2 + b_2\Delta y^2 + a_3\Delta x^3 + b_3\Delta y^3 + \ldots$$

If $a_2\Delta x^2 + b_2\Delta y^2 = 0$, then a higher-order scheme $(p > 2)$ can be artificially indicated in local regions in the solution domain. For this anomaly in solution order to occur, a_2 must be of the opposite sign of b_2 and the mesh ratio must satisfy the relation

$$\Delta x/\Delta y = \sqrt{-b_2/a_2}$$

With the present mesh refinement approach of reducing both Δx and Δy by a factor r, the *higher order behavior does not disappear as the mesh is refined*. To avoid this behavior, the mesh refinement study should fix Δx_0 and obtain solutions on a series of meshes with Δy refined. In this case, $f \to F(\Delta x_0)$ and the numerical solution is expressed as

$$f = F(\Delta x_0) + b_2\Delta y^2 + \ldots \qquad F(\Delta x_0) = F + a_2\Delta x^2 + \ldots$$

With this procedure, the order of the numerical scheme is approached as the mesh is refined in the y-coordinate direction. This process is repeated with a fixed Δy_0 and for a series of meshes with Δx refined.

If the error of the dependent variable is near zero, then numerical round-off errors can result in poor accuracy of the solution of p from Equation 3.21 and as the error of the dependent variable approaches zero, the solution becomes indeterminate. This behavior is understood by writing the numerical scheme order equation in terms of the *discretization error* as given in Equation 5.1. As will be shown subsequently, there is a line across the cavity near the center (y coordinate) where the discretization error is zero. In addition, near the lower and upper walls there are regions where the error is also zero. At locations where the discretization error is zero, Equation 5.1 becomes indeterminate and the evaluation of the order p becomes impossible. The occurrence of zero solution error at certain locations explains the erratic behavior of p across the cavity near the center and at the lower and upper walls when the solutions are in the asymptotic range.

Numerical Solution Error

The ***percent error*** of the numerical solution for the dependent variable f is defined as shown in Equation 5.2 where F is the exact solution to the problem for the

$$\%\ \text{Error of } f = E = 100(f - F)/F \qquad \textbf{(5.2)}$$

dependent variable f. When the numerical solution is greater than the exact solution $f > F$, the percent error will be positive. When the numerical solution is less than the exact solution $f < F$, the percent error will be negative. Since the exact solution is rarely known, the value F is approximated with the Richardson extrapolated value F_{RE} which is obtained for a *second-order* numerical scheme from the

relation

$$F \approx F_{RE} = f_{fine} + (f_{medium} - f_{fine})/3 \qquad \text{when } r = 2$$

The absolute value of the percent error of the horizontal velocity v_x is given in Figure 6 of Reference [20]. VIVID has been used to reproduce this result and is given in Figure 5.1.6. The agreement with the previous results is very good and helps to verify the VIVID code. This figure shows that <u>absolute</u> percent error of v_x is largest near the left and right walls. Further mesh refinement near these walls is needed to reduce this error.

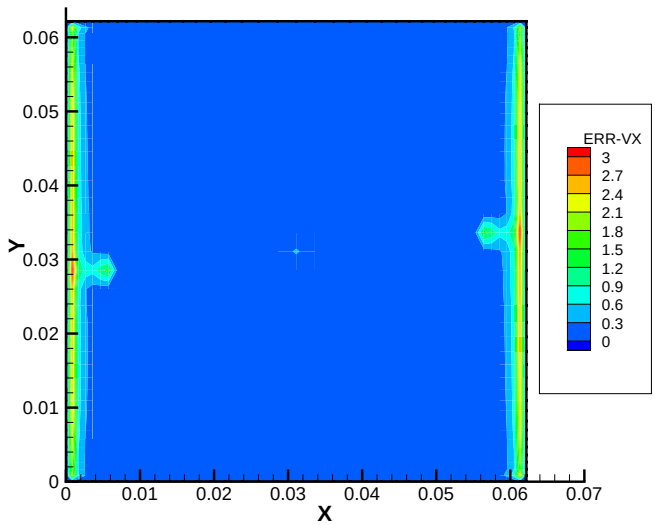

Figure 5.1.6 - Absolute Error of Horizontal Velocity for Fine Mesh

Further evaluation of the sign of the percent error and its asymptotic behavior is illustrated in Figure 5.1.7 for the fine mesh and in Figure 5.1.8 for the coarse mesh. The percent error of the coarse mesh solution should be 16 times the fine mesh

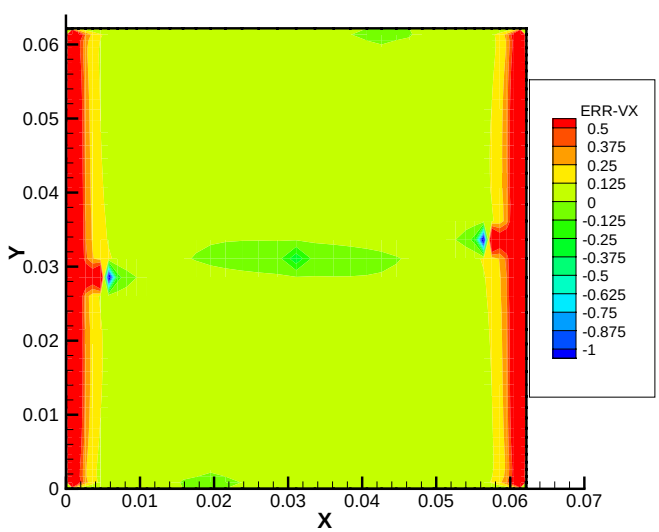

Figure 5.1.7 - Percent Error of Horizontal Velocity for Fine Mesh

solution. If this is true, the two figures should be identical. This condition is nearly satisfied except at the center of the cavity. This result indicates the 3 solutions are in the asymptotic range and the order result should be accurate except at the horizontal (y=H/2) center of the cavity.

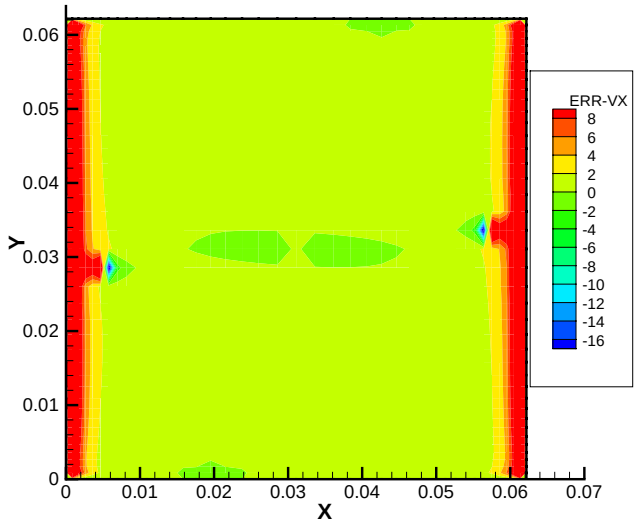

Figure 5.1.8 - Percent Error of Horizontal Velocity for Coarse Mesh

The percent error approach will have problems where the solution variable is zero as one is dividing by zero in Equation 5.2. In the current example problem, this difficulty shows up where $v_x = 0$ across the center of the cavity as shown in Figure 5.1.1. To avoid this problem, a ***discretization error*** can be introduced for each mesh point as

$$e = (f - F) \approx (f - F_{RE})$$

The discretization error of v_x for this problem has been obtained with the VIVID code and is presented in Figure 5.1.9 for the fine mesh and in Figure 5.1.10 for the coarse mesh. Since the error range in Figure 5.1.10 is 16 times the error range in

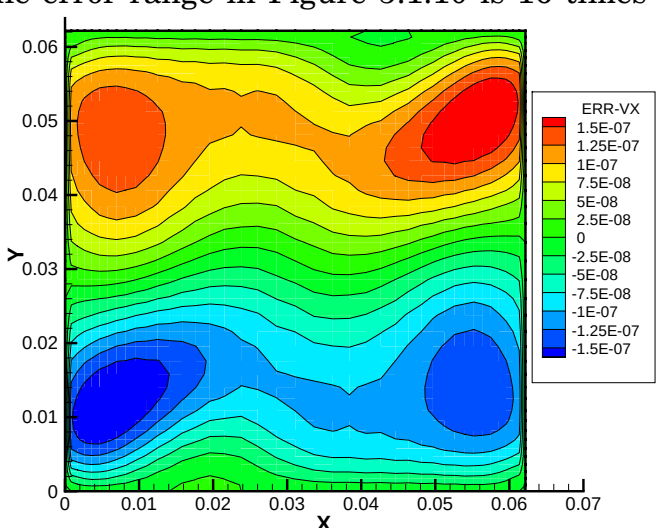

Figure 5.1.9 - Discretization Error of Horizontal Velocity for Fine Mesh

Figure 5.1.9, the two figures should be identical if the solutions are in the asymptotic

range. Again, the results indicate the three solutions are approximately in the asymptotic range. The discretization error is smooth without any unusual behavior for

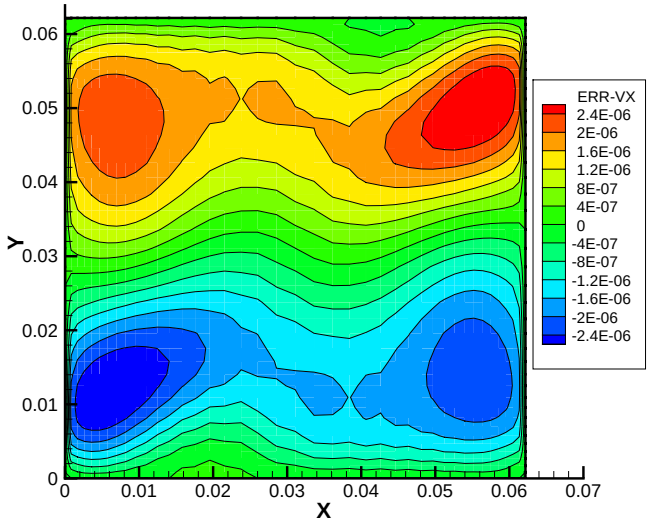

Figure 5.1.10 - Discretization Error of Horizontal Velocity for Coarse Mesh

this problem. It is important to notice that the discretization error is zero near a line across the center of the cavity and, of course, along the four walls of the cavity. There are regions where this error is larger and where mesh refinement is needed if the ideal of a uniform discretization error is to be obtained.

The locations where the percent error and numerical scheme order results for the horizontal velocity have difficulties are presented in Figure 5.1.11 and Figure 5.1.12. Horizontal velocity and discretization error are zero at the interface between

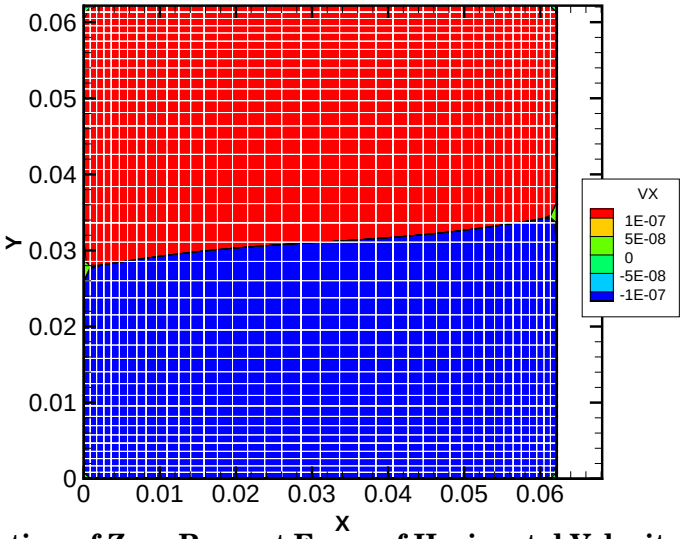

Figure 5.1.11 - Location of Zero Percent Error of Horizontal Velocity for Fine Mesh

the red and blue regions. The regions of zero horizontal velocity occur across the center of the cavity. Comparison of Figure 5.1.11 with Figure 5.1.7 shows that larger errors occur across the center of the cavity. Regions of zero discretization error occur across the center of the cavity and at a small region near the top and bottom walls. A

comparison of Figure 5.1.12 with the numerical scheme order results given in Figure 5.1.3 shows that the erratic behavior of the order calculation occurs in the locations where the discretization errors are near zero. The locations where the discretization

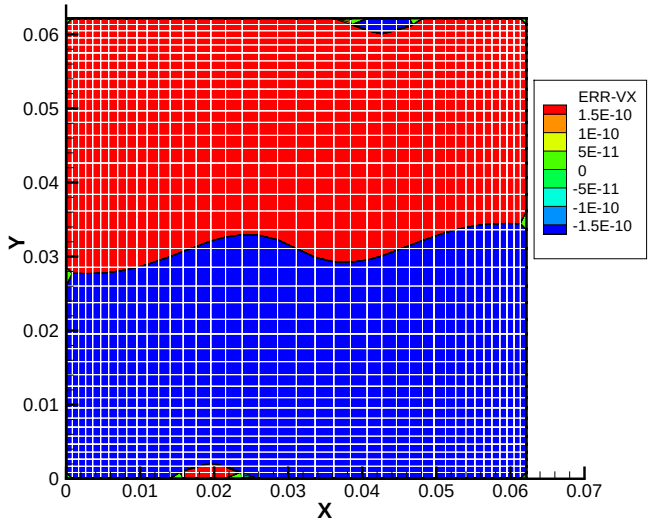

Figure 5.1.12 - Location of Zero Discretization Error of Horizontal Velocity for Fine Mesh

error is zero for the other dependent variables will be different than the present results. In the analysis of the order of a numerical scheme, the behavior of the numerical solution error needs to be understood first. At mesh points where the numerical solution discretization error is near zero, the predictions of the order of a numerical scheme should be ignored as these results are unreliable.

5.2 CFD-ACE Flat Duct Structured Grid Convergence Analysis (Donald Potter)

A two-dimensional flat duct incompressible flow with constant inlet velocity at the left boundary was evaluated with the structured finite volume CFD-ACE numerical program. Steady state solutions were obtained with x and y nodal dimensions of 51 x 11, 101 x 21, and 201 x 41, respectively (refinement factor of 2).

Problem Description:

The second sample problem demonstrates VIVID analysis of incompressible flow in a two-dimensional flat duct: with height H and length L equal to 0.50 and 22 m respectively. The top and bottom walls are fixed at a constant boundary condition temperature of 310 K. Flow entering the duct at the left boundary has a temperature of 300 K. The gas velocity at the inlet on the left is constant across the height with an axial value of 0.0171 m/s. Gas velocity boundary conditions of zero are set for the axial and vertical components at the top and bottom walls. In addition, a vertical velocity of zero is set at the inlet. The Reynolds Number is 1000 and is defined as

$$Re = \rho u_e D_h / \mu = 1000$$

D_h in the Reynolds number is the hydraulic diameter and has a value of 1 m in this problem. The other constant specified values of the parameters in this simulation are as follows:

$$\rho = 1.1563 \ kg/m^3 \qquad c_p = 1000 \ J/kg-K \qquad H = 0.50 \ m$$

$$L = 22.0 \ m \qquad Pr = c_p \mu / k \approx 0.70$$

$$\mu = 1.977 \times 10^{-5} \ kg/m-s$$

The fine mesh axial velocity field (on the coarse grid) is shown in Figure 5.2.1 The x

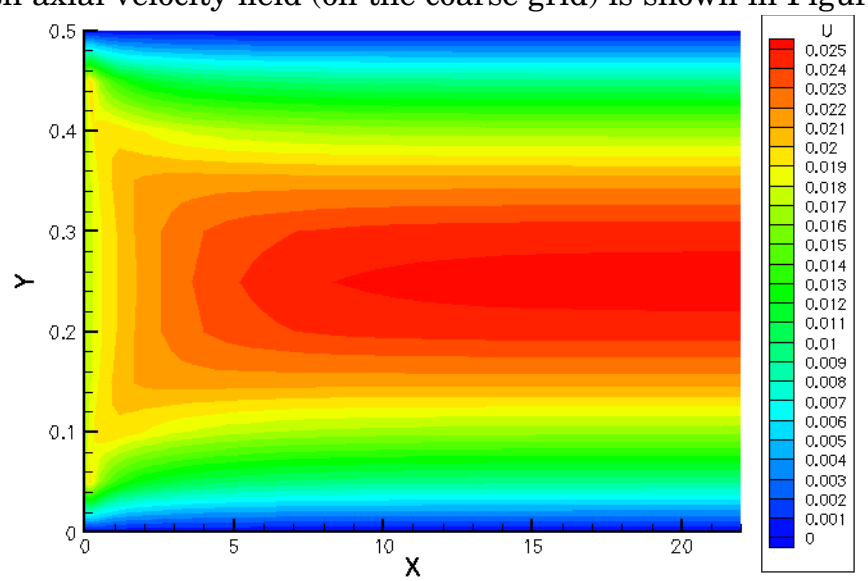

Figure 5.2.1 - CFD-ACE Flat Duct Axial Velocity Contours

and y proportionality was sacrificed to increase detail of the illustration.

Flat Duct Incompressible Flow Solution Analysis:

The order of accuracy (using the estimated exact solution as a reference) over

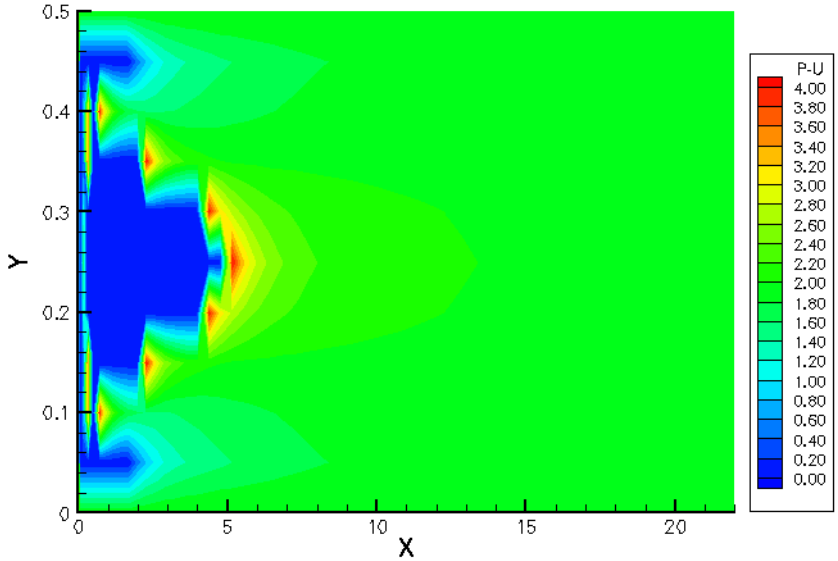

Figure 5.2.2 - CFD-ACE Structured Grid Velocity Order

the domain for fine grid axial velocity solution is shown in Figure 5.2.2, but illustrated at the locations of the coarse grid points (i.e. only locations where three solutions exist). The full width of the duct is illustrated as evidenced by the symmetry of the solution. VIVID limits the order value output between zero and four. Values outside these bounds are set to the limits. A second order accurate solution is seen over the majority of the domain (shown in green). Regions of blue indicate order of accuracy approaching zero. This should be interpreted with caution. These regions are subject to the same asymptotic range limitation presented by Blottner in the first problem.

The definition in Equation 2.1 indicates that a zero order implies that no error reduction is obtained by refinement of the grid. At the inlet and wall boundaries, this is certainly true since the values of dependent variables are established as boundary conditions and do not change regardless of how fine a mesh is used to solve the problem. It should be noted that by definition in VIVID where consecutive solution values are identical (e.g. at specified boundaries), the order value is set to two. It appears the boundary condition specification still propagates lower order contamination into the body of the solution. Possibly, the definition is causing the contour corruption. Interpretation of regions of zero order away from the boundaries requires closer examination of order evaluation Equation 3.21. Variables f_1, f_2, and f_3 refer to the dependent variable solution values for grids one to three from finest to coarsest, respectively. Since expression 3.21 is logarithmic, any argument of the numerator of this function that is one or less will produce a zero entry in the illustration. For this to occur when this argument is positive, there must be less error change between the coarse and medium grid solution than between the medium and fine grid simulation. When this argument is negative, f_2 does not lie between f_3 and f_1. This behavior is

contradictory to the asymptotic region requirement that grid refined solutions must approach the "true" solution from the same side; either from above or below. Further grid refinement is necessary to resolve the problem. This is precisely the erratic behavior discussed by Blottner in Section 5.1 in reference to the indeterminate nature of the order equation; either in the form of Equation 3.21 or 5.1.

Regions of order higher than second also deserve examination (yellow and red areas). Equation 3.21 implies the opposite of the observation made for the blue regions. Error reduction between the coarse and medium solutions is much greater than between the medium and fine grids. The three solutions are well into the asymptotic behavior for the numerical method. An order value of four indicates that the local error in these regions is reduced by a factor of sixteen by doubling the number of grid points. Since the numerical method is not fourth order (i.e. no terms above second order are saved in the numerical method's Taylor series expansion), another explanation must account for this behavior. Blottner advanced the discussion of this topic in the first sample problem through postulation of the sign differences for the error coefficients in the series expansions used to estimate the exact solution of the dependent variables. That discussion is extended here to further define the error coefficients. Discretization error is defined by Blottner[2] as the difference between dependent variable values obtained by solution of the differential equation and the difference equation describing the same problem. It is the discretization error that is used in determining the order of accuracy of the numerical method. Starting with Equation 3.7, repeated here for convenience, the discretization error is represented

$$F_e = f_1 + \alpha\delta_x^p + \beta\delta_y^p \qquad \textbf{(3.7)}$$

by Equation 5.3. Coefficients α and β in this relation are referred to here as the di-

$$E = F_e - f = \alpha\delta_x^p + \beta\delta_y^p \qquad \textbf{(5.3)}$$

rectional error coefficients. They are indicative of the solution error resulting from discretization refinement in the x and y directions, respectively. When multiple grid solutions have been obtained for a problem, it is possible to solve for these coefficients from expressions of the form of Equation 5.4. Examination of the directional

$$\alpha = [(f_1 - f_2)/(r^p - 1)]/\delta_x^p \qquad \textbf{(5.4)}$$

error coefficients for regions showing higher order behavior indicates that they are of opposite sign. Error cancellation produced by multiple directional refinements gives the appearance of higher order behavior.

For comparison purposes, the order evaluation for a second dependent variable from the flat duct numerical solution is shown. Figure 5.2.3 shows the order of accuracy achieved for static pressure. As with the velocity variable, most of the domain exhibits second order behavior. Also similar to the velocity evaluation, the regions showing lower order tendencies are those dominated by boundary conditions. Regions of higher order are more limited than from the previous variable analysis.

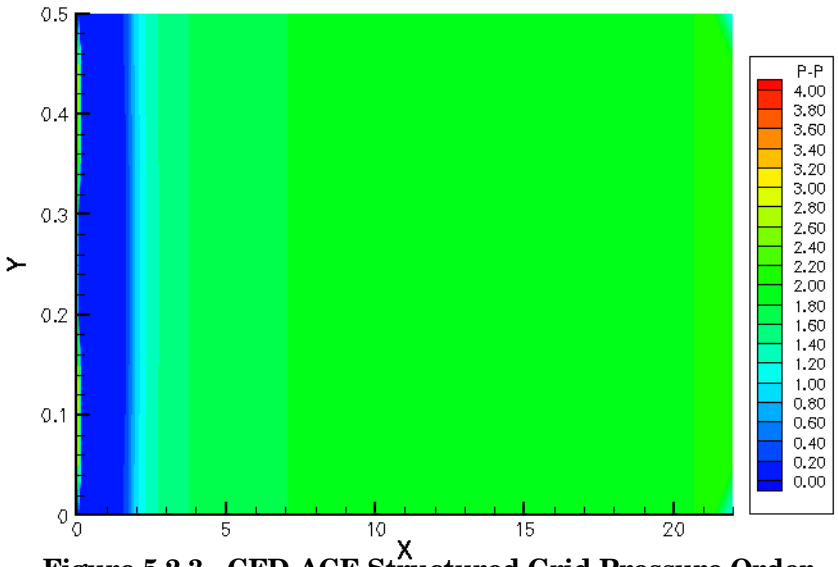

Figure 5.2.3 - CFD-ACE Structured Grid Pressure Order

Proper evaluation of the quality of a numerical solution also requires examination of the error characteristics as well as its order of accuracy (see Fig. 5.2.4). In practical terms, this variable is more meaningful and in many ways more useful to analysts. It is the intent of this document to put equal emphasis of importance on the error and order variables. Figure 5.2.4 illustrates the percent error determined throughout the domain for the axial velocity. This is the error for the fine grid solution (i.e. 201 by 41 grid points). It is instructional to examine the same quantity for the 101 by 21 grid (see Fig. 5.2.5). The error values are based on the Richardson extrapolated estimate of the exact solution. Over the preponderance of the domain, fine grid error is very small. Further, this representation shows that even in regions

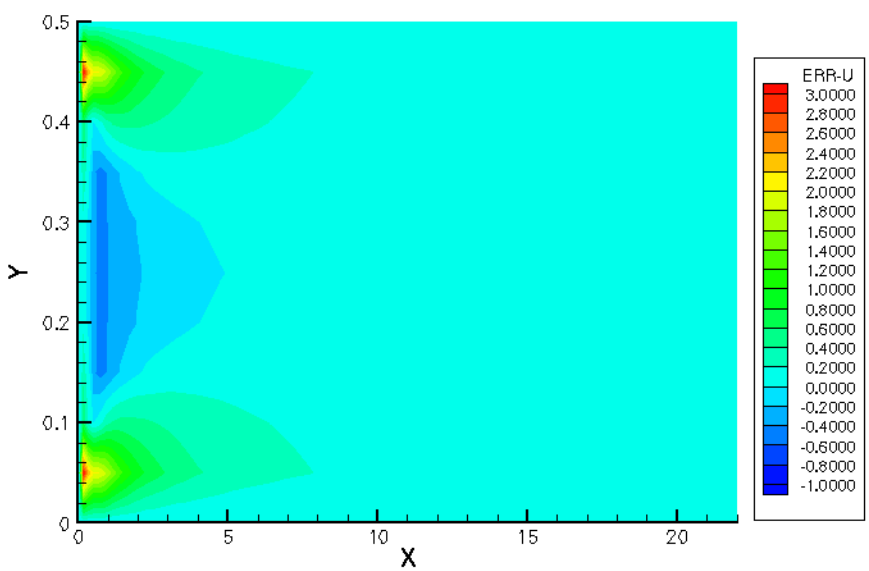

Figure 5.2.4 -CFD-ACE Fine Grid Axial Velocity Error

where order was deemed to be zero, the error is generally less than one percent. Figure 5.2.5 shows that the error for the medium grid solution has grown perceptibly.

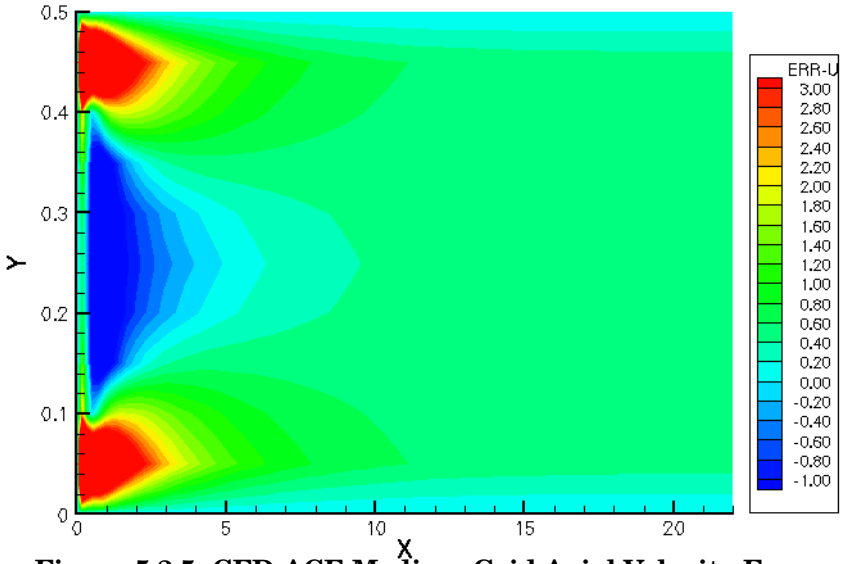

Figure 5.2.5 -CFD-ACE Medium Grid Axial Velocity Error

Examination of the illustration shows the errors are 4 times larger than the fine grid solution. This is a mathematical requirement in the areas where grid refinement has attained second order asymptotic behavior (just as Blottner's comparison of coarse and fine grid solutions required an error ratio of 16).

5.3 SACCARA Compressible Structured Grid Convergence Analysis (Chris Roy)

An axisymmetric nozzle flow was evaluated with the structured finite volume CFD code SACCARA. The inviscid Euler equations were solved using the ideal gas equation of state. Sample Mach number contours are shown in Figure 5.3.1. The flow is slightly supersonic at the inflow (left) and expands to roughly Mach 6 at the exit (right). This problem was solved numerically using five different grid levels. The finest level employed 400x160 grid cells, while the coarsest was 25x10 cells. Grid halving was employed to obtain the coarser meshes from the finest mesh.

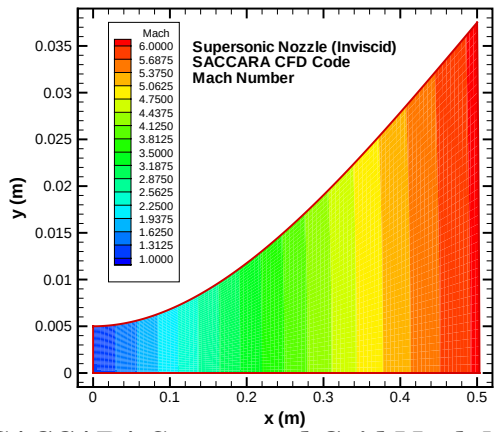

Figure 5.3.1 - SACCARA Structured Grid Mach Number Contours

Various norms of the discretization error in axial velocity are shown in Figure 5.3.2 for the five mesh levels. The normalized dimensionless length parameter h is a

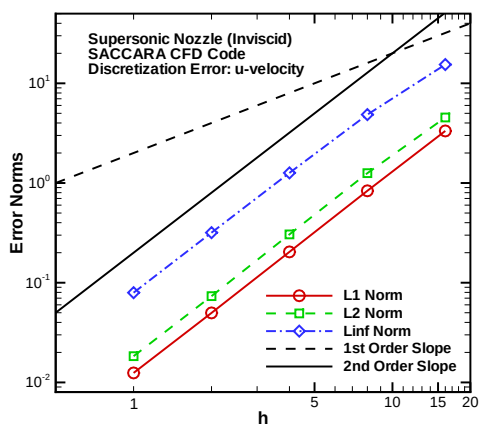

Figure 5.3.2 - SACCARA Structured Grid u-Velocity Error Norms

measure of the grid size (e.g., Δx) and is plotted logarithmically along the abscissa. This parameter is normalized such that h=1 for the fine grid and h=16 for the coarsest grid. The local discretization error for the axial component of velocity is simply the discrete solution minus the exact solution (i.e., $u_{discrete}$-u_{exact}), where the exact solution is estimated using Richardson extrapolation. Discrete L_1 and L_2 norms were calculated by the following equation where *k* indicates the mesh level and *n* is summed over the *N* points used in the norm calculation.

$$\mathrm{L}_1\ \mathrm{norm}_k = \frac{\sum_{n=1}^{N} \left| u_{k,n} - \tilde{u}_{exact,n} \right|}{N}$$

$$\mathrm{L}_2\ \mathrm{norm}_k = \left(\frac{\sum_{n=1}^{N} \left| u_{k,n} - \tilde{u}_{exact,n} \right|^2}{N} \right)^{1/2}$$

In this case, all of the nodal mesh points were used including boundary points.The norms of the discretization error for u-velocity are shown on the ordinate, again on a log scale. The first and second order slopes (dashed and solid lines, respectively) are also shown for reference. All of the error norms display the expected second-order slope on the finer meshes, thus verifying the solution are second-order accurate in space.

The local order of accuracy for the axial component of velocity is presented in Figure 5.3.3 using the three finest meshes. The order of accuracy is approximately

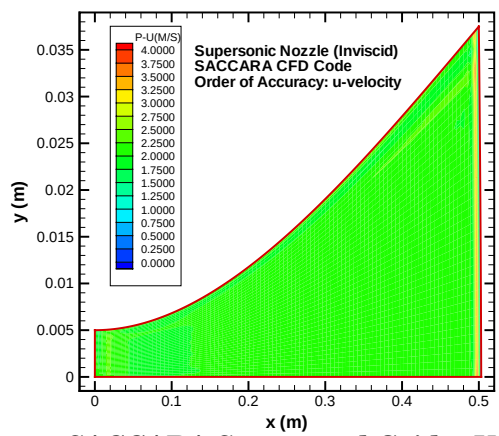

Figure 5.3.3 -SACCARA Structured Grid u-Velocity Order

two everywhere except at the outflow boundary, where errors are introduced due to the fact that the properties here are determined via grid-aligned extrapolation from the interior rather than streamline-based extrapolation. Due to the supersonic nature of the flow, there errors do not influence the solution within the domain.

The local discretization error in the axial velocity is shown in Figure 5.3.4 for the finest mesh. The discretization error is a percentage of the exact (Richardson extrapolated) solution. The errors are largest near the inflow, but the magnitude of the errors is small with a maximum discretization error of approximately 0.03%.

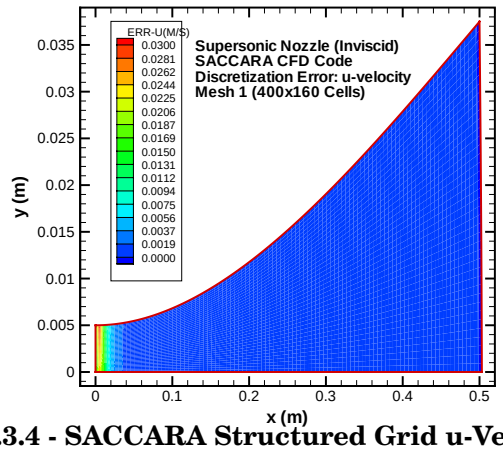

Figure 5.3.4 - SACCARA Structured Grid u-Velocity Error

5.4 Premo Compressible Unstructured Grid Convergence Analysis (Chris Roy)

A two-dimensional Manufactured Solution was solved using the Premo/SIERRA unstructured, finite-volume CFD code. The Manufactured Solution methodology is a code verification procedure used to find coding mistakes (see Reference 23 for details). The Manufactured Solution was comprised of sinusoidal functions of the x and y spatial coordinates and was applied to the inviscid Euler governing equations using the ideal gas assumption. The solution for the mass density is presented below in Figure 5.4.1. Similar solutions were specified for the two velocity components (which

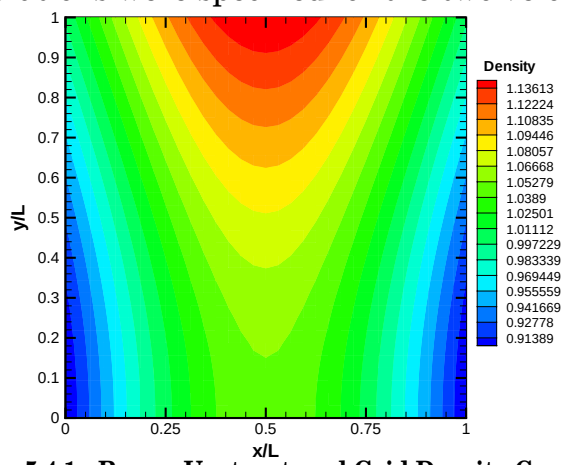

Figure 5.4.1 - Premo Unstructured Grid Density Contours

were chosen to be everywhere supersonic) and the pressure. The Manufactured Solution was solved on five meshes: 9x9, 17x17, 33x33, 65x65, and 129x129 nodes.

Discretization error norms for the mass density relative to the exact manufactured solution are presented in Figure 5.4.2 for the five mesh levels. In this case, the

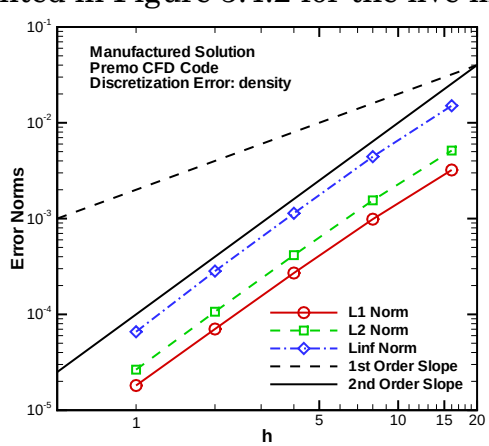

Figure 5.4.2 - Premo Unstructured Grid Density Error Norms

discretization error is found by taking the difference between the discrete solution and the exact (Manufactured) solution. The L_1, L_2, and $L_{infinity}$ (max) norms are shown on the log-log plot versus the normalized grid spacing. The expected second order slope (solid line) is matched for all three norms on the finer meshes, thus verifying that the spatial accuracy is second order.

The local order of accuracy for the density is presented in Figure 5.4.3 using the three finest meshes. The order of accuracy is approximately two throughout the do-

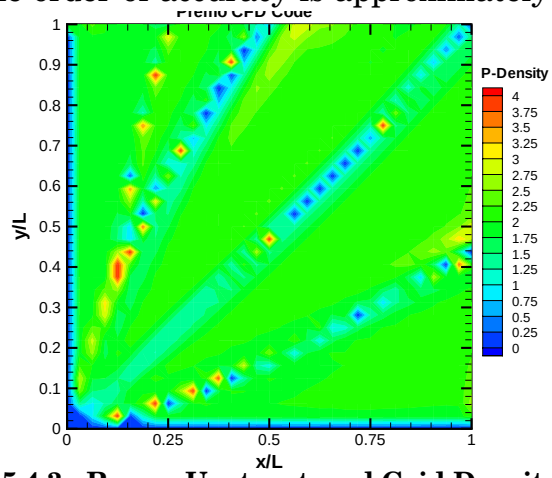

Figure 5.4.3 - Premo Unstructured Grid Density Order

main, with deviations occurring at the inflow boundaries (bottom and left) and along three lines originating from the origin. Dirichlet values were specified at the inflow boundary, thus exactly giving zero discretization error. VIVID produces a zero for the order of accuracy in this case. In order to determine the source of the deviations from second order within the domain, the local discretization error is examined.

The local discretization error in the density for the fine mesh is presented in Figure 5.4.4. There is some error generated at the lower left-hand corner of the do-

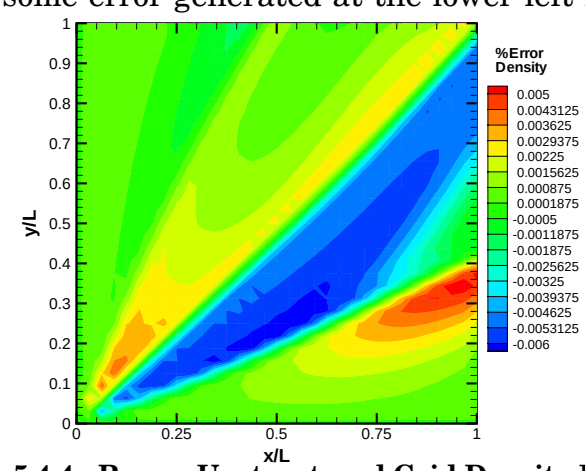

Figure 5.4.4 - Premo Unstructured Grid Density Error

main. This is not surprising since the horizontal and vertical components are approximately equal, leading to velocity vectors that lie along the diagonals (i.e., not grid aligned). This error appears to convect downstream along the diagonal, as well as propagate along the local supersonic characteristics, which in this case are approximately ±30 degrees to the diagonal. The locations where the local discretization error crosses over from positive to negative denotes a location where the order of accuracy evaluation becomes undefined (see Figure 5.4.3), as demonstrated in Section 5.1. It should be noted that the magnitude of the density error on the fine mesh is small, with a maximum of 0.006%.

5.5 Fuego Flat Duct Unstructured Grid Convergence Analysis (Amalia Black)

A two-dimensional flat duct incompressible flow problem similar to that evaluated by CFD-ACE was analyzed with the unstructured grid finite volume Fuego numerical program (see Fig. 5.5.1). VIVID has been used to determine the solution order and error for two dependent variables (axial velocity and pressure) in the 2-D flat duct problem. The flat duct problem is an incompressible laminar flow problem which was simulated with the Fuego code on three non-uniform meshes: coarse (101x21x2) with 2000 elements; medium (201x41x2) with 8000 elements; and fine (401x81x2) with 32000 elements. Since Fuego is a 3-D code, two symmetry planes were specified in the third coordinate direction to simulate a 2-D domain. Exact analytical solutions do not exist for the developing duct flow problem; therefore, Richardson extrapolation method was used to estimate the exact solution.

Problem Description:

The 2-D flat duct geometry is 10 m long by 0.5 m high. Figure 5.5.1 shows the solution for the axial velocity. This problem is similar (different inlet conditions & properties) to the problem demonstrated for CFD-ACE in the second example in Section 5.2. A constant velocity of 1.0 m/s was specified at the inlet and a constant pressure of 0.0 Pa was specified at the outlet. The density was 1 kg/m^3, the molecular viscosity was 0.002 kg/m-s, and the corresponding Reynolds number was 250 based on duct height. The entrance length was estimated to be 5.25 m such that fully developed flow exists over roughly half of the duct. In this region, the axial velocity profile is parabolic across the duct. At the inlet, the solution is singular at the two corners where the inlet intersects the walls.

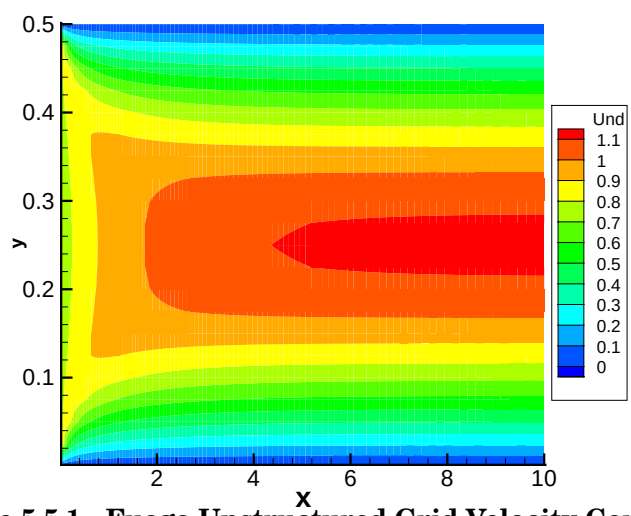

Figure 5.5.1 - Fuego Unstructured Grid Velocity Contours

Flat Duct Incompressible Flow Solution Analysis:

Each double precision simulation was run until a steady state solution was obtained and all residuals were reduced below 1.0e-18. The calculations were performed using the turbulence mechanics in Fuego stable version 0.7.0. The turbulence terms were set to zero to simulate laminar flow. The modified linear profile skew upwind (LPS) convection scheme was used in the Fuego simulation. Expected order of accuracy is between 1.0 and 2.0 based on the Peclet number of the flow. A second order accurate scheme was assumed in the Richardson extrapolated solution.

Figure 5.5.2 shows the order of accuracy contours for the fine mesh solution axial velocity variable throughout the duct (illustrated on the course mesh nodes). It

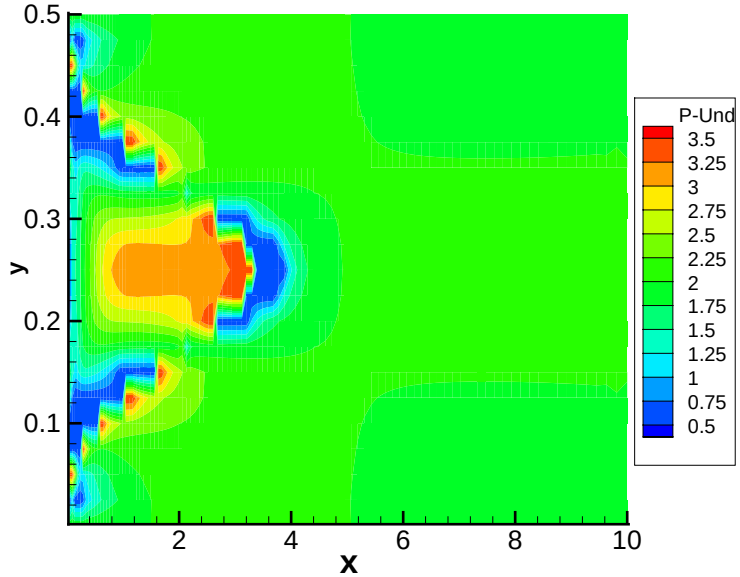

Figure 5.5.2 - Fuego Unstructured Grid Axial Velocity Order

should be noted that the y-axis has been expanded independently of the x-axis. The order pattern produced by VIVID for Fuego is similar to the contour profiles produced for CFD-ACE for a similar flat duct problem (see Fig. 5.2.2 in Section 5.2). A reduction in the lower order regions are seen for the current evaluation. This may be the result of the finer grids used in the Fuego numerical solutions. Twice as many grid points were used on a shorter domain in both the x and y directions. This makes average spacing between grid points one-fourth what it was in the CFD-ACE problem. As expected, variable order values are shown in the entrance region of the duct. A much finer mesh is required near the singularities to capture the gradients in this region. In the fully developed region, the order values are closer to 2.0. A second order scheme should capture a parabolic velocity profile exactly. The calculated discretization error between the Fuego fine solution and the Richardson extrapolated

estimate is shown in Figure 5.5.3. In areas of negative discretization error values in

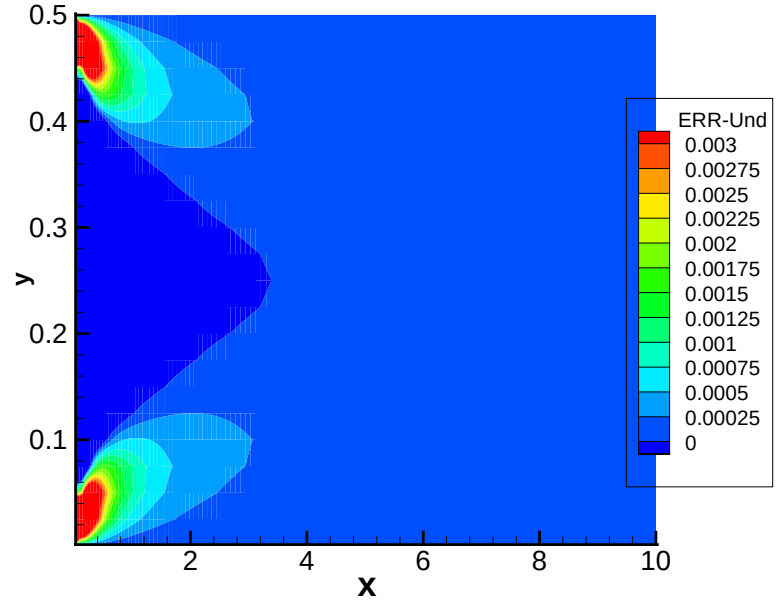

Figure 5.5.3 - Fuego Unstructured Grid Axial Velocity Error

the entrance region of the duct and at the dividing line (between positive and negative error values) where the error values are zero, the extrapolation method will fail (see Figure 5.5.2).

In addition to local order values at discrete nodes, it is important to look at the order of accuracy value based on integral variables. In Figure 5.5.4, the axial velocity L1, L2, and L_∞ norms output from VIVID are shown as a function of grid cells in the x-direction. The slope of the L1 norm curve appears to be approaching second order

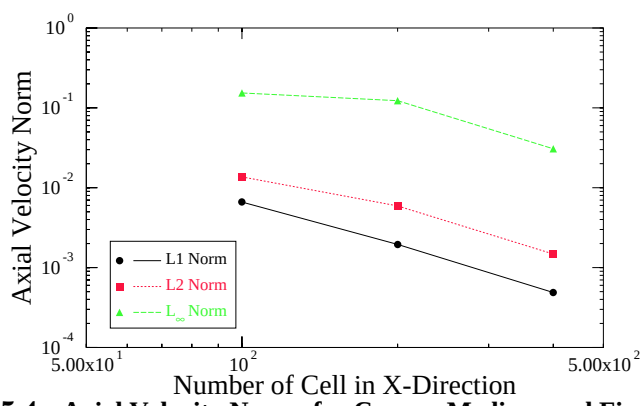

Figure 5.5.4 - Axial Velocity Norms for Coarse, Medium and Fine Mesh Solutions

accuracy. It is difficult to attain a value of 2 due to the lower order region near the inlet. Further grid refinement is required to approach second order behavior over the entire domain.

Figure 5.5.5 shows the order of accuracy plot for the pressure along the duct.

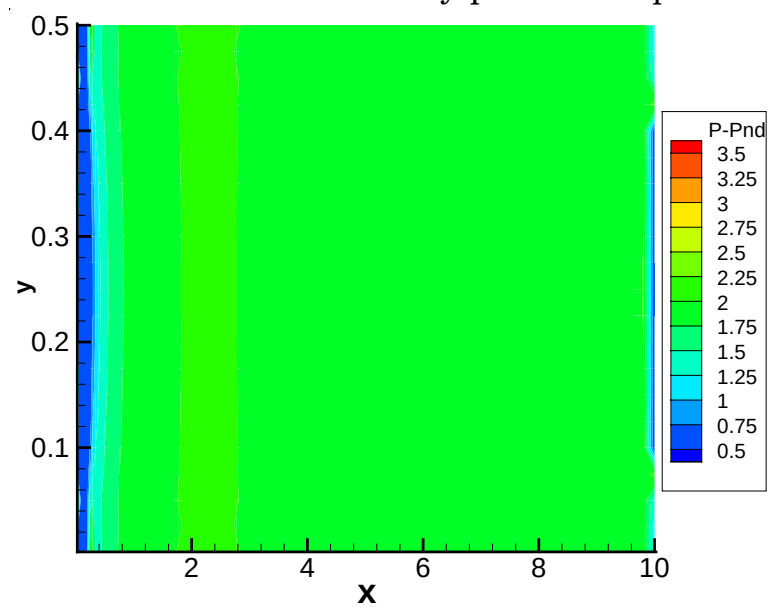

Figure 5.5.5 - Fuego Unstructured Grid Pressure Order

Second order accuracy values appear over most of the duct, except near the inlet where the order values drop to one or less. The calculated discretization error between the Fuego fine mesh solution and the Richardson extrapolated solution is shown in Figure 5.5.6. Again, the errors are on the order of 10^{-4} beyond the entrance region.

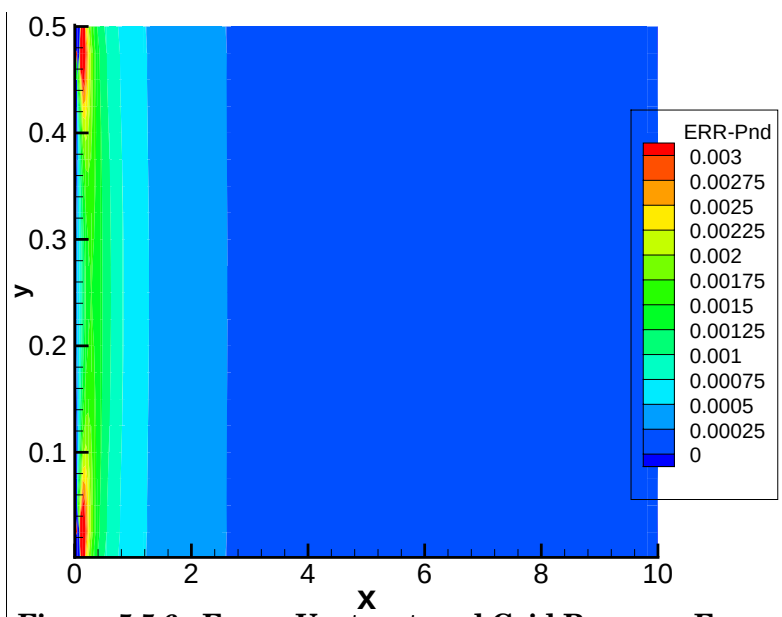

Figure 5.5.6 - Fuego Unstructured Grid Pressure Error

In Figure 5.5.7 L1, L2, and L_∞ norms for pressure are shown as a function of the grid cells in the x-direction. The slope of the norm curves suggests that the solution is converging at a first order rate. Second order convergence was expected for the pressure variable. This value was used in the Richardson extrapolation. The global order value is largely influenced by the low order values near the inlet of the

duct. Finally, the change in slope of the norm curves suggests that the coarse mesh solution is not in the asymptotic region of convergence. Finer meshes are required to accurately capture the developing flow region in the duct.

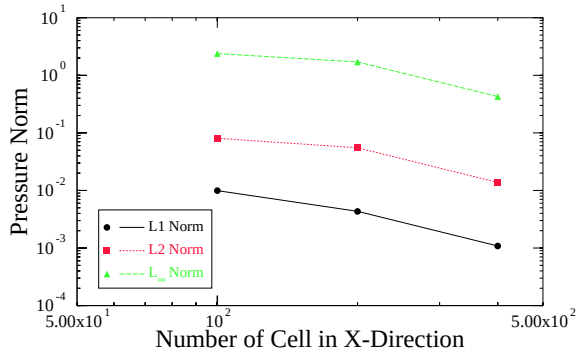

Figure 5.5.7 - Pressure Norms for Coarse, Medium and Fine Mesh Solutions

Overall, the VIVID code has provided the solution order and solution error based on the Richardson extrapolation method. The results indicate that the numerical scheme is first order in the developing flow region and second order in the fully developed region of the duct. A finer mesh is required to accurately capture the gradients in the developing flow region. An additional mesh would also provide a better understanding of the behavior of the order as the mesh continues to be refined.

The 2-D flat duct problem is not ideal for code verification due to the singularities at the inlet-wall locations. In addition, the discretization error in the axial velocity is zero at some locations and Richardson extrapolation does not work at these points. In general, the calculated order of a numerical scheme is very sensitive to the properties of the problem solution (i.e. singularities), numerical error behavior (i.e. zero values), and the computational mesh (i.e. non-uniform spacing).

5.6 Calore Conduction Unstructured Grid Convergence Analysis (Donald Potter)

An example of a two-dimensional class of problems addressed by the unstructured finite element Calore program is included in the sample analyses in this section. The steady state thermal response solution of an infinitely long rectangular cross section rod was evaluated. This is the two-dimensional heat conduction problem discussed in Section 4.4.2. Steady state solutions were obtained with 200 (11x11x2 nodes), 800 (21x21x2 nodes), and 3200 (41x41x2 nodes) elements (i.e. grid refinement factor of 2 in both directions). Since Calore is a 3-D code, two symmetry planes were specified in the third coordinate direction to simulate a 2-D domain. Ben Blackwell of department 9133 provided the simulations.

Problem Description:

The sixth sample problem demonstrates VIVID analysis of heat conduction solutions in a two-dimensional rectangular infinitely long rod: with horizontal (L) and vertical (H) dimensions 0.10 and 0.05 m respectively. As defined in Section 4, this problem used boundary conditions of constant heat flux on the left (i.e. x=0), constant temperature on the right (i.e. x=L), adiabatic at the bottom (i.e. y=0), and convective heat flux at the top (i.e. y=H). A uniform heat generation term was also included. The internal heat generation term was constant throughout the spatial domain with a value of 1.353×10^5 W/m^3. Boundary condition parameters specified for this problem are defined as follows:

$$\dot{q}_{x_o} = 3500 \ W/cm^2 \qquad T_{x_L} = 1000 \ K \qquad \dot{q}_{y_o} = 0 \ W/cm^2$$

$$T_\infty = 25 \ K \qquad h_c = 60 \ W/(m^2 K)$$

The constant specified values of the parameters employed in this simulation are as follows:

$$\rho = 7000 \ kg/m^3 \qquad c_p = 500 \ J/kg-K \qquad H = 0.05 \ m$$

$$L = 0.10 \ m \qquad k = 0.40 \ W/m-K$$

$$q_{gen} = 1.353 \times 10^5 \ W/m^3$$

Temperature field contours for the fine grid solution values on the coarse grid point

locations are shown in Figure 5.6.1 Dimensional proportionality for the x and y coor-

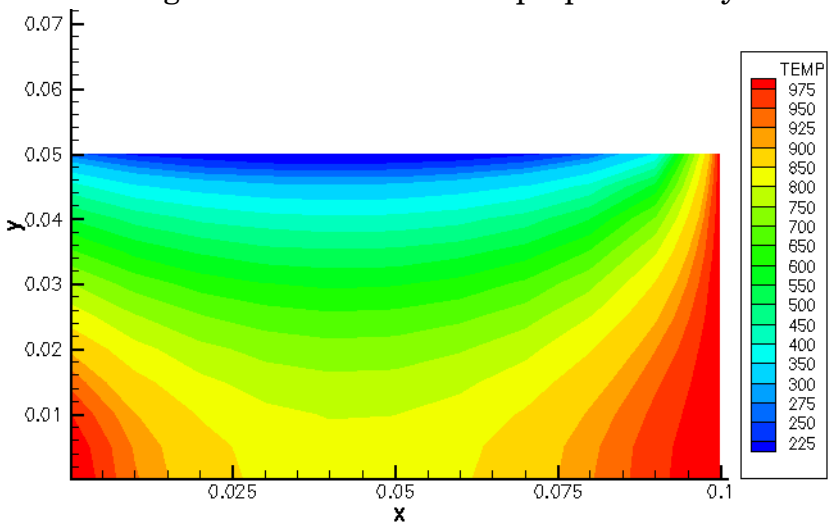

Figure 5.6.1 - Calore Unstructured Grid Temperature Contours

dinates was maintained in this illustration.

Two-Dimensional Heat Conduction Solution Analysis:

Order and error analyses based on both the estimated exact and an analytical solution are demonstrated in this section. Order of accuracy, based on the Richardson estimate of the exact solution, obtained over the domain for the temperature is shown in Figure 5.6.2. Calore showed a predominantly second order solution over

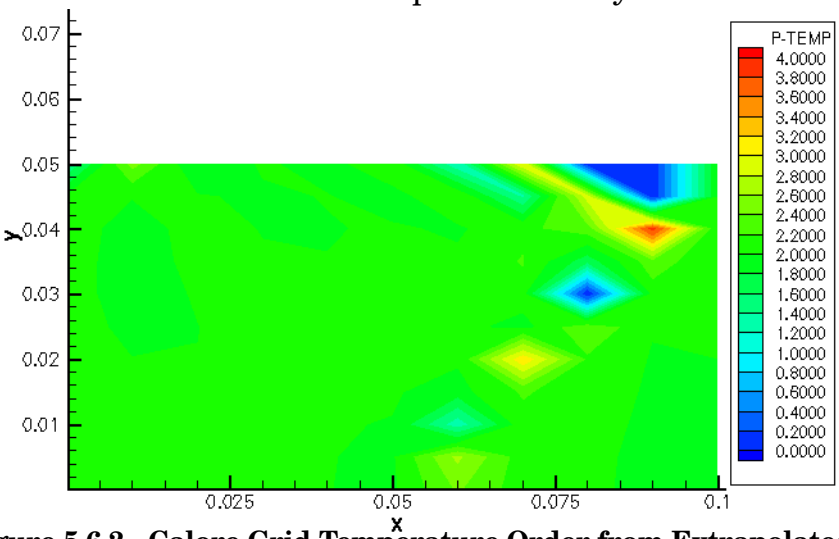

Figure 5.6.2 - Calore Grid Temperature Order from Extrapolated

the majority of the domain. Two small regions of lower order behavior are observed. One of these regions is in proximity to the upper wall convective boundary condition. This could possibly be the result of the heat flux discontinuity in the upper right hand corner of the cross section. Small regions of higher order behavior caused by multi-dimensional refinement error cancellation also appear.

Error analysis of the two-dimensional conduction solution was performed. Temperature percent error determined over the domain is illustrated in Figure 5.6.3. As

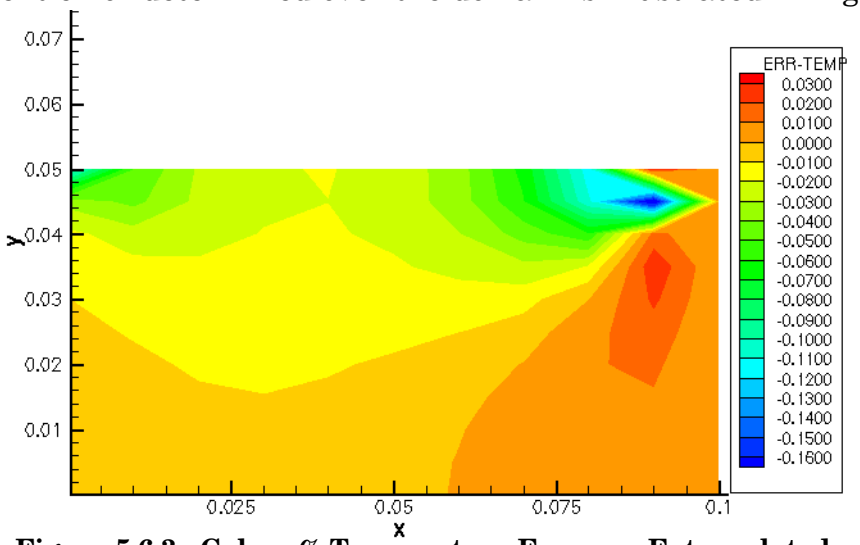

Figure 5.6.3 - Calore % Temperature Error vs. Extrapolated

in the fluid analyses, error values are based on the Richardson extrapolated estimate of the exact solution. The error is very small. Maximum absolute error observed occurs along the convective boundary and is less than 0.2%. For comparison purposes, error based on an exact analytical solution of the conduction problem is shown in Figure 5.6.4. The scales on the two error illustrations were kept the same for ease of

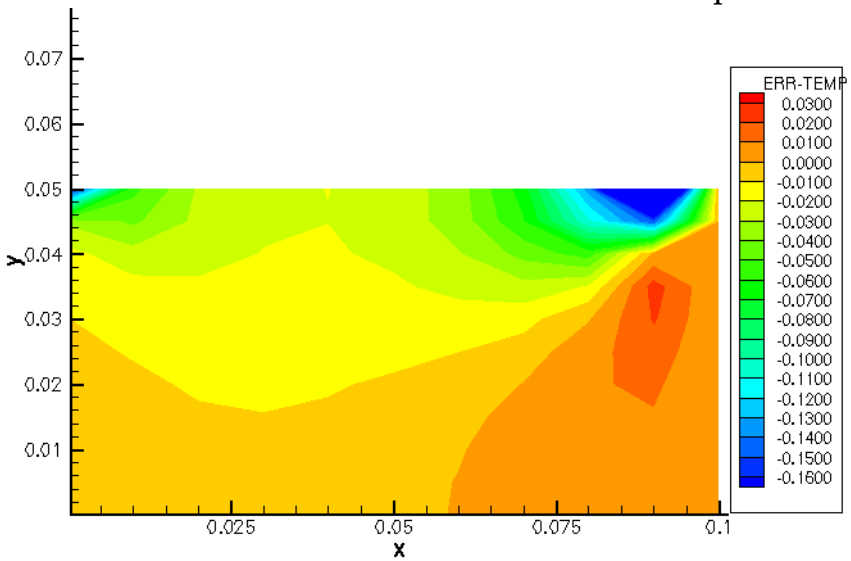

Figure 5.6.4 - Calore Temperature Error vs. Analytical

comparison. Error estimates using the analytical solution are similar to those determined with the extrapolated estimate. Slightly more error is calculated relative to the exact analytical solution along the convective boundary.

In addition to the error calculation illustration with the analytical reference,

the order of accuracy using this reference is shown in Figure 5.6.5. As expected, the

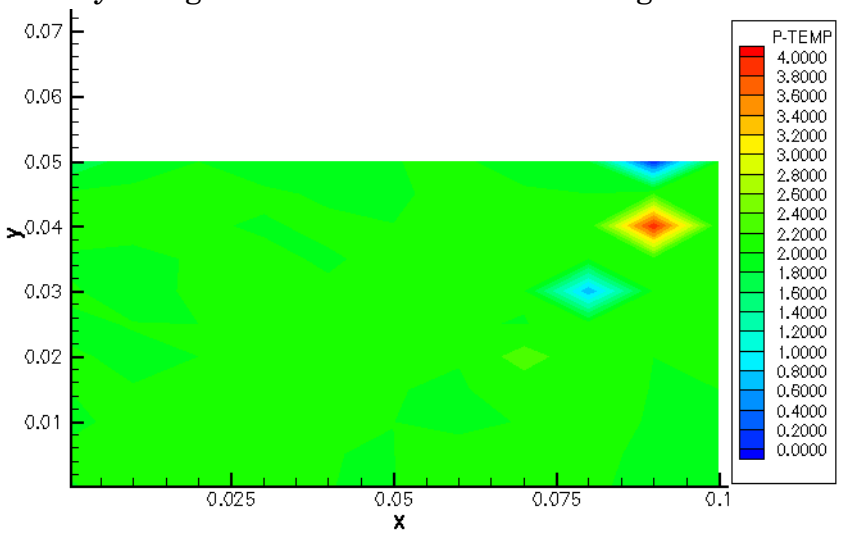

Figure 5.6.5 - Calore Temperature Order from Analytical

results are very similar to those obtained with the extrapolated reference. Non-second order results were reduced even further than in the first examination. Regions of irregular order occurred in the same physical locations as before. Obtaining similar results for both approaches supports the validity of the extrapolation approach when an exact analytical reference is not available for more complex problems.

5.7 FCV-FAM Conduction Structured Grid Convergence Analysis (Donald Potter)

Analysis of the sample problem examined in the previous section was repeated with FCV-FAM, a structured finite volume numerical code written by the author for the explicit purpose of verification research. This code is based on the Face Assembly Method (FAM) for producing the difference equations approximating the governing partial differential equations governing. A limited discussion of the details of this code is contained in Appendix E. Steady state thermal response two-dimensional heat conduction solutions of the infinitely long rectangular rod were again evaluated relative to the Richardson extrapolated exact estimate and analytical references. Solutions were obtained with nominal meshes with 100 (11x11 nodes), 400 (21x21 nodes), and 1600 (41x41 nodes) cells (i.e. grid refinement factor of 2 in both directions).

Problem Description:

Problem description specifics are the same as before and are only partially repeated here. Sample seven heat conduction solutions are for a two-dimensional rectangular infinitely long rod: with horizontal (L) and vertical (H) dimensions 0.10 and 0.05 m respectively. Boundary conditions are as follows: constant heat flux on the left (i.e. x=0), constant temperature on the right (i.e. x=L), adiabatic at the bottom (i.e. y=0), and constant convection coefficient and ambient temperature at the top (i.e. y=H). Heat generation was also included. Internal heat generation was constant throughout the spatial domain at 1.353×10^5 W/m^3. Boundary condition parameters and simulation constant parameter values can be obtained from the problem description in Section 5.6. Temperature field contours for the fine mesh solution are shown in Figure 5.7.1 Dimensional proportionality for the x and y coordinates was

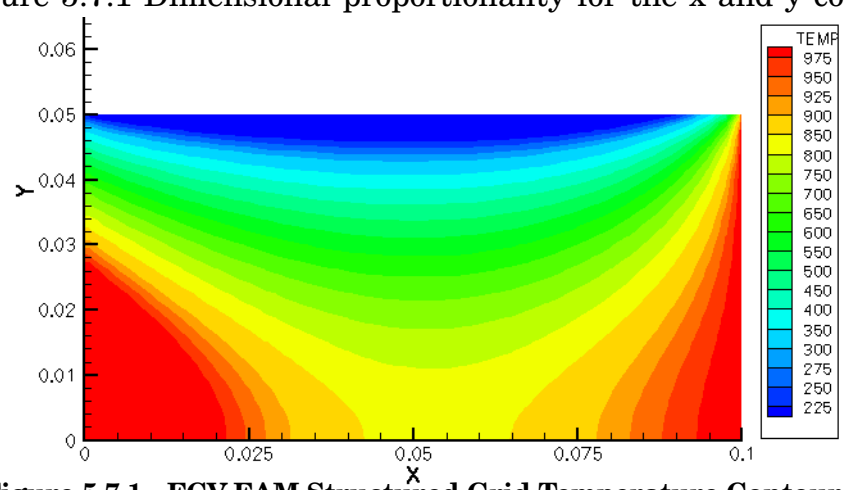

Figure 5.7.1 - FCV-FAM Structured Grid Temperature Contours

maintained in this illustration. The contour scale for this figure is the same as used in Figure 5.6.1 for the Calore solution. Visual comparison of the two solutions shows some significant differences. This will be used to illustrate a potential pitfall of the Richardson extrapolation analysis approach in the next section.

Two-Dimensional Heat Conduction Solution Analysis:

The order of accuracy, based on the Richardson estimate of the exact solution, obtained over the domain for the temperature is shown in Figure 5.7.2 using the same scale as in the previous section. FCV-FAM showed an order of accuracy some-

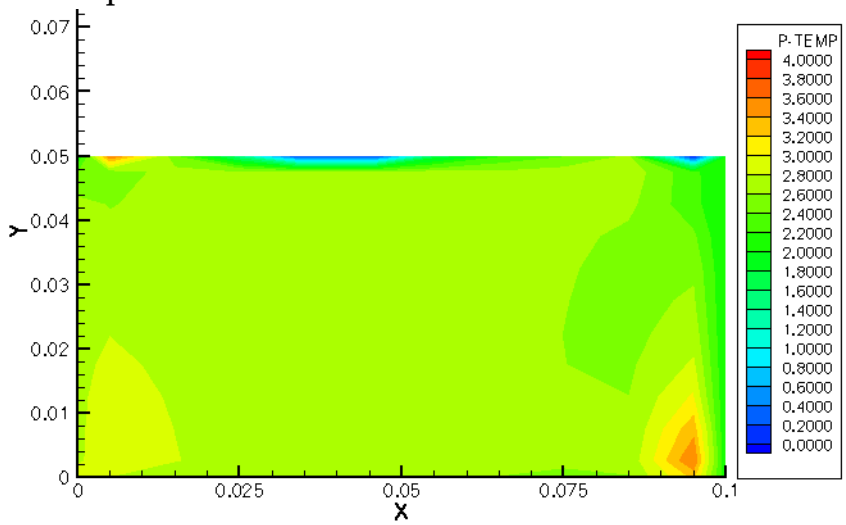

Figure 5.7.2 - FCV-FAM Temperature Order from Extrapolated

what greater than two over the majority of the domain. A global order value of 2.6608 was calculated, compared to 1.8877 for the Calore example. This gives the impression of a more computationally accurate algorithm. Both methods are second order, but have different error coefficients for this problem. As in the previous section, a few small anomalous regions of lower and higher order were present.

Temperature error determined over the domain is illustrated in Figure 5.7.3 for the FAM solution. Like the Calore analysis, error values were based on the Richard-

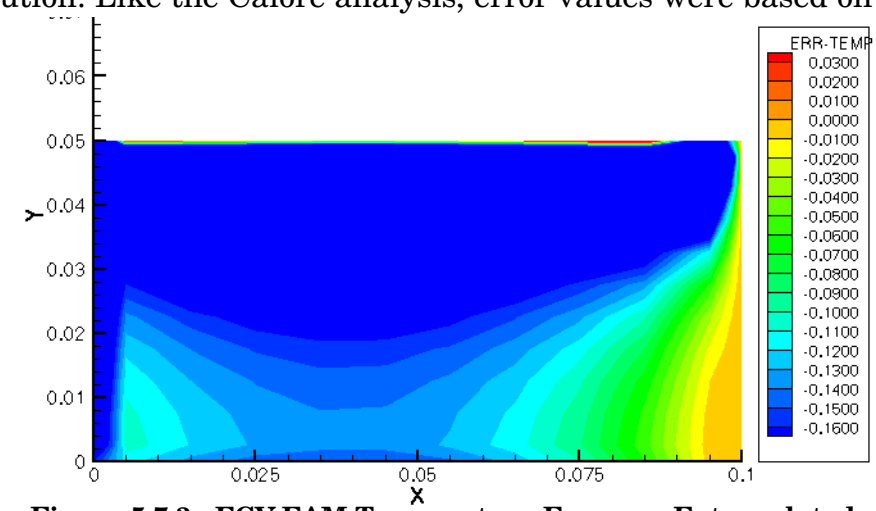

Figure 5.7.3 - FCV-FAM Temperature Error vs. Extrapolated

son estimate of the exact solution. Again, the errors are small, but as illustrated by the larger region of blue contours (common contour scale), absolute values of error for this algorithm are larger (maximum 0.7%) than for Calore. As before, error based on an exact analytical solution of the conduction problem was also determined as a

further check (see Fig 5.7.4). The scale was allowed to float to highlight the difference. Error estimates using the analytical solution are greater than those determined from the extrapolated estimate. They are in excess of 35%. This is the pitfall mentioned in the problem description. Solution error was greatly underestimated by the extrapolation method.

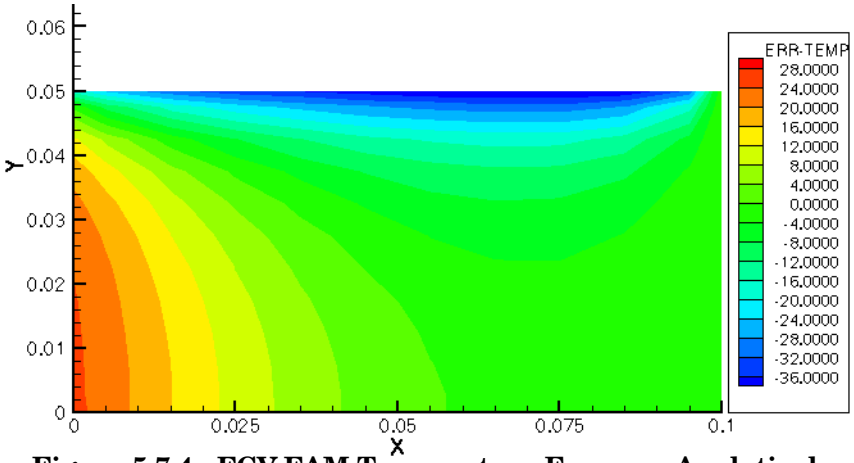

Figure 5.7.4 - FCV-FAM Temperature Error vs. Analytical

In addition to determining error with the analytical reference, the order of accuracy using this reference is shown in Figure 5.7.5. Despite the large errors noted in

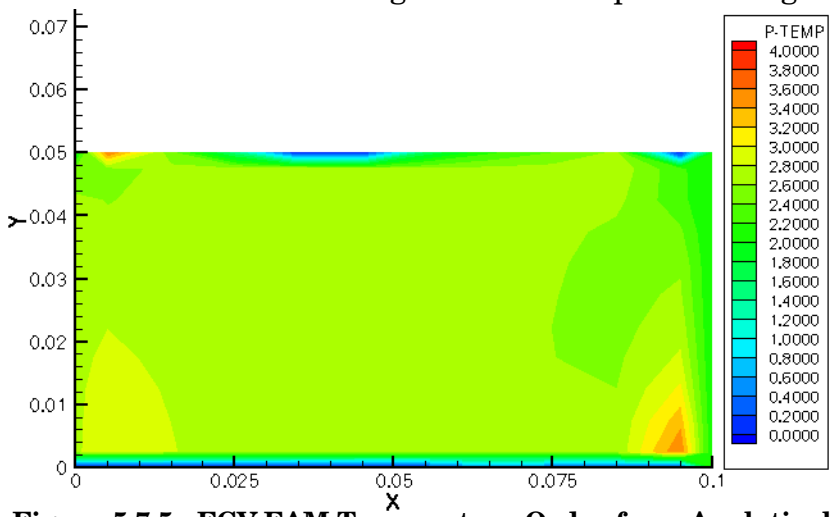

Figure 5.7.5 - FCV-FAM Temperature Order from Analytical

the previous figure, order representation was still above two in the majority of the domain. However, indications of a problem occur along the upper and lower boundary where reduced order of the solution is indicated. These coincide with the regions of large error in the solution. A possible inference of this is that the interior discretization scheme is reasonable, but there is a problem with the boundary condition implemetation. There was a bug in the boundary conditions of this code. VIVID results pointed to this fact and this case demonstrates how the tool can be of value locating such problems. ***This example highlights the precarious nature of relying on Richardson extrapolation as the sole indication of the quality of a solution***.

5.8 Fuego Transient Unstructured Grid Convergence Analysis (Donald Potter)

An example of a convergence study of transient FCV/Fuego solutions is presented in this section. This was an early developmental version of the code and is **NOT** representative of any current tool. The purpose of this example is to demonstrate the transient analysis capability of VIVID. Specific boundary conditions and problem definition details are intentionally omitted due to the experimental nature of the computational tool used. Solutions were generated by Mary McWherter-Payne. Transient order of an incompressible two-dimensional flat duct was evaluated. Solutions were obtained for a mesh with 2000 cells. Each of the three solutions was ran transiently to a final time of 0.012 seconds. The time step for each successive solution was reduced by a factor of two. A backward Euler first order accurate transient solver was used in these solutions.

The axial velocity solution is illustrated in Figure 5.8.1 below. Time is treated as

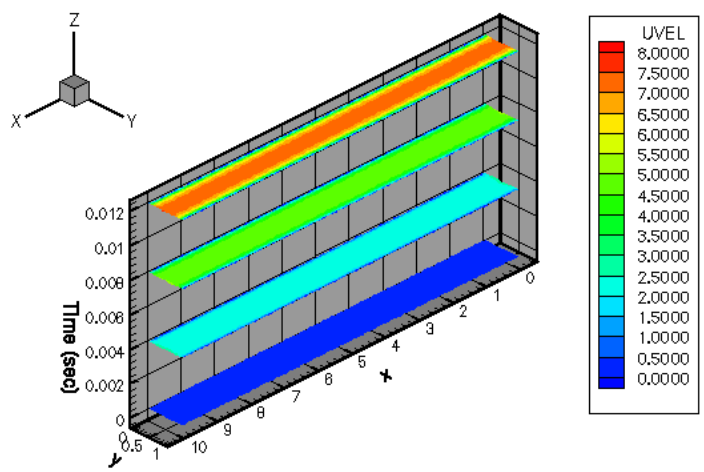

Figure 5.8.1 - FCV Transient Velocity Contours

the third dimension in this process. Order of accuracy was obtained over the domain at each of these time planes for axial velocity with VIVID (see Fig. 5.8.2). In this

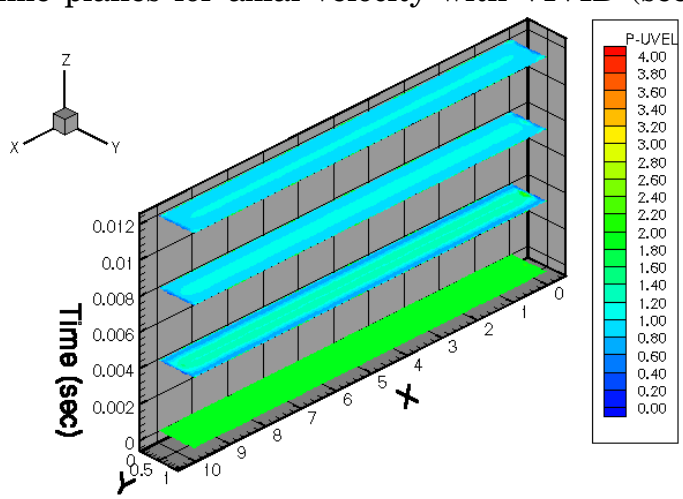

Figure 5.8.2 - FCV Transient Velocity Order

case, the third (i.e. time) dimension was refined by a factor of two between solutions. Only the temporal dimension was refined, x and y geometrical nodal dimensions remained constant. Constant pressure inlet and exit boundary conditions were applied on the ends of the duct at time zero. Contrary to the steady state two-dimensional flat duct examples, flow in this illustration is from right to left. FCV/Fuego demonstrated a variable order of accuracy with time. At the initial time plane, second order was shown everywhere by VIVID definition since the initial condition is the same for all three solutions. For the solutions obtained at a time of 0.004 seconds, a predominantly second order time accurate behavior is attained. Second order behavior occurs at this and all time planes for the solutions as they approach the fixed wall no slip boundary conditions. Second order behavior by definition is also observed at the inlet and exit boundaries. Subsequent time planes of 0.008 and 0.012 seconds demonstrated accuracy behavior tending to first order. This is expected since a backward Euler first order solver is used in this transient solution. The region of higher order behavior decreases with time. This is to be expected as the solution tends toward a steady state with fixed boundary conditions. With the same spatial grid, at a time of infinity, all solutions will ultimately reach the same end point culminating in a second order representation by VIVID definition.

Error analysis for the transient flat duct finite volume solution was performed. Axial velocity error in percentage terms determined throughout the time variant domain is illustrated in Figure 5.8.3. As in previous analyses, error values are based on

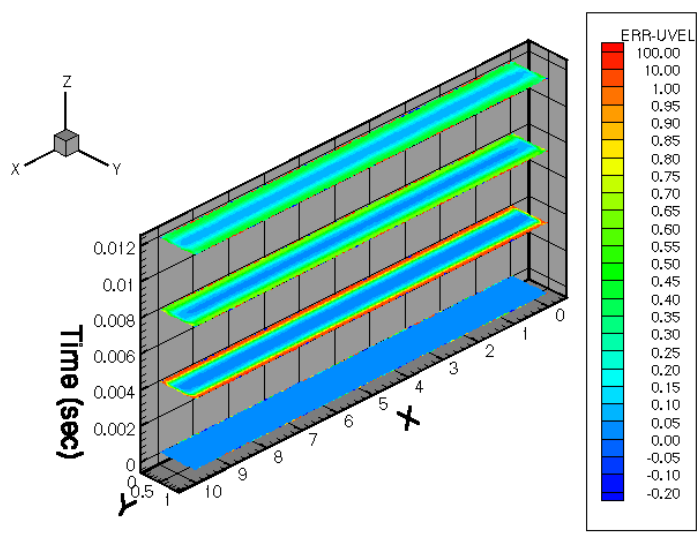

Figure 5.8.3 - FCV Transient Velocity Error vs. Extrapolated

the Richardson extrapolated estimate of the exact solution. Error over the majority of the solution is less than one percent. However, there was a significant region where the error in predicted velocity is excessive. For the most part, the large error region is restricted to the early time plane. This combined with a correspondingly larger region of reduced order determined in the order analysis suggests that a more refined spatial grid may be warranted to resolve the earlier time planes.

5.9 Calore 3-Dimension Unstructured Grid Convergence Analysis (Donald Potter)

Finally, a three-dimensional convergence study for steady state finite element Calore solutions is presented in this section. The order of accuracy of a solid cube in-depth conduction problem was evaluated. Solutions were obtained with grids of 250, 2000, and 16000 elements respectively. The solutions were obtained by Bruce Bainbridge of department 9116. These solutions had a grid refinement factor of two; producing eight times the elements in successive runs. Boundary conditions for the conduction problem are illustrated in Figure 5.9.1. They included convection, fixed

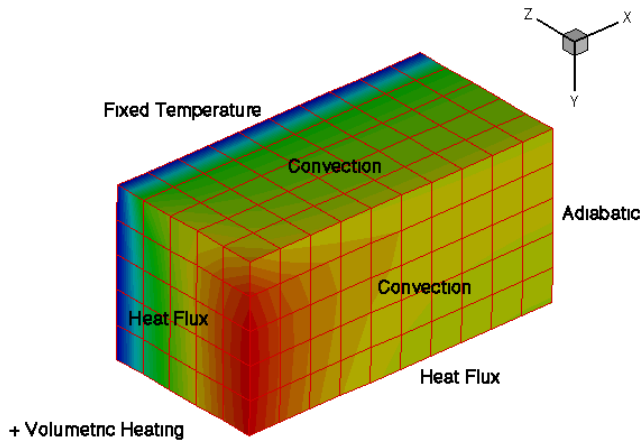

Figure 5.9.1 - Calore 3-D Example Boundary Conditions

heat flux, and fixed temperature.

Order of accuracy was determined over the domain for the temperature solution shown in Figure 5.9.2. Order of accuracy obtained for the fine grid over the domain

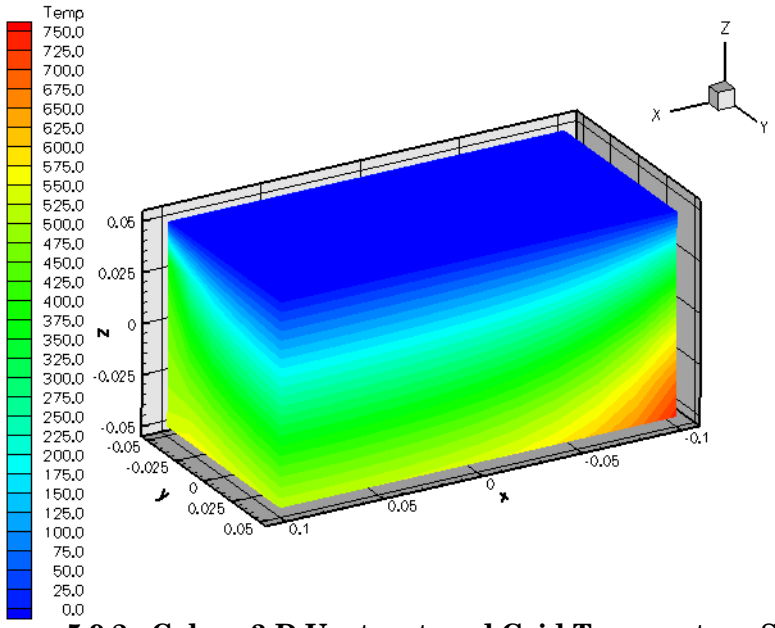

Figure 5.9.2 - Calore 3-D Unstructured Grid Temperature Solution

for the steady state solution of temperature is shown in Figure 5.9.3. Three types of boundary conditions were applied to the cube walls. In addition, a volumetric source term was included. A predominantly second order accurate solution is obtained over

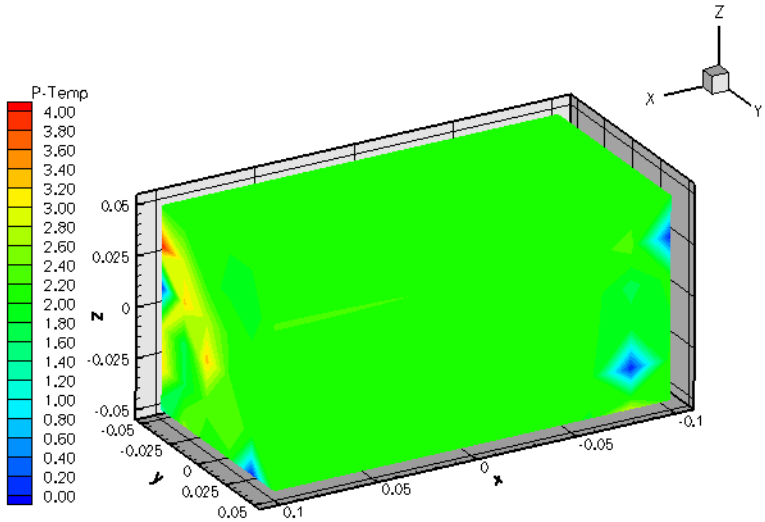

Figure 5.9.3 - Calore 3-D Unstructured Temperature Order

the domain. To aid in visualization of the three-dimensional order, solution slice planes at x, y, and z of zero are illustrated in Figure 5.9.4. As seen earlier, second order behavior occurs by definition as the solutions approach the fixed boundary conditions. Some lower order behavior is observed away from the boundaries. This is likely the result of insufficient grid resolution. Isolated regions of higher order behavior possibly caused by multi-dimensional refinement error cancellation are also apparent in the solution domain.

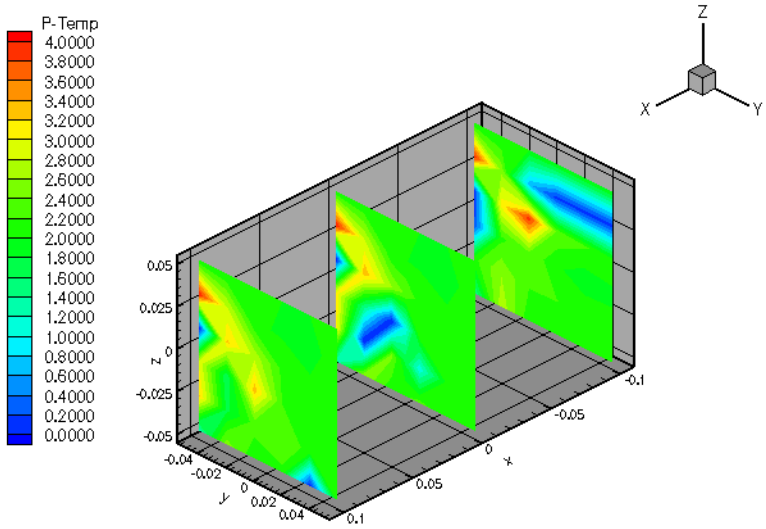

Figure 5.9.4 - Calore 3-D Cross Section Temperature Order

Error analysis was performed for the finite element cube solid conduction solution. Temperature error in determined throughout the domain is illustrated in Figure 5.9.5. As in previous analyses, error values are based on the Richardson extrapolated estimate of the exact solution. Error over the entire solution is less than 0.20 percent. Further error illustration is shown in Figure 5.9.6 with the same

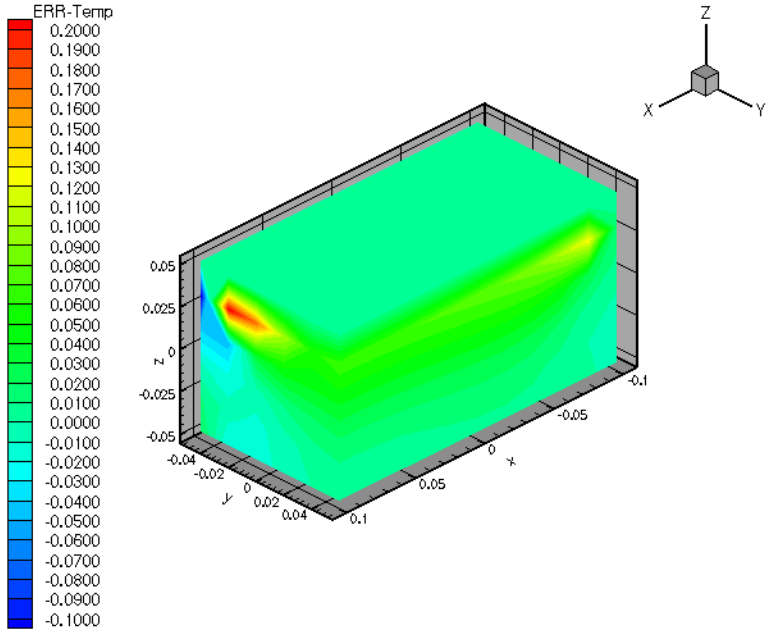

Figure 5.9.5 - Calore 3-D Unstructured Temperature Error

slice planes used for order. For the boundary conditions employed in this problem,

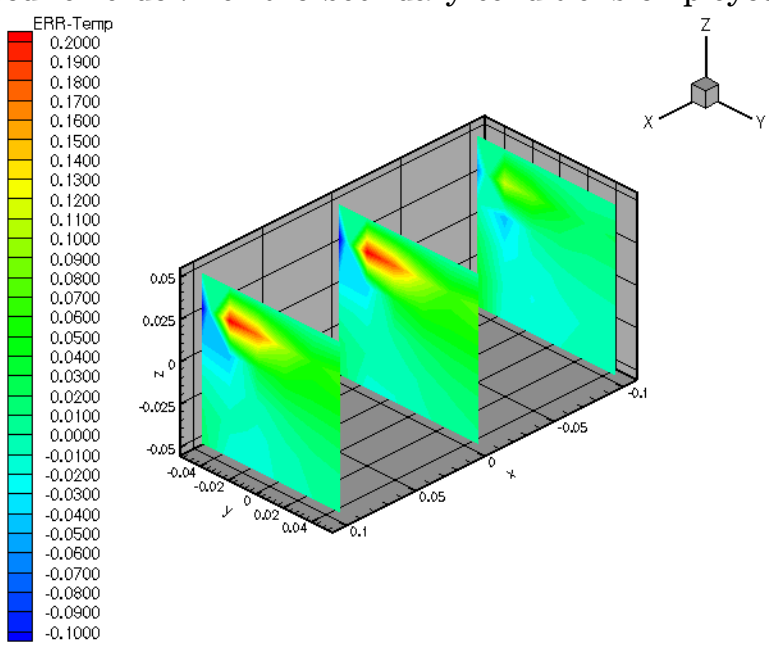

Figure 5.9.6 - Calore 3-D Cross Section Temperature Error

the temperature solution is accurate to within 0.10 percent (even in areas of lower order behavior).

6. Conclusions and Future Work

A software program (VIVID) has been developed to aid in the formalization of the verification process. VIVID enables order of accuracy and error analyses of a variety of numerical method programs. As a numerical solution analysis program, it greatly simplifies the task of performing meaningful detailed grid convergence studies. This is an extremely important milestone in this time of shrinking experimentation and increased demands on the computational community. The Richardson extrapolation procedure forms the heart of the VIVID analysis process. This procedure was developed for finite difference and control volume numerical methods. It is strictly applicable only to uniform mesh problems. It has been applied to a wide variety of simulation programs and mesh types. The results coming from nonuniform applications need to be viewed in an inquisitive light. There is much yet to comprehend from the results. VIVID's employment of the procedure has allowed evaluation of dependent variable solution order behavior for a number of quantities at the local level over solution domains never before attempted.

VIVID evaluates numerical solutions created by computational programs regardless of whether the grid employed is structured or unstructured. As indicated in Sections 2 and 4, both EXODUS and TECPLOT data base formats can be used for the multiple input solutions required in a grid convergence study. Finite volume and finite element solutions were processed with this tool. Due to the similarity of the numerical formulation of finite volume and finite difference techniques, Richardson extrapolation can be applied directly to solutions based on this method. On the contrary, there is an open question as to the applicability of this methodology to the finite element numerical formulation. Regardless, in all cases examined (including finite element solutions), significant regions of second order behavior were observed for programs nominally based on a formal second order numerical scheme. It should also be noted that there were some regions of lower order behavior in these same solutions. It is the consensus of the community that these regions could be an indication of insufficient grid refinement when they occurred away from an imposed boundary condition.

Single dimension and multiple dimension refinement schemes were evaluated in these analyses. Either approach may be used successfully in a grid convergence study. Two-dimensional and three-dimensional numerical examples were presented. Primarily second order behavior characteristics were observed in the steady state three-dimensional solutions. These results were very similar to those of the two-dimensional analyses. Order analyses, based on error relative to extrapolated estimates of the exact solution, yielded similar results to those based on error relative to an exact analytical benchmark when such a comparison was possible. As such, it was concluded that Richardson extrapolated estimates are sufficient for order analyses when benchmarks are not available. Most analyses discussed were of the steady state variety. However, an order of accuracy analysis was performed on a transient two dimensional problem. In this case, VIVID's unique ability to treat time as a third

dimension was invoked. The results illustrated an essentially first order time accurate behavior for the FCV/Fuego finite volume program.

Although significant analysis capability has been provided by the current version of VIVID, considerable work remains to bring it to the level of being broadly useful in the computational community. Currently, it has recognized weakness in three areas. First, it needs broadening in its input and output data formats. Its interface capabilities need to be expanded to handle a wider variety of input and output options commensurate with those in use in the technical community. This should most certainly include the common data base format yet to be standardized among the ASCI establishment. Second, the benchmarking subroutine options are in their infancy. A number of accepted standards within the computational community need to be incorporated into the program; not the least of which is the De Vahl Davis problem. Finally, a more efficient method of handling large computational problems must be developed. For VIVID to fulfill its role as a verification tool for ASCI, it must be able to evaluate tera-flop sized calculations. In its current form, this is not possible. Manipulation of multiple variables in multi-block, three-dimensional, transient solutions for multiple input files quickly leads to a seven dimensioned analysis space. Conventional storage of such arrays rapidly leads to exhaustion of recognized storage addresses for a short word machine (such as the work station environment). A number of approaches may help this situation; beginning with dynamic memory management and parallellization. Addressing these areas will require significant effort in the future with the end result being a dynamic, versatile, robust verification capability.

References

[1] Blottner, F. G., "*Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nosetip*," Journal of Spacecraft and Rockets, Vol. 27, No. 2, pp. 113-122, March-April 1990.

[2] Blottner, F. G., and Lopez, A. R., "*Determination of Solution Accuracy of Numerical Schemes as Part of Code and Calculation Verification,*" SAND98-2222, Sandia National Laboratories, Albuquerque, NM, October 1998.

[3] Richardson, L. F., and Gaunt, J. A., "*The Deferred Approach to the Limit*," Transactions of the Royal Society of London, Seriea A Vol. 226, pp. 299-361, 1927.

[4] Oberkampf, W. L., Blottner, F. G., and Aeschliman, D. P., "*Methodology for Computational Fluid Dynamics Code Verification/Validation*," AIAA Paper No. 95-2226, June 1995.

[5] Oberkampf, W. L., "*A Proposed Framework for Computational Fluid Dynamics Code Calibration/Validation*," AIAA Paper No. 94-2540, June 1994.

[6] Walker, M. A., and Oberkampf, W. L, "*Joint Computational/Experimental Aerodynamics Research on a Hypersonic Vehicle: Part 2, Computational Results*," AIAA 29th Aerospace Sciences Meeting, January 7-10, 1991, Reno, NV.

[7] "*TECPLOT, Productive Power for Data Visualization; User's Manual Version 7*," Bellevue, WA: Amtec Engineering, Inc, August 1996.

[8] Roache, P. J., "*Verification of Codes and Calculations*," AIAA Paper No. 95-2224, June 1995.

[9] "*Handbook of Single-Phase Convective Heat Transfer*," Edited by S. Kakac, R.K. Shah and W. Aung, John Wiley & Sons, New York, 1987

[10] Blackwell, B. F., "*X21B11Y23B01G1 - Infinitely Long Rod of Rectangular Cross Section*," Unpublished Sandia Working Notes on Steady State Analytic Conduction Problem, Dated August 31, 1998.

[11] Dowding, K. J., "*X21B11Y23B01G1T1 - Infinitely Long Rod of Rectangular Cross Section*," Unpublished Sandia Working Notes on Transient Analytic Conduction Problem, Dated August 31, 1998.

[12] "*CFD-ACE Graphical User Interface Manual Version 2.0*," Huntsville, Alabama: CFD Research Corporation, July 1996.

[13] Shadid, J. N., Moffat, H. K., Hutchinson, S. A., et al., "*MP-Salsa A Finite element Computer Program for Solving Reacting Flow Problems, Part 1 - Theoretical Development*," SAND95-2752, May 1996.

[14]Bova, S.W. and Copps, K.C., "*Calore: A Computations Heat Transfer Program. Volume I. Theory Manual*", SNL, Albuquerque, NM, 2004, In Prpearation, SAND Report XXX-XXXX.

[15]Smith, Ober, Lorber, "*SIERRA/Premo - A New General Purpose Compressible Flow Simulation Code*", AIAA-2002-3292, St. Louis, 2002.

[16]"*INCA User's Manual, Version 2.0*," Amtec Engineering, Inc., Bellevue, WA,1995.

[17]Wong, C. C., Blottner, F. G., Payne, J. L., and Soetrisno, M., "*Implementation of a Parallel Algorithm for Thermo-Chemical Non-equilibrium Flow Solutions*," AIAA Paper No. 95-0152, January 1995.

[18]Cochran, R. J., Martinez, M. J., Evans, G.H., Moen, C. D., Blottner, F. G., Burns, S. P., Tieszen, S. R., Nicolette, V. F., and Gritzo, L. A., "SIERRA/Fuego Theory Manual," SAND99-??, Sandia National Laboratories, Albuquerque, NM, Unpublished.

[19]G. de Vahl Davis, "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution," *International Journal for Numerical Methods in Fluids*, Vol. 3, pp. 249-264, 1983.

[20]G. de Vahl Davis and I. P. Jones, "Natural Convection of Air in a Square Cavity: A Comparison Exercise," *International Journal for Numerical Methods in Fluids*, Vol. 3, pp. 227-248, 1983.

[21]M. Hortmann, M. Peric, and G. Scheuerer, "Finite Volume Multi-grid Prediction of Laminar Natural Convection: Bench-Mark Solutions," *International Journal for Numerical Methods in Fluids*, Vol. 11, pp. 189-207, 1990.

[22]Mary A. McWherter-Payne, Amalia R. Lopez and R. J. Cochran, "ASCI SIERRA/Fuego Verification and Validation Effort: 1) MP-Salsa solutions compared to numerical benchmark; and 2) Numerical accuracy/error methods applied to MP-Salsa results", Sandia Memo, March 19, 1998.

[23]Roy C.J., Smith T.M., and Ober C.C., Verification of a compressible CFD code using the method of manufactured solutions. AIAA Paper 2002-3110, 2002.

Appendix A. Program Structure Details

This appendix provides details for the program structure. It is meant to act as a supplement to Section 2 contained in the main body of the report. The paragraphs contained in Section 2 introducing each subprogram are repeated here for the convenience of the reader.

A.1 Input Data Extraction

Input data parsing for VIVID is handled by an original program written specifically for the ASCI/Fuego verification process. Numerical solution data to be evaluated by the VIVID processor is initially input to the **EXTRACT** subprogram. The program name comes directly from its functional purpose. It literally sorts through the input files and ***extracts*** the parts of each solution defined by the user to be evaluated in future analyses. This program can be ran in an independent mode or as part of an order of accuracy analysis.

Depending on user discretion and the mode of operation, one, two, or three input files can be handled in a single run. Order of accuracy analyses can be performed for either two or three numerical solution input files. Currently, numerical solutions may be input to the VIVID program in the form of EXODUS unformatted binary files or TECPLOT Preplot block formatted files. As used in this procedure, EXODUS file solutions are strictly for unstructured grid problems. However, TECPLOT input files may be for structured or unstructured grid numerical solutions. EXTRACT can accept input of a two-dimensional or three-dimensional nature. Input TECPLOT files can have multiple zones, while EXODUS input files may have multiple blocks. As well, multiple dependent variables are allowed for both input formats. Multiple time planes are also accommodated for the EXODUS format. In association with this option, it is possible to choose output for a dependent variable as a function of time at specified coordinate locations from within EXTRACT. At the end of the data sorting process, column formatted output files are written for further analysis. For an independent run of EXTRACT, this may take the form of simple plotting or input to another user defined program. For a run that is part of an order of accuracy evaluation, the output files are directly usable as input to the ORDER program. As an additional visualization feature, EXODUS style input files are translated and output as TECPLOT Preplot formatted files.

A.2 Reference Data Production

As with the input data extraction process, reference baseline data generated by VIVID is produced by an original program written specifically for the ASCI/Fuego verification process. Reference solutions used for comparative analysis purposes by

the VIVID processor are created by the **BENCHMARK** subprogram. The routine name comes from the procedure's functional purpose. It produces ***benchmark*** solution information to be used in future analyses by options defined by the user. Just as with the EXTRACT procedure, this program can be ran in an independent mode or as part of an order of accuracy analysis.

In general, reference data for numerical solution comparative analyses can come from several sources. Currently, efforts within the verification group are focussed on three options. Those options are analytical solutions for relatively simple problems, professional community accepted numerical benchmark solutions, and a relatively new class of manufactured solutions making use of source term inputs determined from solution of the governing partial differential equations for an assumed answer. At this time, only references of the first variety (analytical solutions) are available within BENCHMARK. Baselines from all three categories will be installed in the future. Two analytical solutions are available for current analyses. They are the *Flat Duct Incompressible Flow* problem and *Two-Dimensional Heat Conduction* analysis for a rectangular geometry. Specific boundary condition restrictions are placed on each problem and are discussed in detail in the code functionality section of this document (Section 4). Two problem sub-types are available within each analytical solution class. Flat duct problems may have a constant velocity or fully developed velocity profile inlet flow boundary condition. Heat conduction evaluations may be of a steady state or transient variety.

At the conclusion of the analytical run, a column formatted output file is written for further analysis. For an independent run of BENCHMARK, this may take the form of simple plotting or input to another user defined program. For a run that is part of an order of accuracy evaluation, error analysis using the analytical solution as the baseline value is output in the ORDER subprogram's TECPLOT Preplot formatted file.

A.3 Order of Accuracy and Error Analyses

The analysis functions, including order of accuracy and error estimation, conducted on the numerical solutions input to VIVID are performed by a third original program written specifically for the ASCI/Fuego verification process. Numerical solution data evaluated by the VIVID processor is analyzed by the **ORDER** subprogram. Like the two previous subprograms discussed in this section, the current procedure name also comes directly from its functional purpose. The routine evaluates the ***order*** of accuracy for the numerical method over the region of the solution defined by the user. As stated in the first section, order of accuracy relates the amount of error reduction in a numerical solution for a specified amount of grid refinement. Stated mathematically this means that the ratio of the error between two solutions is equal to the grid refinement factor raised to a power corresponding to the

order of accuracy of the numerical method (see Equ. 2.1). In Equation 2.1, E, r, and p

$$E_2/E_1 = r^p \qquad \textit{(r=mesh 2 nodes/mesh 1 nodes > 1)} \qquad \textbf{(2.1)}$$

are the solution discretization error at the grid points, grid refinement ratio, and order of accuracy, respectively. Subscript one represents the refined grid solution. The refinement ration r is the ratio of the physical size of the nodes in solution two versus solution one. Order of accuracy analyses performed by this subroutine can be done in concert with the VIVID resident EXTRACT and BENCHMARK programs or independent of them if the appropriate columnar files are directly generated by a numerical method. Analyses can be performed with input of either two or three numerical solution files.

It is implicit from the discussion of the EXTRACT program in Section 2.1 that ORDER can evaluate two or three-dimensional geometries for either structured or unstructured grid problems. It is further implied that multiple dependent variables may be analyzed for multiple block or multiple zone input geometries. The multiple time plane option noted in the EXTRACT discussion is used by ORDER to provide a unique feature for two-dimensional geometry order of accuracy analyses. At the user's discretion, time can be interpreted as the third dimensional variable allowing transient order accuracy evaluations for multiple solutions varying time step size.

Estimation of solution error within ORDER can be performed in two different fashions. Error can be estimated from the BENCHMARK solutions as alluded to in Section 2.2. In addition, the option exists to determine error from an estimated exact solution obtained from application of the Richardson extrapolation procedure. The mathematics of this procedure are discussed in detail in Section 3. When using the Richardson approach, the option exists within ORDER to estimate the exact solution from either the formal order of accuracy of the numerical method or from the demonstrated order calculated from the analysis of the input numerical solutions. At the end of the analysis process, a column formatted output file is written for further analysis. This may take the form of simple plotting or input to another user defined program. Additionally, an output file is generated in TECPLOT Preplot block format containing the calculated order of accuracy and error estimate variables for each dependent variable examined.

Appendix B. Code Development and Functionality

This appendix provides details about the code development and functionality. It is meant to act as a supplement to Section 4 contained in the main body of the report. Some of the information contained in Section 4 introducing each subprogram is repeated here for the convenience of the reader. Most paragraphs have been expanded in an effort to ease mastering the use of VIVID to perform grid convergence studies on a series of numerical solutions. With that in mind, Section B.5 contains the interactive screen output during a typical run for an unstructured evaluation.

B.1Main Program: VIVID

VIVID is an interactive procedure that is invoked on the Sandia - Albuquerque 9100 Engineering Sciences IRN LAN by typing *vivid* from a command line window. It may be run in two modes: interactive or control file driven. Following is a discussion of the primary routines structure and functional attributes. Included in the discussion are its use of command line arguments and interrogative features.

B.1.1 Driver Routine Structure

The main program's functionality can be categorized into three parts: command line parser, interactive parser, and program option control. For VIVID, the command line parser determines the disposition of the control file. It checks for its existence on the command line invoking the procedure. If one exists, control file mode is initiated. The file is opened for subsequent reads performed by the main and subprograms executed in the analysis. From this point, the run is controlled by the directions in the namelist format file. No further user interaction is allowed. If there is not a control file, a new control file is opened and the interactive mode is initiated. All interactive instructions made by the user in program queries are written to the new file. This is a namelist format file and is readily interpreted by and available to the user for editing for later runs. Use of the input control file option eliminates the need to pass through the lengthy interactive process associated with a VIVID run.

Following determination of the control file disposition, the screen interactive parser of the driver program is encountered. A single operational parameter defining subprogram control is specified in this section of the routine. If no control file is present, the user is asked to interactively define the subprogram execution option (i.e. ORDER, EXTRACT, or BENCHMARK). Otherwise, the subprogram option is read from the first segment of the namelist control file. Next, the program option control portion of the main routine is executed. This section initiates the desired subprogram via a Fortran subroutine call. Following completion of the selected subprogram option, termination control is reached. For an interactively controlled execution (i.e. no input control file), the user is given the option of initiating another VIVID run. Otherwise, the program is terminated.

B.1.2 Command Line Arguments

VIVID has the capability to interpret command line arguments. This is limited to one argument on the command line invocation. The argument following the *vivid* syntax is the name of the control option input file. If no argument is found, all required input file names and control parameters are defined in the pursuant interactive interrogative screen sessions conducted by the main program and the subprogram codes it executes. A VIVID run may use anywhere from one to three of its constitutive programs during execution. The precise nature of the run is determined by the control option file or any required interactive screen session.

Following activation, the code checks for a command line argument. If one is present, the code assumes this is the name of the control option input file mentioned previously. A sample of the syntax for code execution by control file is shown in the following line.

vivid filename

The file name, *filename* as denoted above, is completely arbitrary. The user may append any name for execution in this mode. A control file by the name *vivid_control.i* is generated automatically by the main program every time VIVID is executed interactively and run to conclusion. This file can be renamed to suit the user's purpose and employed in subsequent runs. Some typical examples are illustrated below.

vivid vivid_control.i

vivid vivid_structured.i

vivid unstructured_control

The only limitation on the input file name is that it must be fifty characters or less in length. Samples of typical input files are contained in Appendix C.

B.1.3 Interactive Interrogation & Constitutive Program Execution

Assuming an input control file run is not invoked, VIVID uses a screen interactive parser to establish the majority of the execution control parameters. Most of these parameters are defined in the subprograms. The specifics of the subprogram interactive interrogations are discussed beginning in Section B.2. However, the driver program does establish two fundamental variables. The first is declaration of the subprogram option. This determines whether to invoke ORDER, EXTRACT, or BENCHMARK. In addition, the main program controls the run repeat option. Whether interactive or control file mode has been chosen, execution parameters containing solution variables and extent are defined within the solution input files.

Constitutive program execution is begun after the subprogram option is defined interactively or from reading the input control file in the main routine. All subprograms are incorporated as subroutines of VIVID. A directed GO TO statement is used in the program to sort the run alternatives. Specifics of the EXTRACT, ORDER, and BENCHMARK subroutines are discussed in Sections B.2, B.3, and B.4, respec-

tively. EXTRACT and ORDER share run control information efficiently through common blocks during order of accuracy runs. Common block dimensions associated with solution size are controlled throughout the main and subprograms by an include file (sample file contained in Appendix D). This allows VIVID to be simply tailored to match limiting problem dimensions or for the computing platform being employed. A simple change of a controlling dimension, prior to compilation on a platform, re-dimensions the entire code. ORDER's communication with BENCHMARK is done explicitly through the subroutine call. This approach was taken due to the single point nature of the evaluations done by the subroutines called under these circumstances.

B.2 EXTRACT Subprogram

EXTRACT was written to isolate and output specific information of interest for further analysis. It can be ran independently or as part of an order of accuracy evaluation. In the independent mode, files are sorted serially with repetition terminated at the user's discretion. In the case of order accuracy analyses, two or three numerical solution files are accepted serially as input, processed per the user's instructions, and output as column formatted files for use by the ORDER subroutine.

The program begins with an interactive interrogation section that establishes an initial set of control variables for the file to be extracted. Domain, grid, input file type, and the all dependent variable option flags are defined first. This group of flags is set on the first run of the program and remains fixed regardless of how many files are processed by EXTRACT. The domain flag defines the overall nature of the extraction run. If the domain option is invoked, every nodal point within every computational zone extracted will be evaluated for at least one dependent variable. Use of either structured or unstructured grid evaluation subroutines are set by the grid flag. In the case of an order of accuracy run of EXTRACT, the domain and grid flags are set in ORDER prior to execution of the extraction subprogram. The input file type flag sets the input data format. This is limited to TECPLOT PREPLOT files or EXODUS data base files. Although, an option has been included to allow an experienced programmer to easily add a subroutine to read a file format of their own designation. Definition of the all dependent variables flag (i.e. *ALLDEP*) determines whether the user will be required to define specific dependent variables for data extraction in subsequent interactive interrogations. Setting it to the affirmative ensures that each dependent variable in a solution file will be extracted. Affirmatives for *ALLDEP* and *DOMAIN* flags for all zones amounts to extraction of an entire file.

Following definition of the initial group of flags, the serial repetitive section of the routine's interactive parser is encountered. This section of the subprogram is repeated for each file evaluated. It begins with definition of the input and output file names. A convenience feature has been included in the interactive input file selection procedure. From the second file on, a list of files with the same root name as the first file processed is presented for the user to choose from. The program's default in-

put selection corresponds to the next logical file choice (i.e. increasing or decreasing file suffix number).

File name selections are immediately followed by specification of a group of variables that define the extraction extent of a given file. This includes extraction zones, extraction variables, extraction time planes, and extraction nodes. First, the extraction zone numbers (blocks for EXODUS files) are defined. The option has been provided for selective evaluation of any or all computational zones within a solution. Zone number selection is allowed for the first file only. The assumption is made that all subsequent files processed in the same execution of EXTRACT should use the same zone definitions. Similar features are provided for file variables to be extracted. Assuming that the all dependent variables flag was answered with a negative in the previous section, the code requires definition of variables to be extracted from the input file. As with zone numbers, this selection is allowed on the first pass only and may include any or all variables within the file. At this point, a divergence of options exists depending on whether the input files are structured or unstructured. Structured input allows the user to select the extraction range on the x, y, and z dimensions by specifying nodal range in each direction. This option allows for isolation of very specific regions within numerical solutions. The selection is made independently for each file processed. If a full domain extraction is desired, the entire nodal extent should be specified in all directions. Currently, a comparable option is not available for unstructured input files. However, unstructured files do present the option of multiple time planes. This allows for examination of transient order of accuracy. Time planes of interest are specified for examination as well as the option of examining specific locations within the solution domain as a function of time.

The serial repetitive interactive parser section concludes after defining the extraction extent variables. Files are immediately read by calls to subroutines specifically written to be compatible with an input type. Control of this feature is accomplished with a directed GO TO structure. Extraction of the data of interest and output to formatted files follows the file read by employing the extraction extent parameters as limits in multiple nested DO loops. Multiple output file types may be produced by this subprogram. A maximum of three file types may be produced for each file processed. Multiple time plane unstructured EXODUS input files may result in the output of two column formatted files as well as a TECPLOT PREPLOT file of the original input data. The two column formatted files are the time dependent information at specific grid locations and the extracted information ready for use in an order of accuracy analysis.

B.3 ORDER Subprogram

ORDER was written to estimate the order of accuracy and error of numerical solution methods. To calculate order independently, three numerical solutions are required for the procedure. Two solution error estimate options are available if the order is assumed equal to the algorithm formal order of accuracy or if a benchmark reference is available.

The program begins with an interactive interrogation section that establishes an initial set of control variables for the files being evaluated. Domain, grid, number of input files, two-dimensionality, and interpolation option flags are defined first. The domain and grid flags are the same as those discussed in the EXTRACT subprogram. The domain flag defines the overall nature of the order run. If the domain option is invoked, every nodal point within every computational zone examined will be evaluated for at least one dependent variable. Use of either structured or unstructured grid evaluation subroutines are set by the grid flag. The flag setting the number of input files is used by both the EXTRACT and ORDER programs for I/O purposes. Special options relating to transient functions are controlled by the dimensionality flag. Definition of the interpolation flag determines how to treat non-coincident grid points. This option is in its early stages of development and is not widely available to the user community.

Following definition of the initial group of flags, control is passed from ORDER to the EXTRACT subprogram. As discussed earlier, the input information to be evaluated in ORDER is sorted out by EXTRACT and passed back to it in the form of columnar input files. An option exists to bypass the extraction run and operate on a series of preexisting ORDER input files. These files could come from previous runs of EXTRACT that can now be operated on directly with a variety of analysis options. Immediately upon regaining control from the extraction subroutine, input and output file names are requested. Current program default input file names are set to the output file names produced by EXTRACT. **Input file names must be entered in order from coarse grid to fine grid.** Each solution must be entered in an independent file. All input solutions can **NOT** be contained in one file. If this is a domain run, a TECPLOT PREPLOT format output file is also produced for multi-dimensional visualization purposes. Immediately after defining the input files, independent and dependent variable columns are set. If the *ALLDEP* flag (discussed in the EXTRACT subprogram) is set to the affirmative, all dependent variable columns in a file will be evaluated for order of accuracy. Otherwise, the user will be queried at this point to define a single independent and a single dependent variable column for evaluation.

Input and output data definitions are immediately followed by specification (through interactive queries or control file reads) of a group of variables that define the analysis options governing the order evaluation. This includes grid refinement factor, numerical solution formal order of accuracy, error calculation options, and benchmark solution flags. First, the grid refinement factor is defined. This is a con-

stant defining the relative spacing relationship between nodal points for successive solutions. Although it must be constant, it does not have to be an integer. Numerical method formal order of accuracy is defined next. This variable relates to the discretization formulation used for the difference equations in the numerical program that produced the solutions. For example, keeping the second order terms in the Taylor series expansions in a numerical method and truncating higher order terms would provide a formal accuracy of two. Subsequent to the formal order definition, control flags defining error calculation options are set. Errors calculated from extrapolated estimates of the exact solution (such as in Equ. 3.10) are a function of the order of accuracy. Thus, the solution error may be estimated from the calculated order of accuracy from the multiple solutions or from the formal order of accuracy specified for the numerical method. A user option is provided for this choice. In addition, the user is given the option of choosing the input solution for which error is to be evaluated. Unless otherwise specified, error is determined for the finest grid solution. Finally, options relating to BENCHMARK solution use are defined. The user is asked to specify the availability of a BENCHMARK program contained within VIVID which matches the current problem type being evaluated. Further, a flag is set defining the use of any such existing BENCHMARK calculation for error determination purposes.

The interactive parser section concludes after defining the analysis control variables. At this point, input files are immediately read by multiple nested explicit and implicit do loops using file extent variables as indices. The read process stores the data in a high dimensional array (6-D for structured & 4-D for unstructured input) for ease of grid point matching between solutions. Array dimensions in the structured storage variable represent three spatial directions, variable number, zone number, and file number. Since unstructured methods do not necessarily adhere strictly to x, y, and z orientations, the storage variable array replaces the three spatial dimensions with a single node number dimension. However, the storage variable size requirements are not reduced. Since the same physical domain must be defined to similar accuracy, a similar number of nodes are required regardless of the grid type. The node number dimension is equal to the product of the x, y, and z dimensions in the structured variable array. Following read completion, the multi-dimensional variable storage array is processed by a grid point coordinate matching subroutine (i.e. GRID_MATCH_S or GRID_MATCH_U). This subroutine compares the values of the spatial variables of the refined grid solution(s) to those of the coarse grid solution to determine matching nodes. This results in a revised storage array with comparable physical points stored at matching array indices across all files. After the grids have been matched, error and order of accuracy analyses are completed. These analyses are based on the detailed equations presented in Section 3.

The analysis results in the production of order of accuracy and error variables for all dependent variables analyzed. Results are subsequently written to a TECPLOT PREPLOT block format file (*order_tec.o*) for multi-dimensional visualization. The original dependent variables analyzed are also written to this file for convenience. Independent and dependent variable names are automatically extracted

from the original input files and output to the TECPLOT file. In addition, the names for the order and error variables are generated automatically from the dependent variable names. As well, a column formatted output file with details of the extrapolation analysis is written to file *order.o*. This file contains specific extracted, extrapolated, and benchmark values at all domain points analyzed for a dependent variable. The order and error variables are also included in this file.

B.4 BENCHMARK Subprogram

BENCHMARK was written to produce reference data for comparison to numerical program solutions. It can be ran independently or as part of an order of accuracy analysis. In the independent mode, it is accessed repetitively directly from the main routine with repetition terminated at the user's discretion. In the case of order accuracy evaluations, it is accessed by the ORDER subprogram and executed at each point for which an accuracy analysis is being performed. Column formatted output files are generated for the reference solutions.

The BENCHMARK subprogram control routine is similar to the main routine of VIVID. It is a driver program with no indigenous analysis capabilities. The subprogram acts as an analysis traffic director based on the user's directions. Currently, it contains two analytical subroutines for verification purposes. However, it is written in a highly modular fashion for ease of expansion to include future verification subroutines. The current reference routines available for comparison are the two-dimensional flat duct incompressible flow program and the two-dimensional heat conduction program. Multiple sub-options are available for both programs. In addition, development work is proceeding on four other reference routines: Couette flow, inviscid cross flow cylinder, three dimensional heat conduction, and manufactured solutions.

Assuming that this is not an input file controlled run, BENCHMARK begins with an interactive interrogation section like other routines within VIVID. In the reference program driver routine, the analytical program option (i.e. flat duct or conduction) is defined first. Depending on run mode (independent or order), the solution extent option is defined; single point or complete geometry domain analysis. Subsequent to the run extent definition, relevant geometry parameters are set. At this point, control is passed to the FLAT_DUCT or CONDUCTION_2D subroutine through a directed GO TO arrangement. These subroutines are discussed in detail in the following two sections.

B.4.1 Flat Duct Incompressible Flow

Functionally, the flat duct subroutine begins with an interactive parser to set up boundary conditions, geometry, and fluid variables. Initially, in the case of an ORDER run, it shows the user what dependent variable is being examined and presents the option to evaluate this quantity analytically. The user may choose to bypass the routine at this point. Should the procedure be continued, an analytical solution based on the following development is executed.

B.4.2 Two-Dimensional Heat Conduction

As in the flat duct subroutine, the heat conduction program begins with an interactive parser. In the heat conduction interactive parser, boundary conditions and material property variables are defined. In a variance from the flat duct approach, the user is allowed the option of defining these terms in an input file. An example of this file is shown in Appendix D. Initially, in the case of an ORDER run, the program displays the dependent variable being evaluated. The user is provided the option to evaluate this quantity analytically. As above, the user may choose to bypass the routine at this point. If processing is continued, an analytical solution of the two-dimensional heat conduction equation is executed.

B.5 Interactive Screen Session for VIVID

Following is a copy of the reports back and forth between the user and the program for a typical VIVID ORDER analysis. This run is for an unstructured grid geometry. The length of the interactive session illustrated here was the driving function for development of the control file run capability. The control file illustrated in Appendix Section C.2 was produced by this run.

```
irn $ vivid

 ****************************************************
*          WELCOME TO PROGRAM              *
****************************************************
****************************************************
*                                  *
*   *    * *** *    * *** ******    *
*   *   *  *  *   *  *  *   *   *
*    * *   *   * *   * *   *   *
*     * *   *   * *   * *   *   *
*      *   ***    *   *** ******    *
*                                  *
****************************************************
****************************************************
****************     VIVID      ****************
**************** Analysis Tools ****************
****************      for       ****************
**************** Verification  ****************
****************************************************
**************** SUN RELEASE   ****************
**************** Version 4.4   ****************
**************** June 2000     ****************
****************************************************
***  FOR INFORMATION ON THIS PROGRAM CONTACT  ***
***     D.L. POTTER AT (505) 844-5573       ***
***     SANDIA NATIONAL LABORATORIES        ***
```

```
   ***        Aerosciences Department          ***
   ************************************************


*******************************************************************************

   WELCOME !! You are Running VIVID/Version 4.4 !  (Revised 6-27-00)
   (Last Minor Moifications Made 9-1-00.)

   Direct Inquiries to: D. L. POTTER
                  SANDIA National Laboratories,
                  Department 9115, {(505) 844-5573}

*******************************************************************************

   *** VIVID Execution began on: Fri Sep  1 15:43:52 2000 ***

   Specify Desired VIVID Subprogram Option:
     1-ORDER (Calculates Error & Numerical Program Effective Computational Order)
     2-EXTRACT (Produces Column Formatted files from TECPLOT/PREPLOT & EXODUS Files)
     3-BENCHMARK (Runs a Benchmark Tool Independently)
     4-Stop Execution !!!
   ENTER HERE (CR->1):1

   How Many GRID POINTS are in Your Fine Mesh File ?
   ENTER HERE [CR->100000]:6601

   How Many TIME PLANES are to be Stored for Analysis ?
    * NOTE * Modifies Dynamic Memory Allocation so Time can be used as Spatial Coordinate.
   ENTER HERE [CR->1]:1

   How Many SPATIAL Dimensions are in Your Problem ?
   ENTER HERE [CR->2]:2

   Currently Making Dynamic Memory Allocation for 14639147 Double Precision Spaces !!!


*******************************************************************************

   WELCOME !! You are Running ORDER/Version 2.1 !  (Revised 3-6-98)
   (Last Minor Moifications Made 3-13-98.)

   Direct Inquiries to: D. L. POTTER
                  SANDIA National Laboratories,
                  Department 9115, {(505) 844-5573}

*******************************************************************************

   *** ORDER Execution began on: Fri Sep  1 15:44:22 2000 ***
```

```
Do You Wish to Evaluate Error at EVERY Node in EACH Zone of Interest ?
 * NOTE 1: This is the DOMAIN Option. At least 1 Dependent Variable is evaluated
        at EVERY Node in EVERY Zone Analyzed !!!
 * NOTE 2: The DOMAIN Option MUST be Activated to Produce TECPLOT Output Files !!!
ENTER HERE [YES/CR OR NO]:y

What Solution Difference Definition do You Want Set as Equivalent to ZERO ?
 * NOTE 1: This is Directly Effects the Order of Accuracy Results When
        Consecutive Solution Differences Become Excedingly Small !!!
ENTER HERE [CR->1.0 E-14]:1.e-7

Is this a STRUCTURED/S or UNSTRUCTURED/U Grid ?
ENTER HERE [S OR U/CR]:u

Is this a 2 or 3 File Evaluation ?
 * NOTE * For a 2 File Evaluation, the Computed Order of Accuracy from the
        Estimated Exact Solution will be IDENTICAL to the Declared FORMAL ORDER !!!
ENTER HERE [CR->3]:3

Specify the Grid Point ZONE/BLOCK Mapping Option:
  1-Compare Points Within LIKE Numbered Zones/Blocks Only.
  2-Compare Points Across ALL Zones/Blocks.
ENTER HERE (CR->1):1

Will You Allow Linear Interpolation Between Grid Points in the ORDER Calculation ?
 * NOTE * This Option is Temporarily Disabled !!!
ENTER HERE [YES OR NO/CR]:n

Do You Need EXTRACT Processing of Your Input Files ?
 * NOTE 1: This is REQUIRED to Create Input Files Compatable with ORDER !!!
 * NOTE 2: Not Necessary if This is a Rerun working with Previously Extracted Files !!!
ENTER HERE [YES/CR OR NO]:y


*******************************************************************************

WELCOME !! You are Running EXTRACT/Version 2.1 !  (Revised 3-13-98)
(Last Minor Moifications Made 3-13-98.)

Direct Inquiries to: D. L. POTTER
                SANDIA National Laboratories,
                Department 9115, {(505) 844-5573}

*******************************************************************************

*** EXTRACT Execution began on: Fri Sep  1 15:45:01 2000 ***
```

```
Specify the Input File Data Base Type:
  1-TECPLOT (PREPLOT BLOCK, POINT, or FEBLOCK Compatable)
  2-EXODUS II (Binary Data)
  3-PLOT3D
  4-ASCI Common Data Base Format
  5-User Supplied Read Subroutine
  6-Stop Execution !!!
ENTER HERE (CR->2):2

Do You Wish to EXTRACT ALL Dependent Variables from the Input Files ?
ENTER HERE [YES/CR OR NO]:y

 * NOTE * Enter DATA BASE Files from Coarse to Fine Grid in Order.

Specify DATA BASE Input File # 1 to be Converted to Column Formatted Data.
 * NOTE 1: The TECPLOT File is a PREPLOT Input file !!!
 * NOTE 2: The EXODUS II File is an Unformatted Binary file !!!
ENTER HERE (CR->extract_i.1):maw_ran_exo.1

ICHS = 22

Specify Output File Name to be Created by Extracting Information from the Input File.
 * NOTE * For Computational Accuracy Evaluations, Output File Names Created here
        MUST be the Same as Input File Names Requested Later by ORDER Subprogram !!!
ENTER HERE (CR->maw_ran_exo_1_extract_1.o):

Specify the Computational Region Numbers to be Converted to a Column Formatted File.
 * NOTE 1: Currently Limited to 10 Data Bocks !!!
 * NOTE 2: This is ELEMENT Blocks for Unstructured !!!
 * NOTE 3: You will be asked to Specify the Total Number of ZONES/BLOCKS in the File !!!
 * NOTE 4: TECPLOT Input File ZONES MAY be Spatial or Temporal in Nature !!!
ENTER HERE (CR->All Zones):

How many ZONES/BLOCKS are there in the TEC/EXODUS Input File ?
 * NOTE 1: For EXODUS Files, This is referring to Spatial Computational Regions !!!
 * NOTE 2: For TECPLOT Files, The ZONES may be Spatial or Temporal !!!
ENTER HERE (CR->1):

Number of Nodes = 451 in File 1.

Element Block/Zone Numbers:     1

Current Extraction # 1 Reading Element Block 1.

Number of INPUT File Times = 1.

Time Plane Values:     1: 0.0000E+00

NTPLANE(NTPLANES) =      1
```

```
Maximum EXODUS Time Plane of Interest =   0.000000E+00 seconds.

Following are the Header Lines from Input File 1.

Refined Mesh 0001

Variables: X        Y        zcoor     VX        VY        Pres

Is the THIRD Variable in the List Above a Coordinate (i.e. Z) ?
 * NOTE * If the Third Coordinate Variable is NOT Shown, It WILL be Supplied !!!
ENTER HERE [YES/CR OR NO]:

INPUT File Time Plane 1 Read from EXTRACTED Block 1 of File 1.

QUAD Elements = 400 for EXTRACTED Block 1 of File 1.

1 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 1.

133 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 1.

266 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 1.

399 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 1.

451 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 1.

 * NOTE * Enter DATA BASE Files from Coarse to Fine Grid in Order.

Specify DATA BASE Input File # 2 to be Converted to Column Formatted Data.
 * NOTE 1: The TECPLOT File is a PREPLOT Input file !!!
 * NOTE 2: The EXODUS II File is an Unformatted Binary file !!!
 * NOTE 3: The Following Files in the Input Family are Present:
        maw_ran_exo.1
        maw_ran_exo.2
        maw_ran_exo.3
        maw_ran_exo.4
*** WHICH ONE DO YOU WISH TO USE ? ***
ENTER HERE (CR->maw_ran_exo.2):

ICHS = 22

Specify Output File Name to be Created by Extracting Information from the Input File.
 * NOTE * For Computational Accuracy Evaluations, Output File Names Created here
        MUST be the Same as Input File Names Requested Later by ORDER Subprogram !!!
ENTER HERE (CR->maw_ran_exo_2_extract_2.o):

Number of Nodes = 1701 in File 2.
```

Element Block/Zone Numbers: 1

Current Extraction # 1 Reading Element Block 1.

Number of INPUT File Times = 1.

Time Plane Values: 1: 0.0000E+00

NTPLANE(NTPLANES) = 1

Maximum EXODUS Time Plane of Interest = 0.000000E+00 seconds.

Following are the Header Lines from Input File 2.

Refined Mesh 0003

Variables: X Y zcoor VX VY Pres

INPUT File Time Plane 1 Read from EXTRACTED Block 1 of File 2.

QUAD Elements = 1600 for EXTRACTED Block 1 of File 2.

1 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 2.

533 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 2.

1066 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 2.

1599 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 2.

1701 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 2.

 * NOTE * Enter DATA BASE Files from Coarse to Fine Grid in Order.

Specify DATA BASE Input File # 3 to be Converted to Column Formatted Data.
 * NOTE 1: The TECPLOT File is a PREPLOT Input file !!!
 * NOTE 2: The EXODUS II File is an Unformatted Binary file !!!
 * NOTE 3: The Following Files in the Input Family are Present:
    maw_ran_exo.1
    maw_ran_exo.2
    maw_ran_exo.3
    maw_ran_exo.4
*** WHICH ONE DO YOU WISH TO USE ? ***
ENTER HERE (CR->maw_ran_exo.3):

ICHS = 22

Specify Output File Name to be Created by Extracting Information from the Input File.

```
 * NOTE * For Computational Accuracy Evaluations, Output File Names Created here
        MUST be the Same as Input File Names Requested Later by ORDER Subprogram !!!
ENTER HERE (CR->maw_ran_exo_3_extract_3.o):

Number of Nodes = 6601 in File 3.

Element Block/Zone Numbers:     1

Current Extraction # 1 Reading Element Block 1.

Number of INPUT File Times = 1.

Time Plane Values:     1: 0.0000E+00

NTPLANE(NTPLANES) =      1

Maximum EXODUS Time Plane of Interest =   0.000000E+00 seconds.

Following are the Header Lines from Input File 3.

Refined Mesh 0005

Variables: X        Y        zcoor     VX       VY       Pres

INPUT File Time Plane 1 Read from EXTRACTED Block 1 of File 3.

QUAD Elements = 6400 for EXTRACTED Block 1 of File 3.

1 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 3.

2133 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 3.

4266 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 3.

6399 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 3.

6601 Grid Points Stored for EXTRACTED Block 1 at EXTRACTED Time Plane 1 of File 3.

*** EXTRACT Execution Ended on: Fri Sep  1 15:45:51 2000 ***

*** ORDER Subroutine Re-Entered on: Fri Sep  1 15:45:51 2000 ***

Specify the ORDER Output File Name.
ENTER HERE (CR->order.o):

Specify the ORDER Tecplot File Name.
ENTER HERE (CR->order_tec.o):
```

```
Specify Tecplot File Title.
ENTER HERE (CR->"Numerical Domain Order"):

Do You Wish to Process ALL Dependent Variables in the ORDER Input Files ?
ENTER HERE [YES/CR OR NO]:

Number of Columns in ORDER Input File 1 =  6

Number of Columns in ORDER Input File 2 =  6

Number of Columns in ORDER Input File 3 =  6

Specify the Constant Grid Refinement Factor.
 * NOTE 1: This DOES NOT have to be an Integer !!!
 * NOTE 2: Between any 2 Grids, A Single Array or Both May be Refined at a time !!!
 * NOTE 3: A Single Refined Array may be in the I or J Direction for the 3 grids !!!
ENTER HERE [CR->2]:

Specify the Numerical Solution Formal Order of Accuracy.
ENTER HERE [CR->2]:

Specify Extrapolation Option for Calculating Error:
  1-Calculate Error using FORMAL Order of Accuracy
  2-Calculate Error using COMPUTED Order of accuracy
ENTER HERE (CR->1):

Specify the Grid for which the Error is Calculated:
  1-Fine Grid
  2-Medium Grid
  3-Coarse Grid
ENTER HERE (CR->1):

Specify the Output Error Type:
  1-Error Output in % Difference Relative to Reference.
  2-Error Output in Simple Difference from Reference.
ENTER HERE (CR->1):

Is there a BENCHMARK Program Available for Comparison to the Numerical Results ?
 * NOTE * This Could be Analytical, Data, etc !!!
ENTER HERE [YES OR NO/CR]:

Matching Grid Points in Files 1, 2, & 3 for ZONE/BLOCK 1 at EXTRACTED Time 1.

1 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.

90 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.

180 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.
```

```
    270 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.

    360 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.

    450 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.

    451 Grid Points Matched in EXTRACTED Zone/Block 1 at EXTRACTED Time 1.

    Variable Name Format -->
    '("VARIABLES = " A4,A4,A8,A7,A7,A9,A9,A9,A11,A5,A5,A7)'

    F2-F1 Equals 0 Denominator Warnings Logged for 143 Order Calculations in Zone 1.

    Log Function Argument LE 0 Warnings Logged for 272 Order Calculations in Zone 1.

    Do You Wish to Run ORDER Again ?
    ENTER HERE [YES OR NO/CR]:

    *** ORDER Execution Ended on: Fri Sep  1 15:46:18 2000 ***


    Do You Wish to Run VIVID Again ?
    ENTER HERE [YES OR NO/CR]:

    *** VIVID Elapsed CPU Usage =   20.695 seconds ***

    *** VIVID Execution Ended on: Fri Sep  1 15:46:19 2000 ***

irn $
```

Appendix C. Sample Control Input Files

This appendix provides the sample control namelist input files for problems solved by VIVID. Included are the input files necessary to handle structured and unstructured order of accuracy analyses. EXODUS and TECPLOT input examples are illustrated. The first four examples are for standard evaluations without the use of BENCHMARK validation subroutines. Samples five and six make use of the flat duct and the two-dimensional conduction routines respectively. Example six shows a transient conduction controlling input file. Finally, the seventh example control file shows a time variant analysis for unstructured grid EXODUS input files.

C.1 Control File for Structured Grid TECPLOT Input Files

```
$VIVID1
 IVOPT=1
 N_N=     8241
 N_E=     8060
 MEM_TOT=    5216767
 NUM_T=    1
 NUM_D=    2
$END
$ORDER1
 DOMAIN='Y'
 ZERO_DIF=    1.00000E-07
 GRID='S'
 NIF=3
 INTERP='N'
 EXTRUN='Y'
$END
$EXTRACT1
 IDOPT=1
 ZONETYPE='S'
 ALLDEP='Y'
 INFILE_E(1)='duct_TFG.2                    '
 OUTFILE_E(1)='duct_TFG_2_extract_1.o       '
 IZONES= 1
 IZONESX= 1
 IZONE(1)=  1
 ZVAR='Y'
 IPLANE(1, 1)=   1  51
 JPLANE(1, 1)=   1  11
 KPLANE(1, 1)=   1   1
 NPTS(1, 1)=      561
$END
$EXTRACT2
 INFILE_E(2)='duct_TFG.4                    '
 OUTFILE_E(2)='duct_TFG_4_extract_2.o       '
```

```
 IPLANE(1, 1)=   1 101
 JPLANE(1, 1)=   1  21
 KPLANE(1, 1)=   1   1
 NPTS(2, 1)=      2121
$END
$EXTRACT3
 INFILE_E(3)='duct_TFG.8                    '
 OUTFILE_E(3)='duct_TFG_8_extract_3.o       '
 IPLANE(1, 1)=   1 201
 JPLANE(1, 1)=   1  41
 KPLANE(1, 1)=   1   1
 NPTS(3, 1)=      8241
$END
$ORDER2
 INFILE_O(1)='duct_TFG_2_extract_1.o       '
 INFILE_O(2)='duct_TFG_4_extract_2.o       '
 INFILE_O(3)='duct_TFG_8_extract_3.o       '
 OUTFILE_O='order.o                        '
 TECFILE='order_tec.o                      '
 TITLE='"Numerical Domain Order"                                    '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IXOPT=1
 IGERR=2
 ITERR=1
 BMRF='N'
 IRUN='N'
$END
$VIVID2
 IRUN='N'
$END
```

C.2 Control File for Unstructured Grid TECPLOT Input Files

```
$VIVID1
 IVOPT=1
 N_N=      6601
 N_E=      6440
 MEM_TOT=     4174247
 NUM_T=    1
 NUM_D=    2
$END
$ORDER1
 DOMAIN='Y'
 ZERO_DIF=    1.00000E-07
 GRID='U'
 NIF=3
```

```
 INTERP='N'
 EXTRUN='Y'
$END
$EXTRACT1
 IDOPT=2
 ALLDEP='Y'
 INFILE_E(1)='maw_ran_exo.1                '
 OUTFILE_E(1)='maw_ran_exo_1_extract_1.o    '
 IZONES= 1
 IZONESX= 1
 IZONE(1)=  1
$END
$READEXO1
 ZVAR='Y'
 NPTS(1, 1)=      451
$END
$EXTRACT2
 INFILE_E(2)='maw_ran_exo.2                '
 OUTFILE_E(2)='maw_ran_exo_2_extract_2.o    '
$END
$READEXO2
 NPTS(2, 1)=     1701
$END
$EXTRACT3
 INFILE_E(3)='maw_ran_exo.3                '
 OUTFILE_E(3)='maw_ran_exo_3_extract_3.o    '
$END
$READEXO3
 NPTS(3, 1)=     6601
$END
$ORDER2
 INFILE_O(1)='maw_ran_exo_1_extract_1.o    '
 INFILE_O(2)='maw_ran_exo_2_extract_2.o    '
 INFILE_O(3)='maw_ran_exo_3_extract_3.o    '
 OUTFILE_O='order.o                        '
 TECFILE='order_tec.o                      '
 TITLE='"Numerical Domain Order"                                    '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IXOPT=1
 IGERR=1
 ITERR=1
 BMRF='N'
 IRUN='N'
$END
$VIVID2
 IRUN='N'
$END
```

C.3 Control File for Unstructured Grid Single Block EXODUS Input Files

```
$VIVID1
 IVOPT=1
 N_N=     6601
 N_E=     6440
 MEM_TOT=    4174247
 NUM_T=   1
 NUM_D=   2
$END
$ORDER1
 DOMAIN='Y'
 ZERO_DIF=   1.00000E-07
 GRID='U'
 NIF=3
 INTERP='N'
 EXTRUN='Y'
$END
$EXTRACT1
 IDOPT=2
 ALLDEP='Y'
 INFILE_E(1)='maw_ran_exo.1              '
 OUTFILE_E(1)='maw_ran_exo_1_extract_1.o    '
 IZONES= 1
 IZONESX= 1
 IZONE(1)=  1
$END
$READEXO1
 ZVAR='Y'
 NPTS(1, 1)=      451
$END
$EXTRACT2
 INFILE_E(2)='maw_ran_exo.2              '
 OUTFILE_E(2)='maw_ran_exo_2_extract_2.o    '
$END
$READEXO2
 NPTS(2, 1)=     1701
$END
$EXTRACT3
 INFILE_E(3)='maw_ran_exo.3              '
 OUTFILE_E(3)='maw_ran_exo_3_extract_3.o    '
$END
$READEXO3
 NPTS(3, 1)=     6601
$END
$ORDER2
 INFILE_O(1)='maw_ran_exo_1_extract_1.o    '
 INFILE_O(2)='maw_ran_exo_2_extract_2.o    '
 INFILE_O(3)='maw_ran_exo_3_extract_3.o    '
```

```
 OUTFILE_O='order.o                    '
 TECFILE='order_tec.o                  '
 TITLE='"Numerical Domain Order"                                    '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IXOPT=1
 IGERR=1
 ITERR=1
 BMRF='N'
 IRUN='N'
$END
$VIVID2
 IRUN='N'
$END
```

C.4 Control File for Unstructured Grid Multiple Block EXODUS Input Files

```
$VIVID1
 IVOPT=1
 N_N=     11663
 N_E=     10632
 MEM_TOT=    48695980
 NUM_T=    1
 NUM_D=    3
$END
$ORDER1
 DOMAIN='Y'
 GRID='U'
 NIF=3
 IZMOPT=1
 INTERP='N'
 EXTRUN='Y'
$END
$EXTRACT1
 IDOPT=2
 ALLDEP='Y'
 INFILE_E(1)='aaa.exo                  '
 OUTFILE_E(1)='aaa_exo_extract_1.o          '
 ZONES='ALL'
 IZONES= 9
 IZONESX= 9
$END
$READEXO1
 NTIMES=    102
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=   102
```

```
 TIMESX(1)=  1.0007E+05
 IZONE(1)=  9  1  2  3  4  5  6  7  8
 ZVAR='Y'
 NPTS(1, 9)=      243
 NPTS(1, 1)=     8279
 NPTS(1, 2)=     1636
 NPTS(1, 3)=     1258
 NPTS(1, 4)=      324
 NPTS(1, 5)=     1023
 NPTS(1, 6)=      250
 NPTS(1, 7)=      225
 NPTS(1, 8)=      243
$END
$EXTRACT2
 INFILE_E(2)='aaa.exo                    '
 OUTFILE_E(2)='aaa_exo_extract_2.o          '
$END
$READEXO2
 NTIMES=    102
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=   102
 TIMESX(1)=  1.0007E+05
 IZONE(1)=  9  1  2  3  4  5  6  7  8
 NPTS(2, 9)=      243
 NPTS(2, 1)=     8279
 NPTS(2, 2)=     1636
 NPTS(2, 3)=     1258
 NPTS(2, 4)=      324
 NPTS(2, 5)=     1023
 NPTS(2, 6)=      250
 NPTS(2, 7)=      225
 NPTS(2, 8)=      243
$END
$EXTRACT3
 INFILE_E(3)='aaa.exo                    '
 OUTFILE_E(3)='aaa_exo_extract_3.o          '
$END
$READEXO3
 NTIMES=    102
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=   102
 TIMESX(1)=  1.0007E+05
 IZONE(1)=  9  1  2  3  4  5  6  7  8
 NPTS(3, 9)=      243
```

```
 NPTS(3, 1)=      8279
 NPTS(3, 2)=      1636
 NPTS(3, 3)=      1258
 NPTS(3, 4)=       324
 NPTS(3, 5)=      1023
 NPTS(3, 6)=       250
 NPTS(3, 7)=       225
 NPTS(3, 8)=       243
$END
$ORDER2
 INFILE_O(1)='aaa_exo_extract_1.o          '
 INFILE_O(2)='aaa_exo_extract_2.o          '
 INFILE_O(3)='aaa_exo_extract_3.o          '
 OUTFILE_O='order.o                  '
 TECFILE='order_tec.o                '
 TITLE='"Numerical Domain Order"                                        '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IEOPT=1
 IGERR=1
 ITERR=1
 BMRF='N'
 IRUN='N'
$END
$VIVID2
 IRUN='N'
$END
```

C.5 Control File for Unstructured Grid EXODUS Input Files with Analytic Flat Duct Verification

```
$VIVID1
 IVOPT=1
 N_N=    20000
 N_E=    19718
 MEM_TOT=    44411296
 NUM_T=    1
 NUM_D=    2
$END
$ORDER1
 DOMAIN='Y'
 ZERO_DIF=    1.00000E-07
 GRID='U'
 NIF=3
 TWODIM='Y'
 IZMOPT=1
 INTERP='N'
 EXTRUN='Y'
```

```
$END
$EXTRACT1
 IDOPT=2
 ALLDEP='Y'
 INFILE_E(1)='exo.2                    '
 OUTFILE_E(1)='exo_2_extract_1.o            '
 ZONES='ALL'
 IZONES= 1
 IZONESX= 1
$END
$READEXO1
 NTIMES=     3
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=     3
 TIMESX(1)=  5.0000E+01
 IZONE(1)=  1
 ZVAR='Y'
 NPTS(1, 1)=      861
$END
$EXTRACT2
 INFILE_E(2)='exo.3                    '
 OUTFILE_E(2)='exo_3_extract_2.o            '
$END
$READEXO2
 NTIMES=     3
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=     3
 TIMESX(1)=  5.0000E+01
 IZONE(1)=  1
 NPTS(2, 1)=     3321
$END
$EXTRACT3
 INFILE_E(3)='exo.4                    '
 OUTFILE_E(3)='exo_4_extract_3.o            '
$END
$READEXO3
 NTIMES=     3
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=     3
 TIMESX(1)=  5.0000E+01
 IZONE(1)=  1
 NPTS(3, 1)=    13041
$END
```

```
$ORDER2
 INFILE_O(1)='exo_2_extract_1.o          '
 INFILE_O(2)='exo_3_extract_2.o          '
 INFILE_O(3)='exo_4_extract_3.o          '
 OUTFILE_O='order.o                    '
 TECFILE='order_tec.o                  '
 TITLE='"Numerical Domain Order"                                    '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IXOPT=1
 IGERR=1
 ITERR=1
 BMRF='Y'
 ANAERR='Y'
 ANAORD='N'
$END
$BENCHMARK1
 IUOPT=1
$END
$FLAT_DUCT1
 IV=3
 H=   1.000000E+00
 IVP=1
 UM=   1.000000E+00
 RE=   2.000000E+01
 PIN=   4.100000E+00
 RHO=   1.000000E+00
 IYCS=1
$END
$FLAT_DUCT2
 IV=1
$END
$FLAT_DUCT3
 IV=2
$END
$FLAT_DUCT4
 IV=4
$END
$FLAT_DUCT5
 IV=4
$END
$FLAT_DUCT6
 IV=4
$END
$ORDER3
 IRUN='N'
$END
$VIVID2
```

```
 IRUN='N'
$END
```

C.6 Control File for Unstructured Grid EXODUS Input Files with Analytic Transient Heat Conduction Verification

```
$VIVID1
 IVOPT=1
 N_N=     1681
 N_E=     1600
 MEM_TOT=    3716859
 NUM_T=    1
 NUM_D=    2
$END
$ORDER1
 DOMAIN='Y'
 ZERO_DIF=    1.00000E-07
 GRID='U'
 NIF=3
 TWODIM='Y'
 IZMOPT=1
 INTERP='N'
 EXTRUN='Y'
$END
$EXTRACT1
 IDOPT=2
 ALLDEP='Y'
 INFILE_E(1)='mesh1_rec_2x1_plot.exo       '
 OUTFILE_E(1)='1.o                         '
 ZONES='ALL'
 IZONES= 1
 IZONESX= 1
$END
$READEXO1
 NTIMES=     2
 TZDIM='N'
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=     2
 TIMESX(1)=  0.0000E+00
 IZONE(1)=  1
 ZVAR='Y'
 NPTS(1, 1)=      121
$END
$EXTRACT2
 INFILE_E(2)='mesh2_rec_2x1_plot.exo       '
 OUTFILE_E(2)='2.o                         '
$END
```

```
$READEXO2
 NTIMES=      2
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=     2
 TIMESX(1)=  0.0000E+00
 IZONE(1)=  1
 NPTS(2, 1)=       441
$END
$EXTRACT3
 INFILE_E(3)='mesh3_rec_2x1_plot.exo        '
 OUTFILE_E(3)='3.o                          '
$END
$READEXO3
 NTIMES=      2
 ALLTIMES='N'
 ITOPT=1
 NTPLANES=  1
 NTPLANE(1)=     2
 TIMESX(1)=  0.0000E+00
 IZONE(1)=  1
 NPTS(3, 1)=      1681
$END
$ORDER2
 INFILE_O(1)='1.o                           '
 INFILE_O(2)='2.o                           '
 INFILE_O(3)='3.o                           '
 OUTFILE_O='order.o                         '
 TECFILE='order_tec.o                       '
 TITLE='"Numerical Domain Order"                                        '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IXOPT=1
 IGERR=1
 ITERR=1
 BMRF='Y'
 ANAERR='Y'
 ANAORD='N'
$END
$BENCHMARK1
 IUOPT=3
 XL= 1.00000E-01
 YH= 5.00000E-02
 ZW= 0.00000E+00
$END
$CONDUCTION_2D1
 IV=1
```

```
 INOPT=1
 HCID='x21b11y23b01g1              '
 TOL=   1.000000E-06
$END
$ORDER3
 IRUN='N'
$END
$VIVID2
 IRUN='N'
$END
```

C.7 Control File for Unstructured Grid EXODUS Transient Input Files

```
$VIVID1
 IVOPT=1
 N_N=     2121
 N_E=     2030
 MEM_TOT=    3841873
 NUM_T=    4
 NUM_D=    2
 TZDIM='Y'
$END
$ORDER1
 DOMAIN='Y'
 ZERO_DIF=    1.00000E-07
 GRID='U'
 NIF=3
 INTERP='N'
 EXTRUN='Y'
$END
$EXTRACT1
 IDOPT=2
 ALLDEP='Y'
 INFILE_E(1)='maw_tran_exo.1              '
 OUTFILE_E(1)='maw_tran_exo_1_extract_1.o    '
 IZONES= 1
 IZONESX= 1
 IZONE(1)=  1
$END
$READEXO1
 NTIMES=     4
 ALLTIMES='N'
 ITOPT=2
 NTPLANES=  4
 NTPLANE(1)=     1     2     3     4
 TIMESX(1)=  0.0000E+00  4.0000E-03  8.0000E-03  1.2000E-02
 TVAR='N'
 ZVAR='Y'
 NPTS(1, 1)=     2121
```

```
$END
$EXTRACT2
 INFILE_E(2)='maw_tran_exo.2              '
 OUTFILE_E(2)='maw_tran_exo_2_extract_2.o   '
$END
$READEXO2
 NTIMES=     7
 ALLTIMES='N'
 ITOPT=2
 NTPLANES=  4
 NTPLANE(1)=    1    3    5    7
 TIMESX(1)=  0.0000E+00  4.0000E-03  8.0000E-03  1.2000E-02
 TVAR='N'
 NPTS(2, 1)=     2121
$END
$EXTRACT3
 INFILE_E(3)='maw_tran_exo.3              '
 OUTFILE_E(3)='maw_tran_exo_3_extract_3.o   '
$END
$READEXO3
 NTIMES=    13
 ALLTIMES='N'
 ITOPT=2
 NTPLANES=  4
 NTPLANE(1)=    1    5    9   13
 TIMESX(1)=  0.0000E+00  4.0000E-03  8.0000E-03  1.2000E-02
 TVAR='N'
 NPTS(3, 1)=     2121
$END
$ORDER2
 INFILE_O(1)='maw_tran_exo_1_extract_1.o   '
 INFILE_O(2)='maw_tran_exo_2_extract_2.o   '
 INFILE_O(3)='maw_tran_exo_3_extract_3.o   '
 OUTFILE_O='order.o                    '
 TECFILE='order_tec.o                  '
 TITLE='"Numerical Domain Order"                                    '
 ALLDEP='Y'
 RF= 2.00
 PF= 2.00
 IXOPT=1
 IGERR=1
 ITERR=1
 BMRF='N'
 IRUN='N'
$END
$VIVID2
 IRUN='N'
$END
```

Appendix D. Dimension & Conduction Files

This appendix contains files relevant to the VIVID code. The first file is the Fortran common block include file. This file sets the dimensions of the code for critical program variables. It is used to resize the program to fit specific purposes prior to compilation. The second file is the namelist input file for a run of the two-dimensional heat conduction subroutine. This particular file is for a transient run.

D.1 Fortran Common Block Include File

```
C
      PARAMETER (N_I=1000, N_J=1000, N_K=10, N_V=15, N_Z=10, N_F=3,
     +          N_T=99, NPE=8, NTV=3*(N_V-3)+3, M_M=100000)
C
      COMMON /CONTROL/ ZERO_DIF,MAXARG,N_N,N_E,NUM_D,NUM_T,NTX
C
```

D.2 Heat Conduction Subroutine Input file

```
 $INPUT
  HCID='x21b11y23b01g1            '
  TINI=   2.500000E+01
  TIME_F=   1.000000E+04
  COND=   4.000000E-01
  VOLSRC=   1.353000E+05
  DENS=   7.000000E+03
  SPECHT=   5.000000E+02
  HFLUX=   3.500000E+03
  TEMPBC=   1.000000E+03
  TCONV=   2.500000E+01
  HCOEFF=   6.000000E+01
  TOL=   1.000000E-06
  XMAX= 1.00000E-01
  YMAX= 5.00000E-02
 $END
```

Appendix E. FCV-FAM Numerical Program

This appendix contains a description of the Finite Control Volume Face Assembly Method (FCV-FAM) numerical heat conduction program. It was written by the lead author of this report for specific use in the development of VIVID. The description below was derived from the development log at the beginning of the source code file.

FCV-FAM is a Finite Control Volume (FCV) computational numerical program that is based on the Face Assembly Method (FAM) introduced to the author by R. J. Cochran. The current version of the program solves the steady state heat conduction problem in two dimensions. Four solvers are included in the code: two direct and two iterative. The direct solvers are Gauss-Jordan and LU Factorization. The iterative solvers are Gauss-Seidel and Jacobi. All 4 solvers have been demonstrated as second order accurate making use of linear interpolating polynomials in the matrix fill subroutine.

This is a cell centered program. It currently handles specified surface temperature and specified heat flux boundary conditions. These conditions can be specified in any fashion on the boundaries. The boundary conditions can be input as distributions on the surfaces or as constants along the surface. An initial temperature is specified to begin the solution with temperature dependent material properties.

Geometry options are currently limited to rectangular and axisymmetric cylindrical geometries. Non-Orthogonal issues have not been dealt with in the initial version. It has two internal gridding options to generate computational meshes. It has a simple uniform gridding option and a geometric progression gridding option to handle high gradient problems.

The code includes the convection difference operators to the energy equation solution subroutine derived from the general transport equation. This implementation included the central and upwind difference operators. Specified and zero-gradient (i.e. fully developed) velocity boundary conditions were included in its capabilities. Varying velocity distributions on the boundaries are allowed. In addition, the code contains the ability to include uniform internal heat generation. The generation term appears along the main diagonal in the matrix formulation.

Input is handled with a NAMELIST format approach. This allows for intuitive descriptive input files. The code allows inputs of the material properties as a function of temperature. As such, it allows for multi-pass solutions to converge on a consistent in-depth and surface temperature solution. This allows for fully coupling the in-depth temperature solution with the surface boundary conditions through a surface energy balance subroutine. The surface energy balance subroutine takes into account surface radiative heat exchange with a background ambient radiation temperature.

A number of output files are generated for analysis, code debugging, and prob-

lem visualization. A TECPLOT block format file is generated for visualization of the temperature field contours. The code contains a restart capability to handle long run time problems. A convergence accelerator option has been included. This option cuts convergence time by about 20%.

Distribution

1 Prof. Graham de Vahl Davis
School of Mechanical and Mfg. Engineering
University of NSW
Sydney NSW 2052, Australia

1 Prof. Joel H. Ferziger
Department of Mechanical Engineering
Stanford University
Stanford, CA 94305-3030

1 Dr. Charles L. Merkle
Computational Mechanics Group
University of Tennessee Space Institute
BH Goethert Parlway MS-10
Tullahoma, TN 37388-9700

1 Dr. Patrick J. Roache
1108 Mesa Loop NW
Los Lunas, NM 87031

1 Mr. James L. Thomas
NASA Langley Research Center
Mail Stop 128
Hampton, VA 23665

1 Prof. Chris J. Roy
Department of Aerospace Engineering
211 Aerospace Engineering Bldg.
Auburn University, AL 36849

1	MS 9042	8345	G. E. Evans
1	MS 9042	8345	W. G. Houf
1	MS 9042	8345	C. D. Moen
1	MS 9042	8345	W. S. Winters
1	MS 9011	8950	J. C. Meza
1	MS 0841	9100	C. W. Peterson
1	MS 0841	9100	D. K. Gartling
1	MS 0826	9113	S. N. Kempka (route to staff)
1	MS 0834	9114	P. R. Schunk

1	MS 0834	9114	M. J. Martinez
1	MS 0834	9114	J. S. Lash
1	MS 0834	9114	A. R. Roach
2	MS 0825	9115	B. Hassan (route to staff)
20	MS 0825	9115	D. L. Potter
1	MS 0825	9115	M. F. Barone
1	MS 0825	9115	B. F. Blackwell
1	MS 0825	9115	F. G. Blottner
1	MS 0825	9115	R. B. Bond
1	MS 0825	9115	L. J. Dechant
1	MS 0825	9115	D. W. Kuntz
1	MS 0825	9115	J. L. Payne
1	MS 0825	9115	W. P. Wolfe
1	MS 0836	9116	E. Hertel (route to staff)
1	MS 0836	9116	R. E. Hogan
1	MS 0836	9117	R. O. Griffith (route to staff)
1	MS 0836	9117	J. E. Brockmann
1	MS 0847	9120	P. J. Wilson
1	MS 0847	9124	J. M. Redmond (route to staff)
1	MS 0824	9130	J. L. Moya (route to staff)
1	MS 0824	9130	T. Y. Chu
1	MS 0828	9132	L. A. Gritzo
1	MS 0828	9132	V. F. Nicolette
1	MS 0828	9132	S. R. Tieszen
2	MS 0828	9133	M. Pilch
1	MS 0828	9133	K. J. Dowding
1	MS 0828	9133	A. R. Black
1	MS 0828	9133	W. L. Oberkampf
1	MS 0827	9140	H. S. Morgan
1	MS 0835	9141	E. A. Boucheron (route to staff)
1	MS 0835	9141	A. A. Lorber
1	MS 0835	9141	A. P. Domino
1	MS 0832	9142	K. F. Alvin (route to staff)
1	MS 0847	9142	S. W. Attaway
1	MS 0847	9142	M. L. Blanford
1	MS 0847	9142	S. W. Key
1	MS 0827	9143	J. R. Stewart
1	MS0321	9200	W. J. Camp
1	MS 0819	9211	T. G. Trucano
1	MS 0819	9231	S. P. Burns
1	MS 0819	9231	M. Christon
1	MS 0819	9231	R. Summers
1	MS 0316	9233	S. S. Dosanjh
1	MS 0316	9233	C. C. Ober
1	MS 0316	9233	T. M. Smith
1	MS 1111	9233	A. G. Salinger
1	MS 0316	9233	R. C. Schmidt
1	MS 1111	9233	J. N. Shadid

1	MS 1179	15340	J. R. Lee
1	MS 1162	15422	M. A. McWherter-Payne
1	MS 9018	8940-2	Central Technical Files
2	MS 0899	9616	Technical Library