

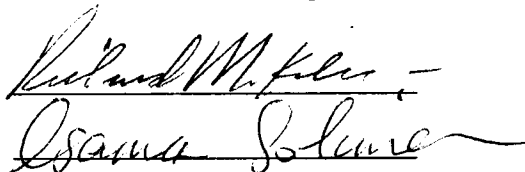
To the Graduate Council:

I am submitting herewith a thesis written by Leif E. Wilkening entitled "A CFD Preprocessor Unifying Fluid/Thermal Simulation Specifications." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science and Mechanics.

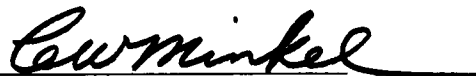


Allen J. Baker, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission to extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in the thesis for financial gain shall not be allowed without my written permission.

Signature Lois E. Wilkerson

Date 7/26/91

A CFD LABORATORY PREPROCESSOR UNIFYING FLUID/THERMAL SIMULATION SPECIFICATIONS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Leif E. Wilkening

August 1991

ACKNOWLEDGEMENTS

During my thesis work, I consider myself fortunate to have been under the guidance of Professor Jerry Baker for his exacting, yet empathetic, way of teaching. To Professors Richard Kelso and Osama Soliman I also owe my appreciation. My graduate experience was further enriched by fellow graduate students Subrata Roy and Alan Luchuk.

On a personal note, I am indebted to Susan, who married me in the midst of this undertaking and was my constant help, and to my family and friends for their encouragement. And most of all, I thank the One who is the α and ω , the beginning and the end (Rev. 22:13).

ABSTRACT

Computational fluid dynamics (CFD) seeks to simulate the phenomena of fluid flow using computers. As this field has advanced in recent decades, many different algorithms and codes have resulted, making comparative evaluations difficult. As a step to remedy this situation, a preprocessor¹ has been modified and extended that can efficiently instruct a variety of two-dimensional incompressible-thermal Navier-Stokes codes. Using more efficient data input streaming and handling, the size of the program has been reduced by 40% while significantly enhancing its capabilities. Utilities are included to enable computation of multiple solutions that speed iterative convergence for large Reynolds and Grashof number flows. The user easily constructs a computational grid using an algebraic macro domain procedure, for either cartesian or curved-sided quadrilaterals. Boundary conditions include walls, symmetry planes, inlets, and outlets, each definable using a minimum of physical data. Operation of the preprocessor program is explained herein with several illustrative examples.

¹J.B. Woods, *A Workstation-Based Engineering Expert System Environment for a CFD Laboratory Navier-Stokes Computer Code*, MS Thesis, University of Tennessee, Knoxville, TN, 1989.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. EQUATION SYSTEMS	3
Navier-Stokes Equations	3
Continuity Enforcement	6
3. BOUNDARY CONDITIONS	10
Primitive Variables	10
Streamfunction-Vorticity	11
4. CFD SOLUTION ALGORITHM	13
5. SOLUTION STRATEGIES	18
Time Stepping Procedure	18
Continuation Parameters	19
Adaptive Remeshing	19
6. PREPROS OPERATION	21
Setup Parameters	22
Grid Construction	23
Initial Condition	29
Boundary Conditions	30
Inlets	32
Walls	33
Continuation Parameters	35
Integration Parameters	37
7. EXAMPLE PROBLEMS	39
Driven Cavity	39
Non-Uniform Driven Cavity	48
Thermal Cavity	55
Kitchen Hood	63
8. CONCLUSIONS	82
LIST OF REFERENCES	83
VITA	85

LIST OF FIGURES

1. Primitive Variables Boundary Conditions	11
2. Transformed Variables Boundary Conditions	11
3. Rectangular Macro Coordinates	26
4. General Quadrilateral Macro Coordinates	26
5. Example of Node Row Specification	31
6. Example of Inlet and Wall Node Numbering	31
7. Driven Cavity Uniform 16x16 Grid	41
8. Driven Cavity 16x16 Boundary Plot	45
9. Driven Cavity Uniform 16x16 Grid Solution	47
10. Driven Cavity Non-Uniform 16x16 Grid	50
11. Driven Cavity Non-Uniform 16x16 Grid Solution	54
12. Thermal Cavity Solution at $Ra=7 \times 10^6$	62
13. Thermal Cavity Solution at $Ra=7 \times 10^6$ with Artificial Diffusion=0.1	62
14. Kitchen Hood Problem	63
15. Kitchen Hood Macro Construction	64
16. Incorrect Kitchen Hood Grid	67
17. Corrected Kitchen Hood Grid	69
18. Kitchen Hood Boundary Plot	78
19. Hood Velocity Distribution at $Re=1000$	80
20. Hood Temperature Distribution at $Re=1000$	81

LIST OF SYMBOLS

α	= temporal artificial diffusion coefficient
β	= thermal expansion coefficient; spatial artificial diffusion coefficient
Δ, δ	= increment in a variable
$\Delta \eta_e$	= length measure of element e
ε	= convergence requirement
η_i	= local coordinate system, $i=1,2$
Θ	= non-dimensional temperature = $(T - T_{min}) / (T_{max} - T_{min})$
θ	= integration implicitness
λ	= Lagrange multiplier
ν	= kinematic viscosity
ρ_0	= fluid uniform density
τ	= incremental area
ϕ	= continuity satisfaction measure
$\phi(x)$	= interpolation function
Ψ	= semi-discretized streamfunction
ψ	= streamfunction
Ω	= semi-discretized vorticity; problem domain
ω	= vorticity
Ar	= Archimedes number = Gr/Re^2
C_A	= mass or mole fraction of inert species A
c_p	= constant pressure specific heat
D_{AB}	= binary diffusion coefficient of inert species A
$\{F(Q)\}$	= load vector of variable Q
f	= flux
Gr	= Grashof number = $g\beta(T_{max} - T_{min})L^3/\nu^2$
g	= gravitational acceleration
$\mathbf{g}$	= gravitational unit vector
J	= jacobian matrix derivative
$\mathbf{j}$	= unit vector in vertical direction
k	= thermal conductivity
$\mathbf{k}$	= unit vector perpendicular to velocity field
L	= length scale
$L(\cdot)$	= partial differential equation
N_k	= interpolating polynomial of degree k
n	= direction normal to surface
$\mathbf{n}$	= unit normal vector
PR	= node spacing progression ratio
Pr	= Prandtl number = $c_p\rho_0\nu$
P	= kinematic pressure = p/ρ
p	= pressure
Q	= semi-discrete solution variable
q	= solution variable
Re	= Reynolds number = UL/ν
s	= direction tangential to surface
s_q	= source term for q

Sc	= Schmidt number = ν_i/D_{AB}
T	= dimensional temperature
t	= time
u	= velocity vector
u	= x component of velocity
v	= y component of velocity
W	= weight function expansion coefficient
w(x)	= weight function
x	= spatial coordinate system
x	= spatial coordinate in horizontal direction
y	= spatial coordinate in vertical direction

Subscripts

n	= normal direction
p	= iteration index
r	= reference quantity
s	= tangential direction

Superscripts

a	= artificial diffusion
ef	= effective parameter
n	= time step index
T	= transpose of a matrix
t	= turbulent quantity

CHAPTER 1

INTRODUCTION

Central to the understanding of fluid mechanics achieved in the last century, and in the early and mid part of this century, was the fluid mechanics or aerodynamics experimental laboratory. The discoveries of Reynolds, Prandtl, and others are closely associated with physical experimentation. In the laboratory, observation of fluid behavior could support or deny hypotheses, and visibly demonstrate known theory to those mastering the field.

From these years of research, the partial differential equation system governing fluid mechanics is reasonably well understood, and a newer field has developed that seeks to model fluid phenomena by generating numerical solutions of these equations on computers. Since its beginnings in the 1920's, computational fluid dynamics (CFD) has evolved methods increasingly able to recreate the detailed physics of fluid flows.

As an aid to understanding the concepts of CFD, an environment is needed that fosters the same atmosphere of experimentation that fluid mechanics laboratories do. These "experiments" would not be physical but rather "computational". Consequently, one is then able to demonstrate the principles of fluid and thermal transport phenomena via computational simulation experiments.

It is the aim of this thesis to provide an interface to the computational tools, i.e., "CFD codes", to the user of CFD. This tool assists the user to correctly set up a problem definition, using minimal known physical data that a CFD code requires for correct execution. It frees the user to concentrate on interpreting the results rather than debugging the input file. The important segments are emphasized, using a command-name structure, leaving the busy work of correctly converting physical data to input to the preprocessor program, called "PREPROS".

Input files can be constructed using PREPROS for different CFD numerical solution

algorithms and codes, enabling their solutions and convergence behavior to be more easily compared, promoting their relative merits and utility to become evident.

The PREPROS code constructs the computational grid, assigns boundary and initial conditions, and defines parameters that control the solution sequences. This information is organized into the input file for a CFD program that actually solves the defined problem. The execution code then outputs the solution in several forms, e.g., a print file or a file formatted for graphically plotting the solution variable distributions. The Navier-Stokes solver codes currently anticipated by PREPROS include the streamfunction-vorticity method, the penalty method, and the Poisson continuity constraint method. The following chapters briefly summarize the governing equations and their required boundary conditions and solution algorithm strategy. Operation of PREPROS with a variety of examples is then fully detailed.

CHAPTER 2

EQUATION SYSTEMS

Navier-Stokes Equations

The starting point for any discussion of computational fluid dynamics is the governing equation set of interest. The subject is two-dimensional, turbulent thermal flow of a incompressible fluid, which is described by a Reynolds-averaged Navier-Stokes equation system. This consists of equations expressing conservation of mass, momentum and energy. A species transport equation is also included for study of inert pollutant diffusion. The induced Reynolds stress tensor represents the effects of turbulence in a distributed sense. The Boussinesq approximation models the bouyancy effects of the gravity body force.

Limiting the analysis to incompressibility does not exclude compressible fluids such as air. It merely limits the flow to small Mach number, that is, the scale velocity is small compared to the speed of sound, making density changes negligible. This problem class includes many common situations such as environmental flows and habitation enclosure ventilation.

The Reynolds-averaged Navier-Stokes equations in vector notation, where $L(\cdot)$ indicates a partial differential equation, is

$$L(\rho_e) = \nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$L(\mathbf{u}) = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot \left(\frac{1}{\text{Re}} + \mathbf{v}' \right) \nabla \mathbf{u} + \nabla P + \frac{\text{Gr}}{\text{Re}^2} \Theta \mathbf{g} = 0 \quad (2)$$

$$L(\Theta) = \frac{\partial \Theta}{\partial t} + (\mathbf{u} \cdot \nabla) \Theta - \nabla \cdot \left(\frac{1}{\text{Re}} \left(\frac{1}{\text{Pr}} + \frac{\mathbf{v}'}{\text{Pr}'} \right) \nabla \Theta - s_\Theta \right) = 0 \quad (3)$$

$$L(C_A) = \frac{\partial C_A}{\partial t} + (\mathbf{u} \cdot \nabla) C_A - \nabla \cdot \left(\frac{1}{\text{Re}} \left(\frac{1}{\text{Sc}} + \frac{\mathbf{v}^t}{\text{Sc}^t} \right) \nabla C_A - s_A \right) = 0 \quad (4)$$

where P is kinematic pressure (p divided by uniform density ρ_0). Superscript t signifies a turbulent parameter. The Reynolds, Grashof, Prandtl and Schmidt number definitions are

$$\text{Re} = \frac{U_r L_r}{\nu_r}, \quad \text{Gr} = \frac{g \beta (T_{\max} - T_{\min}) L_r^3}{\nu_r^2}, \quad \text{Pr} = \frac{c_p \rho_0 \nu_r}{k}, \quad \text{Sc} = \frac{\nu_r}{D_{AB}} \quad (5)$$

where subscript r signifies dimensional reference values. The energy equation dependent variable is potential temperature Θ which is defined as

$$\Theta = \frac{T - T_{\min}}{T_{\max} - T_{\min}} \quad (6)$$

Finally, C_A signifies the mass or mole fraction of an inert species A characterized by binary diffusion coefficient D_{AB} . Since concentration appears only in (4), it is decoupled from the rest of the equations and can be computed separately using any velocity solution.

The multiplier in the diffusion term in equations (2)-(4), the third term in each equation, controls the level of diffusion present in the calculation. Large Reynolds numbers relatively reduce diffusion effects in comparison to convection. Upon Reynolds averaging, the fluid Re effect is diminished by the presence of modelled turbulent diffusion processes. Turbulence is the random fluid motion on a scale smaller than supported by the grid resolution. Another reduction to Re is numerical diffusion, which is imbedded in most CFD algorithms and codes to assist convergence. Either can be viewed in terms of a corresponding Reynolds number, e.g., $\text{Re}^t = \nu^t / \nu_r$, the ratio of dimensional turbulent viscosity to reference fluid viscosity. The effective diffusion term in (2) then becomes

$$\left(\frac{1}{\text{Re}} + \nu^t \right) = \left(\frac{1}{\text{Re}} + \frac{\nu^t}{U_r L_r} \frac{\nu_r}{\nu_r} \right) = \frac{1}{\text{Re}} (1 + \text{Re}^t) = \frac{1}{\text{Re}} \left(1 + \frac{\nu^t}{\nu_r} \right) = \frac{1}{\text{Re}^{\text{eff}}} \quad (7)$$

This concept of "effective" Reynolds number Re^{ef} is valid for the energy and concentration equations as well with the standard assumption that the turbulent Prandtl and Schmidt numbers are equal to their laminar value, giving Pr or Sc as an additional divisor of (7).

Hence, upon introducing the artificial (numerical) viscosity Reynolds number Re^a , the effective Reynolds number, taking the inverse of (7), is

$$Re^{ef} = \frac{Re}{1 + \nu^t} = \frac{Re}{1 + Re^t + Re^a} \quad (8)$$

All sources of diffusion are thereby conceptually summed, so Re^{ef} is a single parameter that can be manipulated for comparative diffusive experiments.

In the absence of artificial viscosity, only turbulent effects modify Re . The turbulent Reynolds number Re^t is of order 10^2 for boundary layer dominated flows, a class which includes many problems of interest. Hence, from (8), an effective Reynolds number up to two orders of magnitude less than Re is a representative value for use in a CFD model comparison.

The reference length is defined from a characteristic length scale of the problem. Once selected, a corresponding reference velocity can be expressed as

$$U_r = \frac{\nu_r}{L_r} \quad (9)$$

For natural and mixed convection non-isothermal problems, Reynolds number based upon this reference velocity is not necessarily a characteristic flow number. Equating the buoyancy induced kinetic energy to the work done by the buoyancy force results in a velocity scale for natural convection (within a factor of 2) of²

$$U_r = \sqrt{g \beta \Theta L_r} \quad (10)$$

The valid Reynolds number, using this new reference velocity, now becomes

²C.D.Upson, P.M.Gresho, and R.L.Lee, *Finite Element Simulation of Thermally Induced Convection in an Enclosed Cavity*, Lawrence Livermore Laboratory, UCID-18602, March, 1980.

$$Re = \sqrt{Gr} \quad (11)$$

The Gr/Re^2 multiplier on the buoyancy term in the momentum equation (2) is also termed the Archimedes number Ar , with definition

$$Ar = \frac{Gr}{Re^2} \quad (12)$$

which becomes unity using definitions (10) and (11). This may be a preferable reference scaling for such non-isothermal problems, because large values of Ar result from using definition (9) which can cause numerical stability problems for a CFD code.

Continuity Enforcement

The equations (2)-(4) form a set of partial differential equations of elliptic character for finite Reynolds number. Equation (1) functions as a constraint on the velocity field that are solutions to the momentum equation (2). This requirement is the most fundamental issue in computing incompressible flows. Numerical methods are identified chiefly by how the constraint of continuity is handled. Procedures for satisfying the continuity constraint fall into two categories. The first are methods that approximate satisfaction of continuity, and the second enforces continuity exactly.

Inexact Continuity Satisfaction Methods

Penalty Method

The penalty method is based on extremizing a momentum equation weak statement subject to an added differential constraint of (1), the continuity equation. Pressure thus becomes replaced by an iterative expression of a multiplier times the error in the continuity satisfaction³

³T.R. Hughes, W.K. Liu, A. Brooks, *Review of Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation*, J. Comp. Physics, vo. 30, no. 1, pp. 1-60, 1979.

$$P^* = -\lambda(\nabla \cdot \mathbf{u}^*) \quad (13)$$

The "Lagrange" multiplier λ must be a large number such that the continuity error is at a small enough level. The resulting "pressure" field P^* is generally a poor approximation to the actual pressure, but the action of the constraint can be quite satisfactory.

Continuity Constraint Method

Another inexact continuity method is the continuity constraint method⁴. The primitive velocity variable is retained with a pseudo-potential function ϕ which is a measure of the error in continuity. This error in the continuity satisfaction is the difference between the exact velocity $\mathbf{u}_{ex}$ and any computed velocity $\mathbf{u}$, e.g.,

$$\nabla \phi = \mathbf{u}_{ex} - \mathbf{u} \quad (14)$$

The curl of a gradient is zero by definition, therefore taking the divergence of (14) results in a Poisson equation,

$$L(\phi) = \nabla^2 \phi - \nabla \cdot \mathbf{u} \quad (15)$$

hence ϕ is a measure of the divergence error distribution of the approximate velocity. A homogeneous Neumann boundary condition for (15) results from (14), and an approximation to pressure may be iteratively updated using

$$P^* = P_{old} + \frac{1}{\Delta t} \sum_p \phi^p \quad (16)$$

where p is an iteration counter.

⁴W.P. Noronha, P.T. Williams, A.J. Baker, *A Time Accurate Unsteady Incompressible Navier-Stokes Algorithm*, AIAA Paper AIAA-90-0238, 1990.

Exact Continuity Satisfaction Methods

Streamfunction-Vorticity Method

This alternate formulation of the governing equations enforces exact satisfaction of continuity by a vector theory transformation into streamfunction and vorticity variables. The continuity equation disappears and is replaced by a kinematic compatibility equation. Two new variables are introduced, streamfunction and vorticity, that replace velocity in two dimensions. Though mathematically ensuring precise mass conservation, this method is practically limited to two dimensions due to the complications of implementing these vector definitions in three dimensions. Pressure and velocity distributions are obtainable from the streamfunction-vorticity solution in a postprocessing operation.

From fluid mechanics [4,5], vorticity is defined as the curl of velocity. In two dimensions, velocity consists of two components and the single non-vanishing scalar component of vorticity is

$$\omega = \nabla \times \mathbf{u} \cdot \mathbf{k} \quad (17)$$

Streamfunction is defined such that its curl is the velocity vector, or

$$\mathbf{u} = \nabla \times \psi \mathbf{k} \quad (18)$$

Both vorticity and streamfunction are vectors in the $\mathbf{k}$ direction due to the two-dimensional nature of the velocity field.

With the basic definitions in place, the conservation equations can be addressed. Taking the $\mathbf{k}$ -component curl of the momentum equation (2) and assuming Re^{eff} is constant results in

$$L(\omega) = \frac{\partial \omega}{\partial t} + (\nabla \times \psi \mathbf{k} \cdot \nabla) \omega - \frac{1}{Re^{\text{eff}}} \nabla^2 \omega - Ar \nabla \times \Theta \mathbf{j} \cdot \mathbf{k} = 0 \quad (19)$$

which, by analogy to the Reynolds transport equation, is termed the vorticity transport equation.

The non-dimensional quantities are as originally defined in (5)-(6).

The energy and concentration equations are coupled to the momentum equation via velocity. But velocity has been transformed into streamfunction via equation (18). Using this identity, the energy and concentration equations become

$$L(\theta) = \frac{\partial \theta}{\partial t} + (\nabla \times \psi \mathbf{k} \cdot \nabla) \theta - \frac{1}{\text{Re}^\alpha \text{Pr}} \nabla^2 \theta - s_\theta = 0 \quad (20)$$

$$L(C_A) = \frac{\partial C_A}{\partial t} + (\nabla \times \psi \mathbf{k} \cdot \nabla) C_A - \frac{1}{\text{Re}^\alpha \text{Sc}} \nabla^2 C_A = 0 \quad (21)$$

An additional relation can be formed between streamfunction and vorticity. Substituting the streamfunction definition (18), into (17), the vorticity definition, gives the streamfunction compatibility equation

$$L(\psi) = \nabla^2 \psi + \omega = 0 \quad (22)$$

Thus, the three kinematic variables, two components of velocity and pressure, have been eliminated. They are obtainable from the two computational variables, streamfunction and vorticity, whenever desired in a post-processing operation. Velocity is obtainable from operations either (17) or (18). A Poisson equation usually provides the pressure via

$$L(p) = \nabla \cdot L(\mathbf{u}) = \nabla^2 p + f(\mathbf{u}, \theta) = 0 \quad (23)$$

The tractable equation system for the streamfunction-vorticity transformation is now complete.

Its boundary conditions are covered in the next chapter.

CHAPTER 3

BOUNDARY CONDITIONS

The incompressible Navier-Stokes equations describe an initial value, elliptic boundary value problem that, for well-posedness, requires specification of the dependent variables on all the boundaries, and an initial condition throughout the computational domain. Both primitive and transformed variable equation sets have this requirement. Boundary and initial conditions are specific to each equation set, with their own characteristic concerns.

Two types of boundary condition specifications are used. One is a "Dirichlet" condition in which the actual value is given, and the second is "Neumann" and/or "Robin", wherein the gradient normal to the boundary is specified in terms of various data. The postprocessing solutions for velocity and pressure also require full boundary data, as does the potential function ϕ in the continuity constraint formulation.

Primitive Variables

Figure 1 illustrates a typical solution domain with inlet, outlet, and walls. At an inlet, velocity or pressure must be specified as Dirichlet. The values to be fixed at the inlet boundary depend upon the known input inflow distributions for all variables. In truth, nothing except perhaps a pressure level is known at a flow outlet. Hence, it is usual to assume a vanishing Neumann condition, which means the derivative normal to the outlet approximates zero. This is the least constraining boundary condition possible for an outflow, since no prior knowledge is assumed, and the outflow geometry must adequately admit this approximation.

Wall conditions are determined by the boundary layer phenomenon of zero wall velocity, the no-slip condition, or by a viscous drag approximation. An inviscid flow simulation ($Re^{ef} \rightarrow \infty$) would employ a flow tangency condition. Along all walls, the Neumann constraint is required

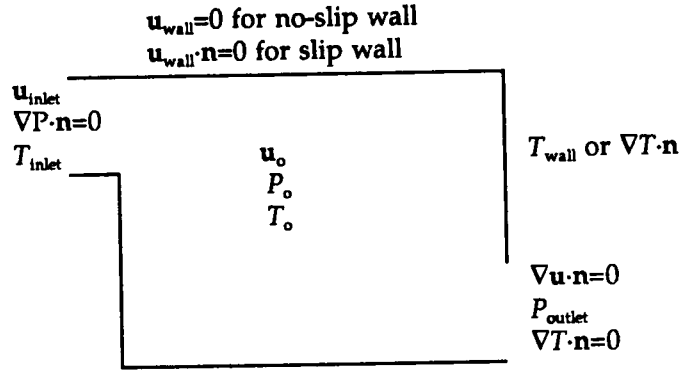


Figure 1. Primitive Variables Boundary Conditions.

for ϕ . For temperature at walls, specifications may be fixed as Dirichlet, Neumann as a heat flux, adiabatic, or with convective heat transfer. For isothermal flows, temperature definitions are unnecessary.

Streamfunction-Vorticity

The boundary and initial conditions for this transformed equation set, shown in Figure 2, are derived from the primitive variable set. Notable is the vorticity condition at the wall,

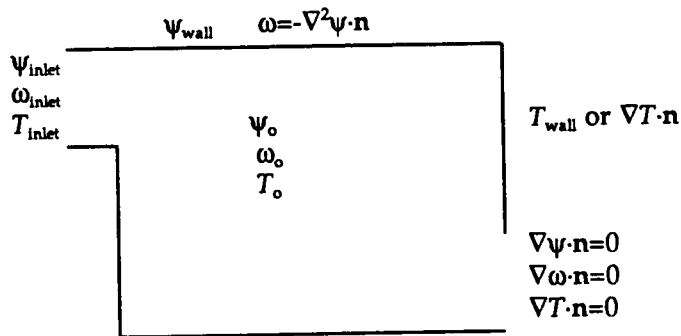


Figure 2. Transformed Variables Boundary Conditions.

defined in terms of the second normal derivative of streamfunction. Thus, vorticity at the wall is a kinematic constraint, since it varies solely with the other kinematic variable. Consider the streamfunction ψ at the wall. A nonporous wall has a constant value of ψ , thus in the tangential direction s

$$\frac{\partial \psi}{\partial s} = \frac{\partial^2 \psi}{\partial s^2} = 0 \quad (24)$$

Inserting this into (17) results in vorticity at the wall defined strictly in terms of the second normal derivative of streamfunction

$$\omega = -\frac{d^2 \psi}{dn^2} \Big|_s \quad (25)$$

A finite difference expression for (25) can be easily constructed. A second order accurate approximation [3] gives

$$\omega_w = -3 \left(\frac{\psi_{w+1} - \psi_w}{(\Delta n)^2} \right) - \frac{1}{2} \omega_{w+1} - \frac{1}{2} \frac{u_s}{\Delta n} + O(\Delta n^2) \quad (26)$$

in terms of the variables on the wall and displaced from the wall a distance Δn , signified by subscript w and $w+1$ respectively, and the velocity component tangent to the wall u_s . The vorticity wall condition (26) requires every wall node to be assigned an off-wall node. These nodes are paired by the logic within PREPROS according to the boundary condition input stream defined by the user. The correct enforcement of the vorticity wall condition is critical to the streamfunction-vorticity formulation. Improper handling will lead to instability in the solution process.

Continuing with the remaining boundary conditions, inlets require Dirichlet data for vorticity, streamfunction, and temperature. At outlets, vanishing normal derivatives are usually defined since the actual distributions are unknown. Streamfunction is naturally specified Dirichlet over walls. Temperature is defined identically to the primitive variables description above.

CHAPTER 4

CFD SOLUTION ALGORITHM

The CFD solution algorithm often involves recasting the conservation law system (CLS) into a weak statement form that allows the exact enforcement of data at the boundaries. The continuous nature of the PDE CLS weak statement is then spatially discretized (semi-discretized). A time integration algorithm then transforms the weak statement into an algebraic system solvable by numerical algebra procedures.

The state variable array for the CLS of primitive variables (1)-(4) and transformed variable streamfunction-vorticity system (19)-(21) is

$$\begin{aligned} q_1 &= \{u, v, \theta; \phi, p\} \\ q_2 &= \{\omega, \theta; \psi, u, v, p\} \end{aligned} \quad (27)$$

respectively. The variables preceding the semi-colon in either CLS satisfy the initial value PDE

$$L(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x_j} (f_j - f_j^*) - s = 0 \quad (28)$$

where f_j is the kinematic flux vector, f_j^* is the dissipative flux vector, and s is a source term. The variables following the semi-colon in (27) each obey the quasi-linear Poisson PDE

$$L(q) = \nabla^2 q - s(q) = 0 \quad (29)$$

A numerical solution algorithm for either equation type results directly from a weak statement employing an arbitrary weight function $w(x)$ integrated over the domain Ω , i.e.,

$$\int_{\Omega} w(x) L(q) d\tau = 0, \text{ for all } w(x) \quad (30)$$

The approximate solution for any variable q is expressed as an N function expansion on members of the trial space $\psi_i(x)$

$$q^N(x, t) = \sum_{i=1}^N \psi_i(x) Q_i(t) \quad (31)$$

Similarly, any weight function w used in (30) can be interpolated as

$$w^N(x) = \sum_{i=1}^N \phi_i(x) W_i \quad (32)$$

where $\phi_i(x)$ are members of the "test space," that may or may not be identical to the trial space.

The dominant error source from a spatial semi-discretization of (30) is phase dispersion. A Taylor series in time may be used to construct a modified CLS form for (28) identifying control mechanisms for this error⁵. Including this dispersive error correction in (28) gives the modified conservation system,

$$L^{Dw}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x_j} (f_j - f_j^v) - s - \frac{h}{|u|} \frac{\partial}{\partial x_j} \left(\alpha A_j \frac{\partial q}{\partial t} + \beta A_j A_k \frac{\partial q}{\partial x_k} \right) + \dots = 0 \quad (33)$$

where $A_j = \partial f_j / \partial q$ is the jacobian matrix of the kinematic flux vector, $|u|$ is the velocity magnitude, h is a length scale, and α and β are coefficients available for optimizing stability in the temporal or spatial dimensions, respectively. This modified CLS yields the "Taylor weak statement" algorithm replacing (30) as

$$\int_{\Omega} w^N(x) L^m(q^N) d\tau = 0, \text{ for all } w^N \quad (34)$$

Taking the derivative of (34) with respect to the weight function interpolation expansion coefficients W_i gives the stationary point, which is the computable weak statement

⁵A.J. Baker and J.W. Kim, *A Taylor Weak Statement for Hyperbolic Conservation Laws*, Int. Jour. Num. Mtd. Fluids, 7, 489-520.

$$TWS^N = \int_{\Omega} \phi_i(x) L^m(q^N) d\tau = 0, \text{ for all } N \quad (35)$$

Time remains as a continuous variable, making (35) an ordinary differential equation system of the form

$$TWS^N = [M] \frac{d\{Q\}}{dt} + \{R\} = \{0\} \quad (36)$$

where $[M]$ is the $N \times N$ square matrix containing all terms multiplied onto the time derivative, and $\{R\}$ is a column matrix containing all other terms. Similarly, the form of (35), for the Poisson equation (29), is

$$TWS^N = [D]\{Q\} + \{S\} = \{0\} \quad (37)$$

where $[D]$ is an $N \times N$ matrix of the discretized laplacian.

Equation (36) produces a set of ordinary differential equations, requiring a time integration procedure to produce an algebraic problem. The one step variable implicit Euler method uses time steps n and $n+1$ in the form

$$\{Q\}^{n+1} = \{Q\}^n + \Delta t \left(\theta \frac{d\{Q\}^{n+1}}{dt} + (1-\theta) \frac{d\{Q\}^n}{dt} \right) \quad (38)$$

where θ is the implicitness factor valued between $1/2$ and one. Time-accurate simulations require $\theta=1/2$, while $\theta=1$ can lead to the fastest attainment of a steady-state solution.

Combining (38) with the ODE system (36) yields

$$\{F(Q)\} = [M]\{Q^{n+1} - Q^n\} + \Delta t (\theta \{R\}^{n+1} + (1-\theta)\{R\}^n) = \{0\} \quad (39)$$

There is no corresponding need to evaluate time derivatives for (37) since it is already in an algebraic form. Having expressed the algorithm in a fully algebraic form, the evolution of the solution is computed via the Newton iteration

$$\frac{\partial(F(Q))}{\partial(Q)^{n+1}} (Q_{p+1} - Q_p) = [J]_p \{\delta Q\}_{p+1} = -[F]_p \quad (40)$$

where $[J] = \partial(F)/\partial(Q)$ is the Newton jacobian square matrix, and p is the iteration counter within each time step indexed by n . The incremental change in the variable is calculated at each iteration, and the total change per time step is

$$\{\Delta Q\}^{n+1} = \sum_p \{\delta Q\}_{p+1} \quad (41)$$

The updated solution at time t^{n+1} is

$$\{Q\}^{n+1} = \{Q\}^n + \{\Delta Q\}^{n+1} \quad (42)$$

To this point, no specific spatial discretization has been chosen. It is the prerogative of the CFD code designer to choose the weight and trial functions. For instance, constant weight functions give the basis of a finite volume algorithm. Or, spatially varying weight functions produces a finite element technique. Additionally, defining the two functions to be identical for all N , i.e.,

$$\phi_i(x) = \psi_i(x) \quad , \quad \text{for } 1 \leq i \leq N \quad (43)$$

results in the Galerkin finite element formulation. The spatial approximation for (31) becomes,

$$q^h(x,t) = U \{N_k(\eta_p)\}^T \{Q(t)\} \quad (44)$$

using the set of k^h degree localized interpolating polynomials $\{N_k\}$. These are expressed in the local element coordinate system η_i , numbered for quadrilaterals from 1 to n -dimensions, which is 2 for this investigation. The algorithm spatial order of accuracy is nominally $2k$, and a linear ($k=1$) degree algorithm is chosen for the codes used in this study, giving 2^{nd} order spatial accuracy.

The choice of the Galerkin Taylor weak statement algorithm completes the requirement. However, an additional step is needed to obtain a versatile form. Applying the Green-Gauss theorem to (33) transfers the spatial derivative from the trial to the test functions resulting in

$$\begin{aligned}
TWS^N = & \int_{\Omega} \{\phi\} \frac{\partial q^N}{\partial \tau} d\tau - \int_{\Omega} \frac{\partial \{\phi\}}{\partial x_j} (f_j - f_j^N)^N d\tau \\
& - \int_{\Omega} \{\phi\} s^N d\tau + \int_{\partial\Omega} \{\phi\} (f_j - f_j^N)^N \hat{n}_j d\sigma \\
& \int_{\Omega} \frac{\partial \{\phi\}}{\partial x_j} \frac{h}{|u|} \frac{\partial}{\partial x_j} \left(\alpha A_j \frac{\partial q}{\partial \tau} + \beta A_j A_k \frac{\partial q}{\partial x_k} \right) d\tau = 0
\end{aligned} \tag{45}$$

The surface integral that has appeared becomes the means for boundary condition enforcement, which is a naturally occurring result of the Galerkin weak statement formulation.

Implementation of this algorithm includes assembling all data over the non-overlapping union of finite elements over the discretization $\Omega^h = \cup \Omega_e$. The resulting matrix statement is solved by the procedure (40)-(44).

CHAPTER 5

SOLUTION STRATEGIES

A CFD solution procedure should be constructed with accuracy and computational efficiency as the primary goals. To achieve this, PREPROS guides code execution via several user controlled provisions, including integration time step, continuation on physical parameters such as Re^{ef} , and remeshing for solution adaptation.

Time Stepping Procedure

Efficiency means the fastest possible attainment of the converged solution, which requires that each time step taken produces the maximum possible change while the Newton iteration remains convergent. Within each time step Δt , the solution is iterated until the biggest change in the variables is less than the convergence criteria ϵ input by the user. Then, a new time step is chosen and iteration starts again.

Each new time step is based on the iterative solution behavior of the previous step. Successful attainment requires that the maximum change in iteration variable decrease with every iteration, or

$$\delta Q_{p+1}^n \leq \delta Q_p^n \quad (46)$$

where p is the iteration counter within time step n . Upon convergence, the new time step is determined by the ratio of observed 1st iterate change, to a pre-determined admissible maximum change signified by the input variable δQ_{max} . Then

$$\Delta t^{n+1} = \frac{\Delta t^n}{TFACT}, \quad TFACT = \frac{\delta Q_{p=1}^n}{\delta Q_{max}} \quad (47)$$

subject to a minimum value of TFACT, usually about 0.1. To accelerate time stepping, if the last

1st iterate change is of the same order as the previous change, or $\delta Q_{p=1}^n \approx \delta Q_{p=1}^{n+1}$, then TFACT may be adjusted downward, increasing Δt^{n+1} .

If (46) is not satisfied, the solution is probably diverging, and a reduced time step $\Delta t'$, hence $\delta Q_{\max}'$ are chosen from the old $\delta Q_{\max}$ and $\delta Q_{p=1}^n$. The number of iterations required to achieve iteration convergence can further modify the time step. The desired optimum is 3 or less iterations per time step, therefore the time step is decreased to encourage this performance when 4 or more iterations per time step are encountered. In case $\delta Q_{p=1}$ exceeds $\delta Q_{\max}$ and convergence is attained, $\delta Q_{\max}$ is redefined as $\delta Q_{p=1}$, limited to an increase of 2.5 times it's previous value.

Continuation Parameters

Convergence of the iteration algorithm requires that the solution estimate at the next time step $n+1$ be within the convergence radius of the iteration procedure. This radius is a function of the physical parameters of the flow, including Re^{ef} and Ar , and the integration parameters of time step and maximum variable change $\delta Q_{\max}$.

To illustrate, computing a solution at large effective Reynolds numbers (Re^{ef}), i.e. 10^6 , requires time steps small enough to achieve convergence at that Re^{ef} . Alternatively, a speedier solution can usually be found at a smaller Re^{ef} that enables larger time step sizes to be used. This solution then becomes the initial condition for a larger Re^{ef} solution. This recycling, called "continuation", continues until the desired parametric state is achieved.

For situations that include thermal effects, a similar procedure is followed for Ar and Pe^{ef} . The inputs for continuation sequence are contained under the command name CONT, for "continuation", as explained in Chapter 6.

Adaptive Remeshing

After a solution has been calculated, inspection of the output may reveal oscillations, called "2Δx waves" (see Chapter 4), which indicate the inability of the grid to support the solution gradients. One might elect to use artificial dissipation to smooth these errors. The preferable

approach is a regriding, then continuing the calculation on the adjusted grid, rather than starting over with a zero initial condition. Using PREPROS, a refined grid is easily defined, and the final solution from the coarse grid is read in and interpolated as the initial condition for the fine grid solution. This procedure is outlined in Chapter 6, Initial Conditions.

CHAPTER 6

PREPROS OPERATION

The inputs required for PREPROS are derived from the theory outlined in Chapters 2-5. These include physical data, boundary and initial conditions, and parameters controlling program solution sequence. PREPROS proceeds through a series of questions that the user answers, in the process defining an actual problem to be solved. The information is analyzed, converted into input data that a CFD program can read in, and output to a file named by the user.

In the file, all data are segregated under various 4-letter command names, that, upon being read by the CFD solver, inform the program what type of data follows. These names, in the same order found in the file, are summarized here:

TITL - title given by the user

TYPE - parameters describing the type of problem

GRID - data to construct the computational grid

NODS - command to print the node map

INIT - initial condition

DIRI - Dirichlet boundary conditions

NEUM - Neumann boundary conditions

OMEG - vorticity kinematic condition nodal connectivity

WALL - pressure Poisson nodal connectivity

CONT - continuation parameters

INTE - integration parameters

The sequence of problem set-up is described below, grouped by subject with the command name that is written in the data file listed in parenthesis. Questions asked by the preprocessor are listed in COURIER type with accompanying explanation. The user must reply

according to the following conventions.

Rules for replies to PREPROS queries:

- The letter "Y" or "y" is sufficient to signify yes.
- Any other letter is no, including carriage return.
- Numbers are input free format, so a decimal point is necessary only for numbers with fractions.

Setup Parameters (TITL,TYPE,PARM)

After an initial message about the program, PREPROS asks for a file name to write the data to.

ENTER NAME OF FILE TO WRITE DATA TO:

Then input the type of output device used, and the title that identifies the problem to be set up.

SPECIFY TERMINAL TYPE 1) TEK 4115
 2) TEK 4107
 3) VT 240
 4) IMAGEN
 5) UNIX WORKSTATION

***** TITLE (TITL) *****

WHAT IS THE TITLE OF THE PROBLEM?

The CFD code algorithm that will be used to solve the problem must be chosen. PREPROS needs to know this in order to construct the proper input data as discussed in Chapters 2 and 3.

***** ALGORITHM AND PROBLEM PARAMETERS (TYPE) *****

SPECIFY COMPUTATIONAL METHOD TO USE:

- 1) STREAMFUNCTION-VORTICITY
- 2) CONTINUITY CONSTRAINT

PREPROS assumes certain finite element parameters, and that the problem is two-dimensional. These parameters are presented to the user as

ALGORITHM AND PROBLEM PARAMETERS

FINITE ELEMENT DEGREE BASIS	1	
DIMENSIONAL ORDER OF PROBLEM	2	
NUMBER OF DEGREES OF FREEDOM/NODE	2	
ELEMENT TYPE		QUADRILATERALS

Other questions describing the problem follow:

IS THIS PROBLEM ISOTHERMAL?

Answer yes for isothermal problems, or no for non-isothermal problems, which require defining input temperature data.

IS THIS AN AXISYMMETRIC PROBLEM?

If yes, pick the axis of symmetry.

THE AXIS OF SYMMETRY IS: 1) X-AXIS
 2) Y-AXIS

The next question is used for inert contaminant transport studies, and requires additional input.

DOES THIS PROBLEM HAVE TRANSPORT OF AN INERT SPECIES?

The default values used for the fluid properties are those of air at standard atmosphere. For isothermal problems, only the first three values are output since the second three concern thermal properties. Defining continuation Reynolds and temperature difference numbers (explained in CONT command name) effectively overrides these values.

***** PHYSICAL PARAMETERS (PARM) *****

FLUID PROPERTIES OF AIR (70 DEG. F, 1 ATM):

1 GRAVITATIONAL ACCELERATION....	0.3220E+02	(FT/SEC^2)
2 KINEMATIC VISCOSITY.....	0.1591E-03	(FT^2/SEC)
3 DENSITY.....	0.7559E-01	(LBM/FT^3)
4 SPECIFIC HEAT.....	0.2370E+00	(BTU/(LBM*DEG F))
5 THERMAL CONDUCTIVITY.....	0.4176E-05	(BTU/(FT*SEC*DEG F))
6 COEFF. OF THERMAL EXPANSION...	0.8182E-02	(1/DEG R)

DO YOU WANT TO CHANGE ANY OF THESE VALUES?

Reference length is the characteristic length of the problem, such as inlet jet width or wall length.

It is used to scale the non-dimensional numbers in equation (5).

REFERENCE LENGTH..... 1.0000
DO YOU WISH TO CHANGE THE REFERENCE VALUE?

Grid Construction (GRID)

As discussed in Chapter 4, numerical solution of the governing PDE system requires definition of a computational grid over the problem domain. The grid is constructed of grid lines, with node points at their intersection. These nodes are where the solution is calculated. Generally, a more accurate solution is obtained as more nodes are used. Unfortunately, the time to compute

a solution also increases with the number of nodes used.

The trade-off between computational time and solution accuracy can be ameliorated by non-uniform grid construction. A grid with nodes clustered in one area and spread out in another can reduce the error associated with the spatial discretization without overly increasing solution time. Nodes should be clustered in the direction of the greatest solution gradients, generally where the velocity magnitude changes the most in a small distance. Such circumstances are found at high velocity inlets and along no-slip walls because of boundary layer effects. Therefore the node distribution should be denser at those areas, and progressively coarser towards the domain interior.

The type of geometry used to construct the grid can be one of three choices of increasing generality and complexity. All permit the clustering of nodes and node sparseness along user designated planes. Purely rectangular problems require the minimum information, quadrilaterals need more, and curved-sided quads require the most input. The inputs needed for each is summarized in Table 1.

Table 1. Grid Geometry Choices.

Quadrilateral Geometry Type	Progression Ratio	No. of Elements	Macro Length	Macro Vertices	Macro MidSides
Rectangular	yes	yes	yes	no	no
Non-Rectangular	yes	yes	no	yes	no
Curved-Sided	yes	yes	no	yes	yes

***** GRID CONSTRUCTION (GRID) *****

DEFINE PROBLEM GEOMETRY: 0 - RECTANGULAR QUADRILATERAL

Results in a grid that contains rectangular elements, with multiple progression ratios definable in each direction.

1 - GENERAL QUADRILATERAL

Enables quadrilaterals of nonrectangular character to be constructed, requiring vertex coordinates of the domain to be input.

2 - CURVED-SIDED QUADRILATERAL

Additionally requires the mid-side coordinates of the domain for quadrilaterals with circular arcs as sides.

Two types of information are needed to construct the grid: element, and macro domain information. The element information consists of the number of elements, and their spacing, in each direction η_1 and η_2 , which are the local coordinates of the grid with the origin in the lower left corner. The η_1 axis is in the horizontal direction, and η_2 is in the vertical direction. Positive is to the right and up.

The element length $\Delta\eta_e$ for M elements in each dimension is

$$\Delta\eta_e = PR \Delta\eta_{e-1} \quad 1 \leq e \leq M \quad (48)$$

so that for $PR > 1$, the elements grow larger along the positive direction of the axis, and for $PR < 1$, they get smaller. Uniform spacing results for $PR=1$.

Further information is needed because grids may be constructed in separate sections called *macro domains*. Individual coordinates, progression ratios, and numbers of elements can be input for each.

For rectangular domains, the η_i , and x and y coordinates coincide, and therefore only the x and y values are needed for each direction to define the macro dimensions. As shown in Figure 3, the initial value of x_i and y_i of the first macro is input, then the final value x_f or y_f of each successive macro.

For the general and curved-sided quadrilaterals, the macro domain definition is demonstrated in Figure 4. The x and y coordinates of each macro vertex, signified by V_i at the solid circles, must be input. They are numbered starting at the lower left corner, and progress in the η_1 direction, then along each η_1 row successively in the η_2 direction. Mid-side coordinates,

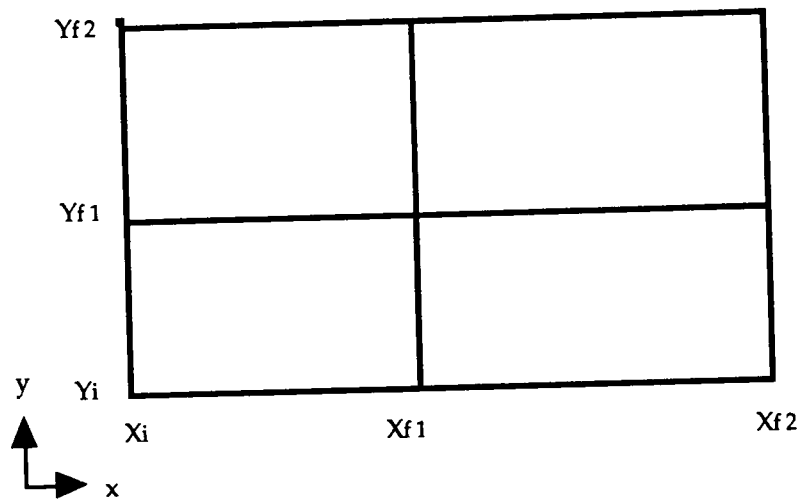


Figure 3. Rectangular Macro Coordinates.

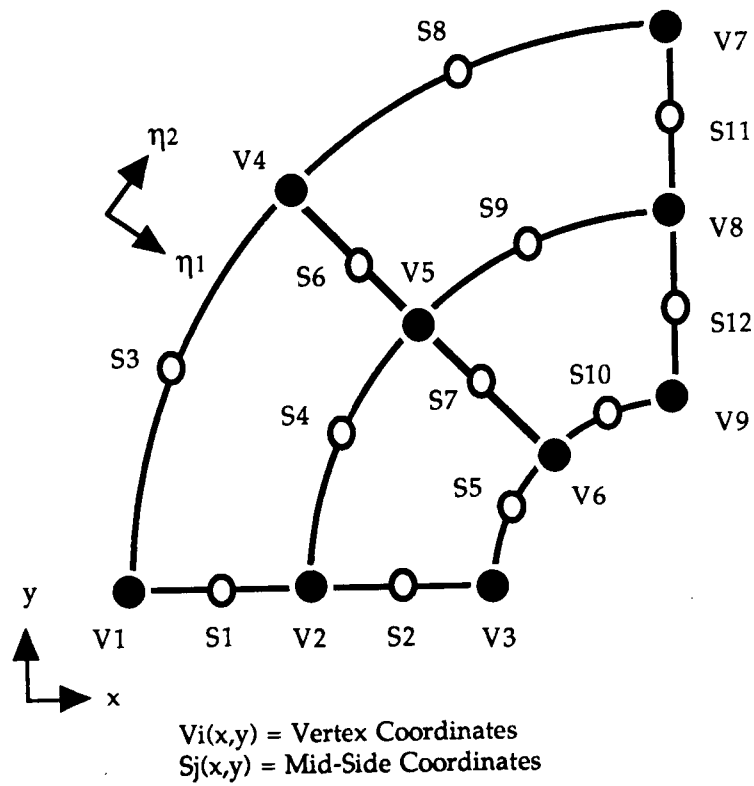


Figure 4. General Quadrilateral Macro Coordinates.

needed for curved-sided quads only, progress similarly, which are represented by S_j at the empty circles.

A previously generated grid can be used for the present problem definition, or a new grid constructed. In order to manipulate an old grid, it must be of the same geometry type designated above. An old grid cannot be changed to another type of geometry; a new one must be defined.

DO YOU WANT TO 1) BEGIN WITH AN OLD GRID
 2) DEFINE A NEW GRID

For a reply of "1", the name of an existing file containing the grid data must be supplied.

ENTER NAME OF FILE TO READ IN:

To construct a new grid, answer "2" and questions will be asked to provide the information for the new grid. The procedure to define a new grid is given first for rectangular quadrilaterals.

ENTER NUMBER OF MACRO DOMAINS IN X DIRECTION:

Enter the number of macro domains in the horizontal direction.

ENTER INITIAL COORDINATE (FT) :

The initial horizontal coordinate of the first macro.

ENTER FINAL COORDINATE (FT) :

The final horizontal coordinate of the first macro.

ENTER PROGRESSION RATIO:

Progression ratio for the first macro.

If more than one macro domain exists in one direction, the questions asking for the final coordinate and progression ratio will repeat for each macro domain. After the first macro is defined, the initial coordinate for the next macro is the final coordinate from the previous macro.

The input of non-rectangular quads is different because the macro coordinates are input separately for each corner. Instead of the initial and final coordinates of each macro, the x and y coordinate of each vertex are needed, and additionally for curved-sided quads, the mid-side coordinates. First, as for rectangular quads, the number of macros, progression ratio, and number of elements are input.

ENTER NUMBER OF MACRO DOMAINS IN ETA1 DIRECTION:

Enter the number of macro domains in the horizontal (eta1) direction.

ENTER PROGRESSION RATIO FOR MACROREGION 1:

The progression ratio of the first macro in the eta1 direction.

ENTER NUMBER OF ELEMENTS IN MACROREGION 1:

The number of elements in the first macro in the eta1 direction. For more than one macro in this direction, the questions repeat for progression ratio and elements numbers.

ENTER NUMBER OF MACRO DOMAINS IN ETA2 DIRECTION:

Enter the number of macro domains in the vertical (eta2) direction.

ENTER PROGRESSION RATIO FOR MACROREGION 1:

The progression ratio of the first macro in the eta2 direction.

ENTER NUMBER OF ELEMENTS IN MACROREGION 1:

The number of elements in the first macro in the eta2 direction.

MACRO VERTEX NUMBERING BEGINS IN THE LOWER LEFT CORNER AND CONTINUES IN THE ETA1 DIRECTION FIRST, THEN ALONG ETA2. FOR THE MACRO PARTITIONS INPUT, THE NUMBERING SCHEME IS

For example, 2 macros defined in the x direction and 1 in the y direction would result in

```
4----- 5----- 6
|         |         |
1----- 2----- 3
```

FOR EACH MACRO VERTEX, ENTER X AND Y COORDINATES (FT) ON THE SAME LINE SEPARATED BY A SPACE OR COMMA.

FOR MACRO VERTEX 1, ENTER X AND Y COORDINATES (FT).

FOR MACRO VERTEX N, ENTER X AND Y COORDINATES (FT).

If you want to change the grid, answering no to the following enables checking of all grid information just input:

DO YOU WANT TO CONTINUE WITH THIS GRID?

Once the grid is completed, PREPROS determines whether the nodes are numbered in the horizontal (eta1) or vertical (eta2) direction. Whichever direction has the least number of nodes is the selected ordering. If the number of nodes in each direction are equal, then the eta1 direction

is chosen. A message is output to remind the user of the direction of node numbering, since nodes are used to specify boundary conditions. The direction of numbering must be known to correctly specify the nodes that correspond to boundary conditions.

The nodes that correspond to the vertices of the macros are also output for the user's reference. Optionally, they can also be output into a file called "vertex.map" if the user so wishes. When using a workstation with multiple windows, this file can be accessed when defining boundary conditions and kept open on the screen. As an example, for 2 macros in the x direction, each with 5 elements, and one macro of 10 elements in the y direction,

THE VERTICES OF THE MACRO DOMAINS ARE

111	115	121
1	6	11

WOULD YOU LIKE A FILE CONTAINING THE NODE MAP OF THE MACRO VERTICES?

If so,

THEY ARE CONTAINED IN A FILE CALLED VERTEX.MAP, WHICH CAN BE ACCESSED NOW.

Initial Condition (INIT)

A previous solution can be used as the initial condition. Or, the solution can be started with all variables specified zero (except for boundary conditions), called a "cold start". Computational effort is saved by starting with a previous solution, which is not as unconverged as a cold start.

***** INITIAL CONDITION (INIT) *****

INITIAL CONDITION IS: 1) ZERO (COLD START)

No initial input - all values are zero.

2) UNIFORM VALUE

One value for each variable is input across the domain.

3) READ FROM DATA FILE

A data file containing a previous solution is read in. It is specified by

ENTER DATA FILE CONTAINING INITIAL CONDITION TO BE READ IN:

The effective use of this utility is to input a complete solution generated from a previous run which the CFD solver has output and stored in a file called OUTT.DAT. Specify this file, and PREPROS inserts the solution from the file after the command name INIT, which is read by the solver and assigns the data as the initial value of the solution variables.

A previous solution based on a different grid can also be used by interpolating the data onto the current computational grid. This ability offers increased flexibility in progressing from one solution to another. For example, use a relatively coarse grid until $2\Delta x$ waves appear in the solution. Then, refine the grid, read in the coarse grid solution to be interpolated, and restart. The solution to use for restarting is the one before oscillations are present.

(This can be done without running PREPROS by editing the file, and inserting command name INIT and after it the initial condition just before the GRID command name. Don't forget to change any parameters that will continue the desired evolution of the calculation, such as Re^{ef} .)

As another means to accelerate convergence of a thermal cold start, the reference temperature can be used as the initial condition of temperature. This utility can generally be used to encourage convergence of thermal problems since any non-zero temperature distribution is usually closer to the converged state.

DO YOU WANT TO SPECIFY THE REFERENCE TEMPERATURE AS THE UNIFORM INITIAL TEMPERATURE CONDITION?

Boundary Conditions (DIRI, NEUM, OMEG)

Boundary conditions are assembled from walls, inlets, outlets, and symmetry planes. Wall and inlet locations are specified by the node numbers at each end, so the nodes corresponding to the location of these physical features must be known. Each must lie along a node row, as must the two defining nodes, as seen in Figure 5. Node number calculation can be aided by a node map of the macro vertices generated during grid definition. Symmetry planes are defined by picking what sides of the domain are planes about which the problem is symmetric. Wherever no walls, inlets, or symmetry planes are defined on the domain perimeter, PREPROS assumes an outlet.

The node numbers of the walls and inlets must be input in a counterclockwise fashion. This ordering enables the preprocessor to determine their proper orientation and their corresponding boundary conditions. Each is defined by the beginning and ending node of the inlet or wall in a counterclockwise direction. Figure 6 demonstrates this scheme for an example problem. The next sections explain in detail the input procedure for inlets and walls.

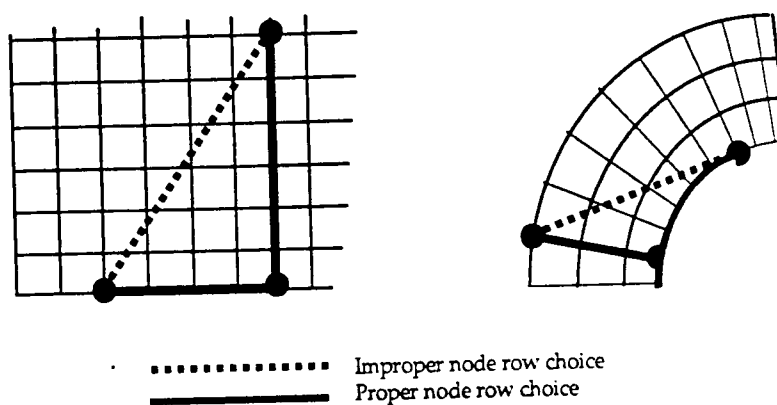


Figure 5. Example of Node Row Specification.

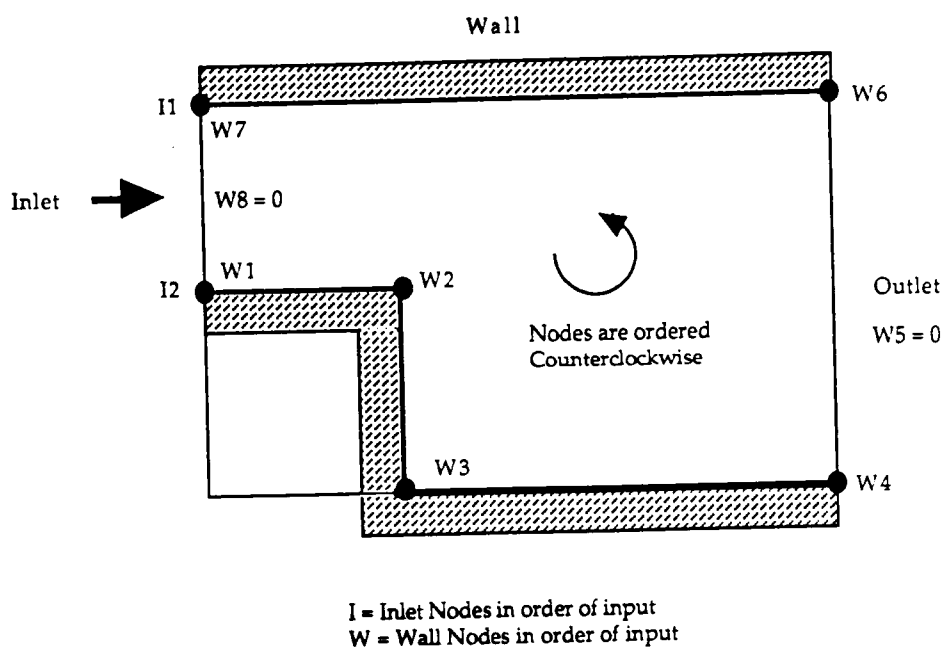


Figure 6. Example of Inlet and Wall Node Numbering

Inlets

Locations of inlets are defined by the node on each side of the inlet (initial and final) numbered counterclockwise, as shown in Figures 5 and 6. This order enables proper node succession of the inlet. For a natural convection problem, where the only force affecting the flow is thermal buoyancy, no inlets (or outlets) are present.

```
***** BOUNDARY CONDITIONS (DIRI, NUEM, ROBN) *****
```

```
***** INLETS *****
```

```
HOW MANY INLETS ARE THERE?
```

```
ENTER FIRST NODE OF INLET 1:
```

```
ENTER LAST NODE OF INLET 1:
```

Flow conditions into the domain must be defined by the user. Upon declaring an inlet, the velocity profile must be specified as slug (uniform), parabolic, half-parabolic, or defined by each individual node. Either full or half parabolic requires the input of maximum velocity, and the node at which the maximum occurs.

```
CHOOSE AN INLET VELOCITY DISTRIBUTION:
```

- 1 - UNIFORM
- 2 - NON-UNIFORM
- 3 - PARABOLIC
- 4 - HALF-PARABOLIC

For options 3 and 4, input maximum velocity. Option 1 uses this velocity uniformly across the inlet. Input velocity node by node for option 2.

```
ENTER MAXIMUM FLUID VELOCITY AT INLET:
```

For non-uniform profiles, input velocity for all "N" nodes.

```
ENTER VELOCITY FOR NODE 1:
```

↓

```
ENTER VELOCITY FOR NODE N:
```

The nodes and inflow velocity are printed at the inlet for each node, and PREPROS asks if they are correct.

```
INLET 1      NODE      INFLOW VELOCITY
```

IS THIS DATA CORRECT?

If so, PREPROS continues on to the next inlet, or on to wall definitions. If it is not correct, the preprocessor loops back and queries the inlet data again.

Walls

Any segment of the domain perimeter can be a wall. If an inlet or outlet bisects a wall, then that side contains two walls which must be defined separately. Any side that is entirely an inlet or an outlet contains no walls. Walls can be given a non-zero velocity in the tangential direction. A velocity must be defined at each wall for an affirmative answer to the following question:

***** WALLS *****

DO ANY WALL SEGMENTS HAVE A NON-ZERO TANGENTIAL VELOCITY?

The numbering of walls is counterclockwise, starting at any wall. See Figure 6 for an example. Input the initial node of the wall, and then the final. If there is a wall that connects to this node, simply input the final node of the next contiguous wall. Continue counterclockwise until a break in the form of an inlet, outlet, or symmetry plane is encountered. In this case, input a 0, and resume with the initial node of the next wall. When a circuit around the domain is complete and all features are entered, input a 0 for the next final node, then a 0 for the next initial node, and PREPROS will advance to the next input section of symmetry planes.

ENTER WALL SEGMENT 1 INITIAL NODE:

ENTER WALL SEGMENT 1 FINAL NODE:

For non-isothermal flows, the wall energy condition can either be a fixed temperature, or adiabatic (zero heat flux). Temperature can be defined as a constant over the wall, or defined by individual nodes.

CHOOSE THE TEMPERATURE CONDITION:

- 1 - ADIABATIC
- 2 - FIXED TEMPERATURE
- 3 - FIXED HEAT FLUX

For option 2,

IS THE TEMPERATURE UNIFORM?

If so,

ENTER TEMPERATURE OF WALL SEGMENT (DEG F):

or if non-uniform wall temperature distribution, for N nodes,

ENTER TEMPERATURE OF WALL NODE 1:

↓

ENTER TEMPERATURE OF WALL NODE N:

Option 3 requires the input of heat flux.

ENTER HEAT FLUX (BTU/(H FT²)):

If tangentially moving walls were signified,

ENTER TANGENTIAL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

The nodes for this wall segment are printed, and the opportunity is given to change the
wall.

WALL SEGMENT 1	NODE	TEMPERATURE
----------------	------	-------------

WALL VELOCITY=

IS THIS DATA CORRECT?

If not, the queries for that side repeat.

To save computational effort, a symmetry plane can be assigned at one of the four walls.

If indicated, the proper boundary conditions are assigned at that boundary.

DO YOU WANT TO DEFINE A BOUNDARY AS A SYMMETRY PLANE?

If so,

ENTER WHICH WALL :	1 - TOP
	2 - BOTTOM
	3 - RIGHT
	4 - LEFT

A reference value can be specified for streamfunction and pressure since it is the
difference in these variables that is of interest.

DO YOU WANT A NODE TO BE SET AT A REFERENCE VALUE OF STREAMFUNCTION?

If so, input the node and it's value.

WHAT NODE?

ENTER REFERENCE VALUE:

At this point, all the boundary conditions are specified, and can be plotted on the screen according to which graphics device picked at the start of the program.

DO YOU WISH TO SEE A PLOT
OF THE SPECIFIED BOUNDARIES?

If yes,

THE FOLLOWING PLOT SHOWS THE SELECTED GRID AND THE LOCATION OF
THE INLETS, OUTLETS, AND WALLS.

PRESS [RETURN] TO CONTINUE

PLEASE WAIT.....

After viewing the plot, press [return] again to continue.

Continuation Parameters (CONT)

Continuation parameters enable calculation of several interim solutions in one execution of the solver (see Chapter 5). These solutions are steps along the path to the final desired condition. Each solution is a physical state defined by the continuation parameters. The parameters are the non-dimensional numbers that describe the flow conditions (see Chapter 2). They are effective Reynolds number Re^{ef} and continuation temperature difference ΔT^c used for the energy equation. Temperature is used in determining the effective Grashof or Rayleigh number, and therefore the bouyancy multiplier, which for mixed convection is Gr/Re^2 or Ar , and for free convection is $RaPr$.

For isothermal (and to a lesser degree, non-isothermal) situations, computational difficulty increases as Reynolds number increases. To successfully achieve high Re solutions requires starting at a low value and increasing in a series of steps. A suggested procedure is to start at $Re^{ef} = 10$, and increase to 100, 500, 1000, and then 2000. From there, increase by less than a multiple of 2. The specific problem ultimately determines the speed of Re^{ef} stepping. If convergence is not reached at some step, the target Re^{ef} should be reduced until successful convergence. The same scheme should be used to step ΔT^c from 10 to the maximum value desired.

***** CONTINUATION SEQUENCE (CONT) *****

For natural convection problems, Reynolds number has no meaning apart from the thermal parameters, and it therefore is not input. Re is calculated from the square root of Grashof number knowing ΔT . The Reynolds number for mixed convection is still input as Re^{ef} .

GR/Re^2 (ARCHIMEDES NO.) SHOULD BE NEAR 1 FOR OPTIMAL SOLUTION STABILITY.

DO YOU WANT TO CHANGE THIS?

ENTER NUMBER OF EFFECTIVE REYNOLDS NUMBERS. FOR NATURAL CONVECTION, INPUT ZERO. Re IS CALCULATED FROM THE SQUARE ROOT OF GR .

For a non-zero reply,

ENTER EFFECTIVE REYNOLDS NUMBERS, ONE PER LINE

The numbers just entered are printed out for inspection.

EFFECTIVE REYNOLDS NUMBERS:

If incorrect, they can be changed.

DO YOU WANT TO CHANGE THESE REYNOLDS NUMBERS?

Queries continue until all are entered.

Additionally, artificial dissipation (see Chapter 4) can be input for each continuation step, satisfying the need for diffusive error control as Re^{ef} increases.

DO YOU WANT TO USE ARTIFICIAL DISSIPATION?

If so, enter one per line,

ENTER ART. DISSIPATION COEFFICIENTS, ONE PER LINE

They are printed out for verification.

ARTIFICIAL DISSIPATION COEFFICIENTS:

DO YOU WANT TO CHANGE THESE DISSIPATION COEFFICIENTS?

Peclet number can also be input if explicit control of the energy equation is necessary (appropriate only when starting a thermal calculation from an isothermal initial condition).

DO YOU WANT PECKET NUMBER TO BE CALCULATED FROM EFFECTIVE REYNOLDS NUMBER?

If not, input the effective Peclet numbers desired.

ENTER NUMBER OF EFFECTIVE PECKET NUMBERS

Enter each one per line.

The numbers just entered are printed out for inspection.

If incorrect, signify to change them.

Queries continue until all are entered.

Enter each one per line.

The numbers just entered are printed out for inspection.

If incorrect, signify yes.

Queries continue until all are entered.

The result of these inputs is a stack of continuation numbers to be calculated. Isothermal problems require Re^{ef} , and non-isothermal problems ΔT^c .

When using the streamfunction-vorticity method, velocity can be post-processed either from vorticity (17) or streamfunction (18), which necessitate different boundary conditions. Using vorticity results in velocity being fixed as Dirichlet on all walls and inlets, whereas for the streamfunction operation, velocity is not fixed at these boundaries but rather is calculated directly from streamfunction. Specify which by

The partial differential equation system for the problem defined is integrated in time until completion is indicated by satisfying one of several ending conditions. For indication of the implicitness factor, a steady-state or unsteady calculation must be picked.

***** INTEGRATION PARAMETERS (INTE) *****

CHOOSE TYPE OF SOLUTION: 1) APPROACH TO STEADY-STATE
 2) TIME ACCURATE UNSTEADY

Several other limiting factors must be specified. Default values are available, and are used unless changed. The salient ones are the maximum number of iterations, maximum change in variables per step, and maximum time step size.

INTEGRATION PARAMETERS:

1	DTZERO	=	INITIAL INTEGRATION SIZE.....	0.005
2	TSTOP	=	TIME STATION TO STOP INTEGRATION.....	100.0
3	DTMAX	=	MAXIMUM INTEGRATION STEP SIZE.....	5.0
5	THETA	=	INTEGRATION IMPLICITNESS.....	0.5
6	NITMAX	=	MAX. NO. OF ITERATIONS.....	100
7	NPRINT	=	ITERATION INTERVAL FOR PRINTING SOLUTION....	100
8	EPS	=	ITERATION CONVERGENCE REQUIREMENT.....	0.005
9	DQMAX	=	MAXIMUM VARIABLE CHANGE PER STEP.....	0.1

DO YOU WISH TO CHANGE ANY OF THESE PARAMETERS?

THE DATA FILE IS NOW READY FOR EXECUTION.

Upon program completion, PREPROS writes all data into the file with the name specified, and is now ready to be used with CFD code OMEGSI or PHI. Either program by default needs an input file called INPUT.DAT, so the output file from PREPROS must be renamed. Upon solution completion, the results should be examined using the output file, called OUTPUT.DAT, which lists the computational history and solution in tabular form to determine if the desired state was achieved. Also, the solution variables can be plotted from the PLOTT.DAT file output by the CFD code.

If the solution has not arrived at the condition wanted, the file containing the solution can be read in as the initial condition for a new computation, as directed in Initial Conditions on page 29.

CHAPTER 7

EXAMPLE PROBLEMS

To aid in the proper use of PREPROS, four examples of increasing complexity and usefulness are presented. The program questions and replies follow, with figures of the plots that appear on the screen. The first is a CFD benchmark of a driven cavity, so-called because one wall of a square cavity is given a non-zero tangential velocity. This is a crucial test of an algorithm to support singularities at the corners of the moving wall. A uniform grid is illustrated, then a non-uniform one. Another benchmark is a thermal cavity, whose opposing vertical walls are at hot and cold temperatures. Artificial viscosity is then demonstrated with the thermal cavity. The last example is a kitchen with an exhaust hood over a hot stove surface.

Driven Cavity

```
$ RUN PREPROS
```

```
THIS PROGRAM ALLOWS THE USER TO INTERACTIVELY CREATE AN INPUT DATA FILE  
FOR EXECUTION OF A 2-DIMENSIONAL FLUID/THERMAL CFD CODE.
```

```
- POSITIVE REPLIES TO QUESTIONS REQUIRE AN UPPER-CASE Y, NEGATIVE  
REPLIES REQUIRE ONLY A RETURN.
```

```
NAME OF FILE TO WRITE DATA TO ?  
DRIVEN.DAT
```

```
SPECIFY TERMINAL TYPE      1) TEK 4115  
                           2) TEK 4107  
                           3) VT 240  
                           4) IMAGEN  
                           5) UNIX WORKSTATION
```

```
3
```

```
***** TITLE (TITL) *****
```

```
WHAT IS THE TITLE OF THE PROBLEM?  
16X16 DRIVEN CAVITY
```

```
***** ALGORITHM AND PROBLEM PARAMETERS (TYPE) *****
```

```
SPECIFY COMPUTATIONAL METHOD TO USE:
```

- 1) STREAMFUNCTION-VORTICITY
- 2) CONTINUITY CONSTRAINT

```
1
```

```
FINITE ELEMENT BASIS DEGREE:      1  
DIMENSIONAL ORDER OF PROBLEM:    2
```

NUMBER OF DEGREES OF FREEDOM / NODE: 2
ELEMENT TYPE: QUADRILATERALS

IS THIS PROBLEM ISOTHERMAL?
Y

DOES THIS PROBLEM HAVE TRANSPORT OF AN INERT SPECIES?

IS THIS PROBLEM AXISYMMETRIC?

***** PHYSICAL PARAMETERS (PARM) *****

MATERIAL PROPERTIES OF AIR (70 DEG. F, 1ATM):

1 GRAVITATIONAL ACCELERATION.... 0.3220E+02 (FT/SEC**2)
2 KINEMATIC VISCOSITY..... 0.1591E-03 (FT^2/SEC)
3 DENSITY..... 0.7559E-01 (LBM/FT**3)
DO YOU WANT TO CHANGE ANY OF THESE VALUES?

REFERENCE LENGTH..... 1.0000

DO YOU WANT TO CHANGE THE REFERENCE VALUE?

***** GRID CONSTRUCTION (GRID) *****

DO YOU WANT TO 1) USE AN OLD GRID
2) DEFINE A NEW GRID
2

DEFINE DOMAIN GEOMETRY: 0 = RECTANGULAR QUADRILATERAL
1 = GENERAL QUADRILATERAL
2 = CURVED-SIDED QUADRILATERAL
0

NAME OF FILE TO WRITE NEW GRID TO ?
DRIVEN.GRID
NUMBER OF MACRO DOMAINS IN X DIRECTION?

2
INITIAL X COORDINATE?
0

ENTER FINAL X COORDINATE OF MACRO DOMAIN 1
.5

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 1
1

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 1
8

ENTER FINAL X COORDINATE OF MACRO DOMAIN 2
1

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 2
1.

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 2
8

NUMBER OF MACRO DOMAINS IN Y DIRECTION?
2

INITIAL Y COORDINATE?

0

ENTER FINAL Y COORDINATE OF MACRO DOMAIN 1

.5

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 1

1

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 1

8

ENTER FINAL Y COORDINATE OF MACRO DOMAIN 2

1

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 2

1

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 2

8

NODES ARE NUMBERED IN THE ETAl DIRECTION

WOULD YOU LIKE TO SEE A PLOT OF THE GRID?

Y

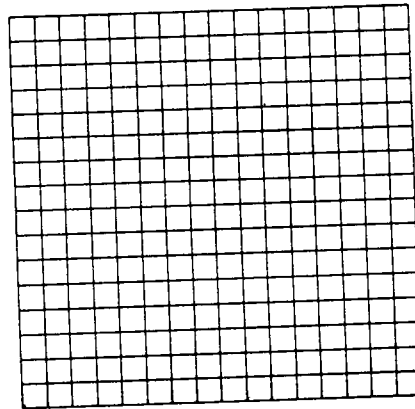


Figure 7. Driven Cavity Uniform 16x16 Grid.

WOULD YOU LIKE A FILE CONTAINING THE NODE MAP OF THE MACRO VERTICES?

Y

THEY ARE CONTAINED IN A FILE CALLED NODE.MAP, AND ARE PRINTED HERE ALSO.

273 281 289

103 110 119

1 8 17

DO YOU WANT TO CONTINUE WITH THIS GRID?

Y

***** INITIAL CONDITION (INIT) *****

SPECIFY AN INITIAL CONDITION: 1 ZERO (COLD START)
2 SPECIFY UNIFORM VALUE
3 READ FROM DATA FILE

1

***** BOUNDARY CONDITIONS (DIRI, NEUM, ROBN) *****

TO AID IN DEFINING BOUNDARY CONDITIONS, DO YOU WANT TO
SEE THE GRID PLOTTED WITH THE BOUNDARY NODE NUMBERS?

SPECIFY BOUNDARY CONDITIONS

* * * * * INLETS * * * * *

INLETS MUST BE LOCATED ON THE DOMAIN PERIMETER.
INLETS ARE SPECIFIED BY THE NODE ON EACH SIDE OF THE INLET.
INPUT THE INITIAL AND FINAL NODES IN A CCW DIRECTION.

HOW MANY INLETS ARE THERE?

0

* * * * * WALLS * * * * *

WALLS ARE INPUT BY THE BEGINNING AND ENDING NODES OF EACH SEGMENT IN A
COUNTERCLOCKWISE DIRECTION STARTING WITH THE FIRST INLET OR NODE.

1) INPUT INITIAL NODE OF SEGMENT.

2) INPUT FINAL NODE OF EACH CONTIGUOUS WALL SEGMENT.

3) WHEN A BREAK IS ENCOUNTERED (I.E., INLET OR OUTLET), INPUT A ZERO FOR
FINAL
NODE, AND PROCEED AS IN 1 AND 2.

4) WHEN THERE ARE NO MORE SEGMENTS, INPUT A ZERO FOR FINAL AND INITIAL
NODES.

DO YOU WANT TO SPECIFY ANY WALL-SEGMENTS WITH A NON-ZERO TANGENTIAL
VELOCITY?

Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1

1

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1

17

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

0

NO-SLIP WALL SEGMENT 1 NODE

1
2
3
4
5
6
7
8
9
10
11

12
13
14
15
16
17

WALL VELOCITY= 0.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 2

289

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

0

NO-SLIP WALL SEGMENT 2 NODE

17
34
51
68
85
102
119
136
153
170
187
204
221
238
255
272
289

WALL VELOCITY= 0.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 3

273

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

-1

NO-SLIP WALL SEGMENT 3 NODE

289
288
287
286
285
284
283
282
281

280
279
278
277
276
275
274
273

WALL VELOCITY= -1.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 4

1

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

0

NO-SLIP WALL SEGMENT 4 NODE

273
256
239
222
205
188
171
154
137
120
103
86
69
52
35
18
1

WALL VELOCITY= 0.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

0

IS THIS DATA CORRECT?

Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

0

DO YOU WANT TO DEFINE A BOUNDARY AS A SYMMETRY PLANE?

DO YOU WANT A NODE TO BE SET AT A REFERENCE VALUE OF STREAMFUNCTION?

DO YOU WANT TO SET THE PRESSURE AT A NODE TO ZERO?

DO YOU WISH TO SEE A PLOT
OF THE SPECIFIED BOUNDARIES?

Y

THE FOLLOWING PLOT SHOWS THE SELECTED GRID AND THE LOCATION OF
THE INLETS, OUTLETS, AND WALLS

PRESS [RETURN] TO CONTINUE

PLEASE WAIT.....

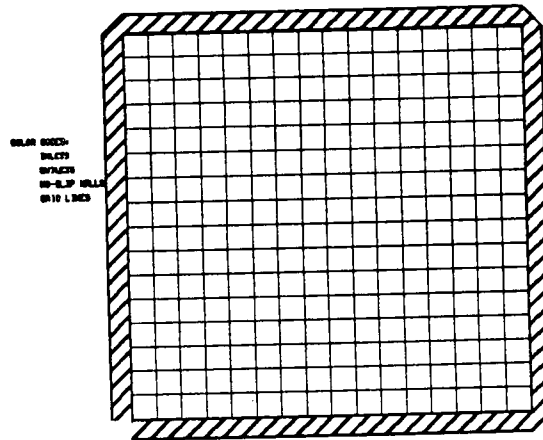


Figure 8. Driven Cavity 16x16 Boundary Plot.

***** CONTINUATION PARAMETERS (CONT) *****

ENTER NUMBER OF EFFECTIVE REYNOLDS NOS. (IF ZERO INPUT, ONLY THE PHYSICAL
RE.NO.
IS USED)

4

ENTER EFFECTIVE REYNOLDS NOS., ONE PER LINE

10

100

500

1000

EFFECTIVE REYNOLDS NUMBERS ARE 10.000 100.00 500.00 1000.0

DO YOU WANT TO CHANGE THESE?

DO YOU WANT TO USE ARTIFICIAL DISSIPATION CONTINUATION?

***** INTEGRATION PARAMETERS (INTE) *****

CHOOSE TYPE OF SOLUTION: 1) ASYMPTOTIC STEADY-STATE
2) TIME ACCURATE UNSTEADY

1

SHOULD VELOCITY BE POST-PROCESSED FROM 1) VORTICITY
2) STREAMFUNCTION

2

INTEGRATION CONTROL PARAMETERS

```

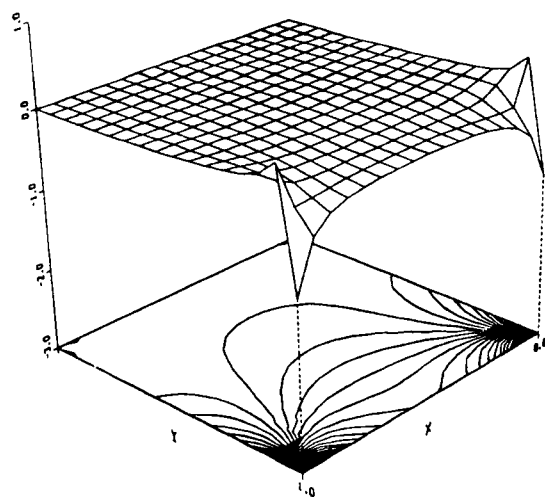
1 TSTART = TIME STATION AT WHICH SOLUTION IS STARTED..... 0.000000E+00
2 TSTOP  = TIME STATION AT WHICH INTEGRATION IS STOPPED.... 0.100000E+03
3 DTZERO = INITIAL INTEGRATION STEP SIZE..... 0.500000E-02
4 DTMAX  = MAXIMUM INTEGRATION STEP SIZE ..... 0.500000E+01
5 THETA1 = IMPLICITNESS FACTOR..... 0.100000E+01
6 THETA2 = ENERGY IMPLICITNESS FACTOR..... 0.100000E+01
7 NITMAX = MAXIMUM NUMBER OF TOTAL ITERATIONS ..... 100
8 EPS    = ITERATION CONVERGENCE REQUIREMENT..... 0.500000E-02
9 DQMAX  = INITIAL MAXIMUM VARIABLE CHANGE PER ITERATION .. 0.100000E+00
10 NPRINT = TIME STATION INTERVAL FOR PRINTING SOLUTION..... 100
DO YOU WANT TO CHANGE ANY OF THESE PARAMETERS?

```

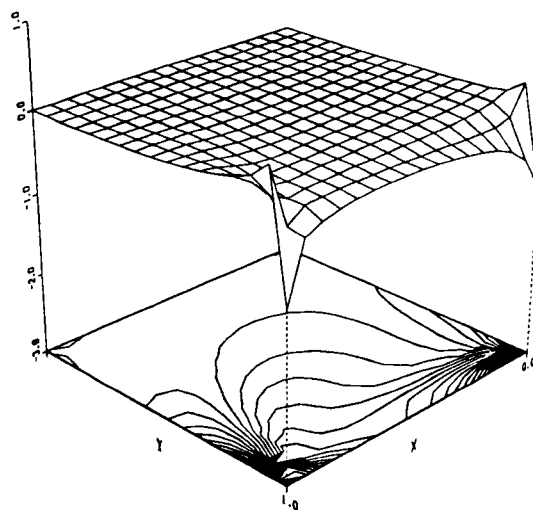
THE DATA FILE DRIVEN.DAT IS NOW READY FOR EXECUTION.

FORTRAN STOP

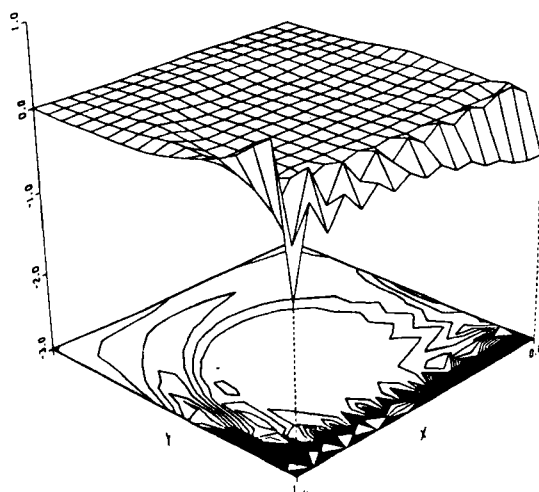
The solution variables of vorticity for $Re^{\text{ef}}=10$ to 1000 and velocity at $Re^{\text{ef}}=1000$ for the uniform 16x16 driven cavity are shown in Figure 9. Severe oscillations have rendered the solution highly inaccurate. This problem can be remeshed, adding more resolution at those areas of greatest diffusive error.



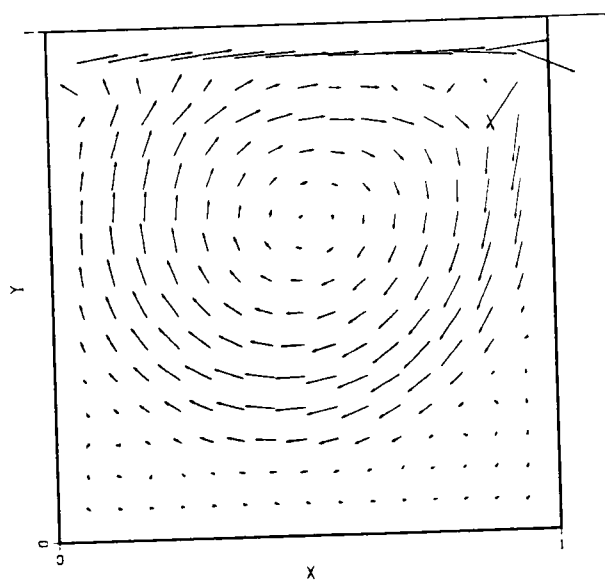
a)



b)



c)



d)

Figure 9. Driven Cavity Uniform 16x16 Grid Solution.
Distribution of Vorticity at a) $Re=10$, b) $Re=100$, c) $Re=1000$, and Velocity at d) $Re=1000$.

Non-Uniform Driven Cavity

The driven cavity is now remeshed with the same number of elements (16x16), but with a non-uniform spacing to cluster the nodes at the wall, necessary to reduce the dispersive error shown in Figure 9 without using artificial diffusion.

\$ R PREP5

THIS PROGRAM ALLOWS THE USER TO INTERACTIVELY CREATE AN INPUT DATA FILE FOR EXECUTION OF A 2-DIMENSIONAL FLUID/THERMAL CFD CODE.

- POSITIVE REPLIES TO QUESTIONS REQUIRE AN UPPER-CASE Y, NEGATIVE REPLIES REQUIRE ONLY A RETURN.

NAME OF FILE TO WRITE DATA TO ?
DRIVEN2.DAT

SPECIFY TERMINAL TYPE 1) TEK 4115
 2) TEK 4107
 3) VT 240
 4) IMAGEN
 5) UNIX WORKSTATION

3

***** TITLE (TITL) *****

WHAT IS THE TITLE OF THE PROBLEM?
DRIVEN CAVITY NON-UNIFORM 16X16

***** ALGORITHM AND PROBLEM PARAMETERS (TYPE) *****

SPECIFY COMPUTATIONAL METHOD TO USE:
 1) STREAMFUNCTION-VORTICITY
 2) CONTINUITY CONSTRAINT

1

FINITE ELEMENT BASIS DEGREE: 1
DIMENSIONAL ORDER OF PROBLEM: 2
NUMBER OF DEGREES OF FREEDOM / NODE: 2
ELEMENT TYPE: QUADRILATERALS

IS THIS PROBLEM ISOTHERMAL?
Y

DOES THIS PROBLEM HAVE TRANSPORT OF AN INERT SPECIES?

IS THIS PROBLEM AXISYMMETRIC?

***** PHYSICAL PARAMETERS (PARM) *****

MATERIAL PROPERTIES OF AIR (70 DEG. F, 1ATM):

1 GRAVITATIONAL ACCELERATION.... 0.3217E+02 (FT/SEC**2)
2 KINEMATIC VISCOSITY..... 0.1591E-03 (FT^2/SEC)
3 DENSITY..... 0.7559E-01 (LBM/FT**3)
DO YOU WANT TO CHANGE ANY OF THESE VALUES?

REFERENCE LENGTH..... 1.0000

DO YOU WANT TO CHANGE THE REFERENCE VALUE?

***** GRID CONSTRUCTION (GRID) *****

DO YOU WANT TO 1) USE AN OLD GRID
2) DEFINE A NEW GRID

2

DEFINE DOMAIN GEOMETRY: 0 = RECTANGULAR QUADRILATERAL
1 = GENERAL QUADRILATERAL
2 = CURVED-SIDED QUADRILATERAL

0

NAME OF FILE TO WRITE NEW GRID TO ?
DRIVEN2.GRID

NUMBER OF MACRO DOMAINS IN X DIRECTION?

2

INITIAL X COORDINATE?

0

ENTER FINAL X COORDINATE OF MACRO DOMAIN 1

.5

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 1

1.2

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 1

8

ENTER FINAL X COORDINATE OF MACRO DOMAIN 2

1.

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 2

.8

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 2

8

NUMBER OF MACRO DOMAINS IN Y DIRECTION?

2

INITIAL Y COORDINATE?

0

ENTER FINAL Y COORDINATE OF MACRO DOMAIN 1

.5

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 1

1.2

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 1

8

ENTER FINAL Y COORDINATE OF MACRO DOMAIN 2

1

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 2

.8

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 2

8

NODES ARE NUMBERED IN THE X DIRECTION

WOULD YOU LIKE TO SEE A PLOT OF THE GRID?

Y

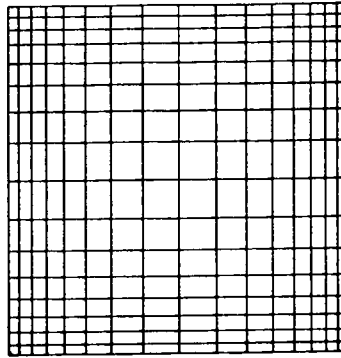


Figure 10. Driven Cavity Non-Uniform 16x16 Grid.

THE VERTICES OF THE MACRO DOMAINS ARE

```
273  281  289
137  145  153
  1    9   17
```

WOULD YOU LIKE A FILE CONTAINING THE NODE MAP OF THE MACRO VERTICES?

DO YOU WANT TO CONTINUE WITH THIS GRID?

Y

***** INITIAL CONDITION (INIT) *****

SPECIFY AN INITIAL CONDITION: 1 ZERO (COLD START)
2 SPECIFY UNIFORM VALUE
3 READ FROM DATA FILE

1

DO YOU WANT THE SOLUTION TO BE OUTPUT IN A FILE THAT CAN BE INTERPOLATED
ONTO ANOTHER GRID?

***** BOUNDARY CONDITIONS (DIRI, NEUM, ROBN) *****

TO AID IN DEFINING BOUNDARY CONDITIONS, DO YOU WANT TO
SEE THE GRID PLOTTED WITH THE BOUNDARY NODE NUMBERS?

SPECIFY BOUNDARY CONDITIONS

RECALL THAT NODES ARE NUMBERED IN THE X DIRECTION

* * * * * INLETS * * * * *

INLETS MUST BE LOCATED ON THE DOMAIN PERIMETER.
INLETS ARE SPECIFIED BY THE NODE ON EACH SIDE OF THE INLET.
INPUT THE INITIAL AND FINAL NODES IN A CCW DIRECTION.

HOW MANY INLETS ARE THERE?

0

* * * * * WALLS * * * * *

WALLS ARE INPUT BY THE BEGINNING AND ENDING NODES OF EACH SEGMENT IN A COUNTERCLOCKWISE DIRECTION STARTING WITH THE FIRST INLET OR NODE.

- 1) INPUT INITIAL NODE OF SEGMENT.
- 2) INPUT FINAL NODE OF EACH CONTIGUOUS WALL SEGMENT.
- 3) WHEN A BREAK IS ENCOUNTERED (I.E., INLET OR OUTLET), INPUT A ZERO FOR FINAL NODE, AND PROCEED AS IN 1 AND 2.
- 4) WHEN THERE ARE NO MORE SEGMENTS, INPUT A ZERO FOR FINAL AND INITIAL NODES.

DO YOU WANT TO SPECIFY ANY WALL-SEGMENTS WITH A NON-ZERO TANGENTIAL VELOCITY?

Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1

1

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1

17

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW BEING POSITIVE)

0

NO-SLIP WALL SEGMENT 1 NODE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

WALL VELOCITY= 0.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 2

289

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW BEING POSITIVE)

0

NO-SLIP WALL SEGMENT 2 NODE

17

34
51
68
85
102
119
136
153
170
187
204
221
238
255
272
289

WALL VELOCITY= 0.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 3

273

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

-1

NO-SLIP WALL SEGMENT 3 NODE

289
288
287
286
285
284
283
282
281
280
279
278
277
276
275
274
273

WALL VELOCITY= -1.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 4

1

ENTER PARALLEL VELOCITY OF WALL SEGMENT (BASED ON COUNTERCLOCKWISE FLOW
BEING POSITIVE)

0

NO-SLIP WALL SEGMENT 4 NODE

273

256
239
222
205
188
171
154
137
120
103
86
69
52
35
18
1

WALL VELOCITY= 0.0000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

0

IS THIS DATA CORRECT?

Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

0

DO YOU WANT TO DEFINE A BOUNDARY AS A SYMMETRY PLANE?

DO YOU WANT A NODE TO BE SET AT A REFERENCE VALUE OF STREAMFUNCTION?

DO YOU WANT TO SET THE PRESSURE AT A NODE TO ZERO?

DO YOU WISH TO SEE A PLOT
OF THE SPECIFIED BOUNDARIES?

N

***** CONTINUATION PARAMETERS (CONT) *****

ENTER NUMBER OF EFFECTIVE REYNOLDS NOS. IF NATURAL CONVECTION,
INPUT 1 REYNOLDS NO. WITH A VALUE OF ZERO. THEN THE EFFECTIVE REYNOLDS
NUMBER

WILL BE CALCULATED FROM THE SQUARE ROOT OF GRASHOF NUMBER.

4

ENTER EFFECTIVE REYNOLDS NOS., ONE PER LINE

10

100

1000

2000

EFFECTIVE REYNOLDS NUMBERS ARE 10.000 100.00 1000.0 2000.0

DO YOU WANT TO CHANGE THESE?

DO YOU WANT TO USE ARTIFICIAL DISSIPATION CONTINUATION?

***** INTEGRATION PARAMETERS (INTE) *****

CHOOSE TYPE OF SOLUTION: 1) ASYMPTOTIC STEADY-STATE
2) TIME ACCURATE UNSTEADY

```

1 SHOULD VELOCITY BE POST-PROCESSED FROM  1) VORTICITY
                                           2) STREAMFUNCTION
2

```

INTEGRATION CONTROL PARAMETERS

```

1 TSTART = TIME STATION AT WHICH SOLUTION IS STARTED..... 0.000000E+00
2 TSTOP  = TIME STATION AT WHICH INTEGRATION IS STOPPED.... 0.100000E+03
3 DTZERO = INITIAL INTEGRATION STEP SIZE..... 0.500000E-02
4 DTMAX  = MAXIMUM INTEGRATION STEP SIZE ..... 0.500000E+01
5 THETA1 = IMPLICITNESS FACTOR..... 0.100000E+01
6 THETA2 = ENERGY IMPLICITNESS FACTOR..... 0.100000E+01
7 NITMAX = MAXIMUM NUMBER OF TOTAL ITERATIONS ..... 100
8 EPS    = ITERATION CONVERGENCE REQUIREMENT..... 0.500000E-02
9 DQMAX  = INITIAL MAXIMUM VARIABLE CHANGE PER ITERATION .. 0.100000E+00
10 NPRINT = TIME STATION INTERVAL FOR PRINTING SOLUTION..... 100

```

DO YOU WANT TO CHANGE ANY OF THESE PARAMETERS?

THE DATA FILE DRIVEN2.DAT IS NOW READY FOR EXECUTION.

FORTRAN STOP

At an effective Reynolds number of 1000, the oscillations are markedly improved. Figure 11 shows the vorticity and velocity solution. There is still the "peak" vorticity at the corner due to the singularity of the driven wall moving to that corner. More nodes are located near the walls where the gradients are highest, and fewer are in the relatively quiet interior. Thus, non-uniform meshes can significantly improve the solution quality without the penalty of more nodes.

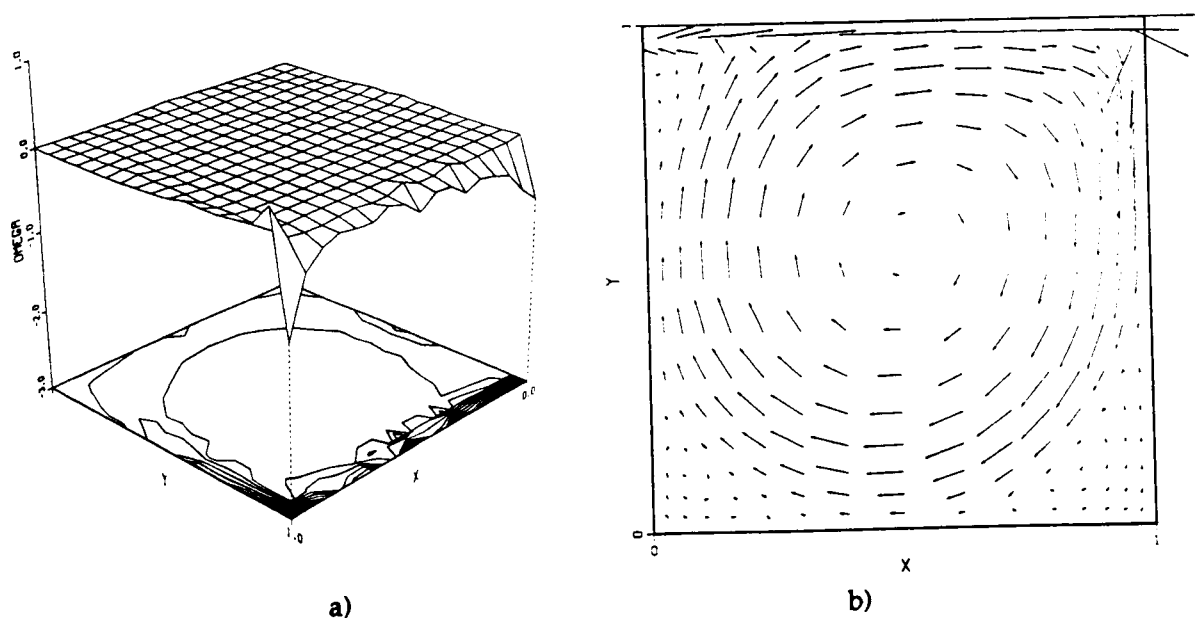


Figure 11. Driven Cavity Non-uniform Grid Solution at $Re=10^3$, a)Vorticity, b)Velocity.

The thermal cavity is now constructed of two opposing vertical walls at hot and cold temperatures. It uses the same uniform 16x16 grid of the driven cavity.

THIS PROGRAM ALLOWS THE USER TO INTERACTIVELY CREATE AN INPUT DATA FILE FOR EXECUTION OF A 2-DIMENSIONAL FLUID/THERMAL CFD CODE.

NAME OF FILE TO WRITE DATA TO ?
THERMAL.DAT

3

WHAT IS THE TITLE OF THE PROBLEM?
THERMAL CAVITY 16X16

SPECIFY COMPUTATIONAL METHOD TO USE:

1

```

1
FINITE ELEMENT BASIS DEGREE: 1
DIMENSIONAL ORDER OF PROBLEM: 2
NUMBER OF DEGREES OF FREEDOM / NODE: 2
ELEMENT TYPE: QUADRILATERALS

```

DOES THIS PROBLEM HAVE TRANSPORT OF AN INERT SPECIES?

***** PHYSICAL PARAMETERS (PARM) *****

MATERIAL PROPERTIES OF AIR (70 DEG. F, 1ATM):

```

1 GRAVITATIONAL ACCELERATION.... 0.3217E+02 (FT/SEC**2)
2 KINEMATIC VISCOSITY..... 0.1591E-03 (FT^2/SEC)
3 DENSITY..... 0.7559E-01 (LBM/FT**3)
4 SPECIFIC HEAT..... 0.2370E+00 (BTU/(LBM*DEG F))

```

5 THERMAL CONDUCTIVITY..... 0.4176E-05 (BTU/(FT*SEC*DEG F))
6 COEFF. OF THERMAL EXPANSION... 0.8182E-02 (1/DEG R)

DO YOU WANT TO CHANGE ANY OF THESE VALUES?

REFERENCE LENGTH..... 1.0000

DO YOU WANT TO CHANGE THE REFERENCE VALUE?

***** GRID CONSTRUCTION (GRID) *****

DO YOU WANT TO 1) USE AN OLD GRID
2) DEFINE A NEW GRID

1

DEFINE DOMAIN GEOMETRY: 0 = RECTANGULAR QUADRILATERAL
1 = GENERAL QUADRILATERAL
2 = CURVED-SIDED QUADRILATERAL

0

NAME OF FILE TO READ GRID FROM ?
DRIVEN.GRID

NODES ARE NUMBERED IN THE X DIRECTION

WOULD YOU LIKE TO SEE A PLOT OF THE GRID?

See Figure 7 on page 40 for the grid plot.

THE VERTICES OF THE MACRO DOMAINS ARE

273 289
1 17

WOULD YOU LIKE A FILE CONTAINING THE NODE MAP OF THE MACRO VERTICES?

DO YOU WANT TO CONTINUE WITH THIS GRID?
Y

***** INITIAL CONDITION (INIT) *****

SPECIFY AN INITIAL CONDITION: 1 ZERO (COLD START)
2 SPECIFY UNIFORM VALUE
3 READ FROM DATA FILE

1

DO YOU WANT TO SPECIFY THE REFERENCE TEMPERATURE AS THE UNIFORM INITIAL
TEMPERATURE CONDITION?

DO YOU WANT THE SOLUTION TO BE OUTPUT IN A FILE THAT CAN BE INTERPOLATED
ONTO ANOTHER GRID?

***** BOUNDARY CONDITIONS (DIRI, NEUM, ROBN) *****

TO AID IN DEFINING BOUNDARY CONDITIONS, DO YOU WANT TO
SEE THE GRID PLOTTED WITH THE BOUNDARY NODE NUMBERS?

SPECIFY BOUNDARY CONDITIONS

RECALL THAT NODES ARE NUMBERED IN THE X DIRECTION

* * * * * INLETS * * * * *

INLETS MUST BE LOCATED ON THE DOMAIN PERIMETER.
INLETS ARE SPECIFIED BY THE NODE ON EACH SIDE OF THE INLET.
INPUT THE INITIAL AND FINAL NODES IN A CCW DIRECTION.

HOW MANY INLETS ARE THERE?

0

* * * * * WALLS * * * * *

WALLS ARE INPUT BY THE BEGINNING AND ENDING NODES OF EACH SEGMENT IN A COUNTERCLOCKWISE DIRECTION STARTING WITH THE FIRST INLET OR NODE.

1) INPUT INITIAL NODE OF SEGMENT.

2) INPUT FINAL NODE OF EACH CONTIGUOUS WALL SEGMENT.

3) WHEN A BREAK IS ENCOUNTERED (I.E., INLET OR OUTLET), INPUT A ZERO FOR FINAL NODE, AND PROCEED AS IN 1 AND 2.

4) WHEN THERE ARE NO MORE SEGMENTS, INPUT A ZERO FOR FINAL AND INITIAL NODES.

DO YOU WANT TO SPECIFY ANY WALL-SEGMENTS WITH A NON-ZERO TANGENTIAL VELOCITY?

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1

1

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1

17

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
 2 - FIXED TEMPERATURE
 3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 1 NODE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 2

289

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

2

ENTER TEMPERATURE OF WALL SEGMENT (DEG. F)
2590000

NO-SLIP WALL SEGMENT 2	NODE	TEMPERATURE
	17	2.590E+06
	34	2.590E+06
	51	2.590E+06
	68	2.590E+06
	85	2.590E+06
	102	2.590E+06
	119	2.590E+06
	136	2.590E+06
	153	2.590E+06
	170	2.590E+06
	187	2.590E+06
	204	2.590E+06
	221	2.590E+06
	238	2.590E+06
	255	2.590E+06
	272	2.590E+06
	289	2.590E+06

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 3

273

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 3	NODE
	289
	288
	287
	286
	285
	284
	283
	282
	281
	280
	279
	278
	277
	276
	275
	274
	273

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 4

1

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
 2 - FIXED TEMPERATURE
 3 - FIXED HEAT FLUX

2

ENTER TEMPERATURE OF WALL SEGMENT (DEG. F)

0

NO-SLIP WALL SEGMENT 4	NODE	TEMPERATURE
	273	0.00000
	256	0.00000
	239	0.00000
	222	0.00000
	205	0.00000
	188	0.00000
	171	0.00000
	154	0.00000
	137	0.00000
	120	0.00000
	103	0.00000
	86	0.00000
	69	0.00000
	52	0.00000
	35	0.00000
	18	0.00000
	1	0.00000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

0

IS THIS DATA CORRECT?

Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

0

DO YOU WANT TO DEFINE A BOUNDARY AS A SYMMETRY PLANE?

DO YOU WANT A NODE TO BE SET AT A REFERENCE VALUE OF STREAMFUNCTION?

DO YOU WANT TO SET THE PRESSURE AT A NODE TO ZERO?

DO YOU WANT TO SPECIFY A UNIFORM TEMPERATURE ALONG SOME NODE SEGMENT?

DO YOU WANT TO SPECIFY A HEAT SOURCE?

DO YOU WISH TO SEE A PLOT
OF THE SPECIFIED BOUNDARIES?

See Figure 8 on page 44.

***** CONTINUATION PARAMETERS (CONT) *****

GR/RE^2 (ARCHIMEDES NO.) SHOULD BE NEAR 1 FOR OPTIMAL SOLUTION STABILITY.
DO YOU WANT TO CHANGE THIS?

ENTER NUMBER OF EFFECTIVE REYNOLDS NUMBERS. IF NATURAL CONVECTION, INPUT A ZERO. THEN EFFECTIVE REYNOLDS NO. WILL BE CALCULATED FROM THE SQUARE ROOT OF GRASHOF NUMBER.

0

EFFECTIVE REYNOLDS NUMBERS ARE 0.00000E+00
DO YOU WANT TO CHANGE THESE?

SHOULD EFFECTIVE PECLET NUMBERS BE CALCULATED FROM EFFECTIVE REYNOLDS NUMBER TIMES PRANDTL NUMBER? (IF NOT, YOU WILL HAVE TO INPUT THE PECLET NUMBERS.)

Y

FOR EFFECTIVE REYNOLDS NUMBER 0.0000E+00,
ENTER NUMBER OF CONTINUATION TEMPERATURE DIFFERENCES.

8

ENTER TEMPERATURE DIFFERENCES, ONE PER LINE.

10

100

1000

10000

100000

500000

1000000

2590000

FOR EFFECTIVE REYNOLDS NUMBER 0.0000E+00,

TEMPERATURE DIFFERENCES ARE:

10.000	100.00	1000.0	10000.	0.10000E+06
0.50000E+06	0.10000E+07	0.25900E+07		

DO YOU WANT TO CHANGE THESE?

ARE THERE ENERGY SOURCES?

DO YOU WANT TO USE ARTIFICIAL DISSIPATION CONTINUATION?

***** INTEGRATION PARAMETERS (INTE) *****

CHOOSE TYPE OF SOLUTION: 1) ASYMPTOTIC STEADY-STATE
2) TIME ACCURATE UNSTEADY

1

SHOULD VELOCITY BE POST-PROCESSED FROM 1) VORTICITY
2) STREAMFUNCTION

2

INTEGRATION CONTROL PARAMETERS

1	TSTART = TIME STATION AT WHICH SOLUTION IS STARTED.....	0.000000E+00
2	TSTOP = TIME STATION AT WHICH INTEGRATION IS STOPPED....	0.100000E+03
3	DTZERO = INITIAL INTEGRATION STEP SIZE.....	0.500000E-02
4	DTMAX = MAXIMUM INTEGRATION STEP SIZE	0.500000E+01
5	THETA1 = IMPLICITNESS FACTOR.....	0.100000E+01
6	THETA2 = ENERGY IMPLICITNESS FACTOR.....	0.100000E+01
7	NITMAX = MAXIMUM NUMBER OF TOTAL ITERATIONS	100
8	EPS = ITERATION CONVERGENCE REQUIREMENT.....	0.500000E-02

```
9 DQMAX = INITIAL MAXIMUM VARIABLE CHANGE PER ITERATION .. 0.100000E+00
10 NPRINT = TIME STATION INTERVAL FOR PRINTING SOLUTION..... 100

DO YOU WANT TO CHANGE ANY OF THESE PARAMETERS?
```

```
THE DATA FILE THERMAL.DAT IS NOW READY FOR EXECUTION.

FORTRAN STOP
```

The resulting velocity and temperature solution computed by the streamfunction-vorticity method is shown in Figure 12 for $Ra=7 \times 10^6$. At this large Rayleigh number, the single circulation cell has started to divide into an upper left and a lower right cell. Some oscillations in the velocity solution are seen, and the temperature also appears non-smooth. As one example remedy, artificial viscosity is added at a modest level of 0.1 and the solution recomputed, resulting in the solutions shown in Figure 13. The solution oscillations are indeed eliminated. Also gone are the forming opposing recirculation cells. Unfortunately, the dispersive error has thus been eliminated at the expense of losing a physical feature of the flow. This highlights the pitfall of using artificial diffusion, namely, the danger of obscuring flowfield details. The preferable remedy for non-smooth solutions due to dispersive error is to remesh to a finer grid, enabling the proper resolution of the solution.

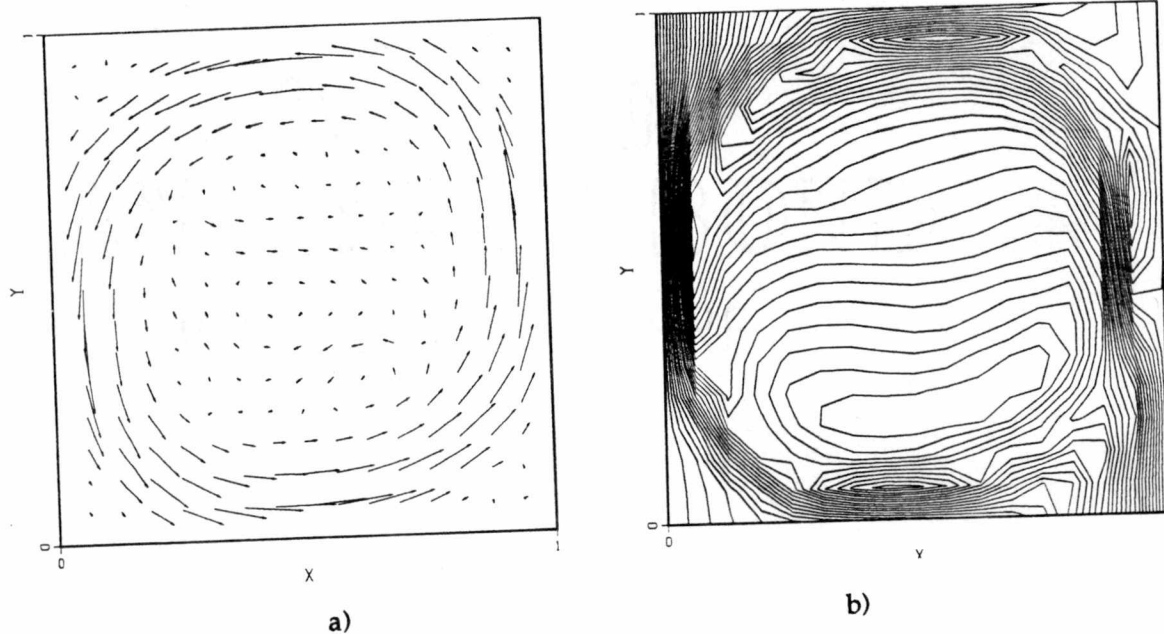


Figure 12. Thermal Cavity Solution at $Ra=7 \times 10^6$, a) Velocity, b) Temperature.

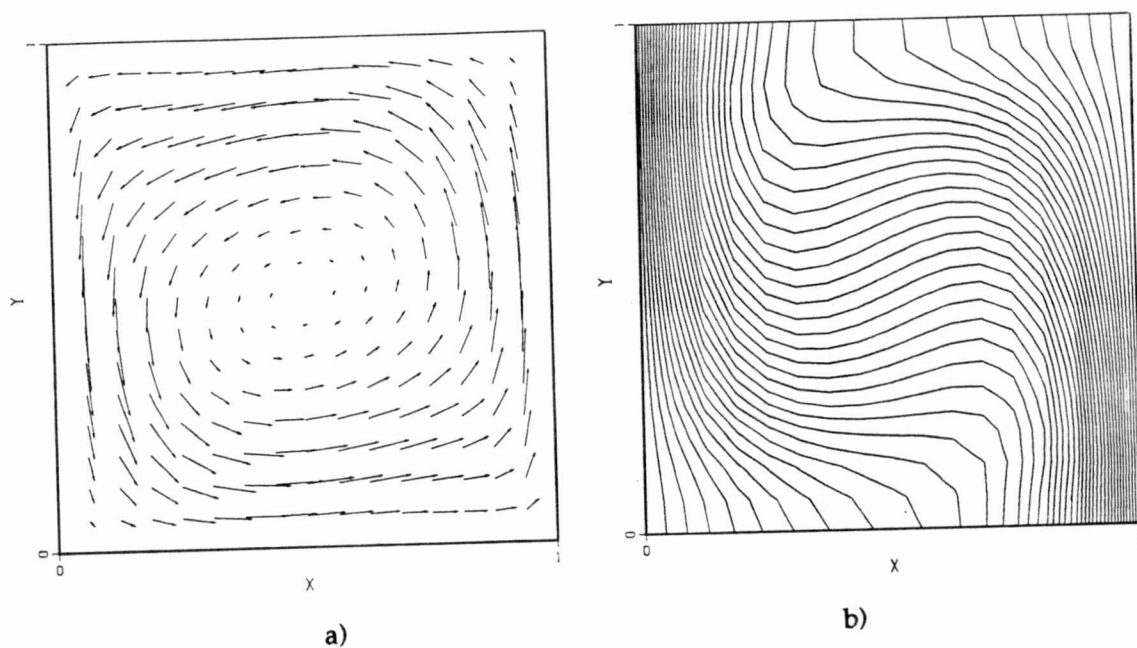


Figure 13. Thermal Cavity Solution at $Ra=7 \times 10^6$ with Artificial Diffusion=0.1, a) Velocity, b) Temperature.

Kitchen Hood

This example is one that might be encountered in the ventilation industry. The investigation is whether the convective strength of a hood is dominant over the bouyant strength of hot air from a stove. If not, the heated air will reenter the room, which is undesirable. Figure 14 shows the problem geometry and boundary conditions. The inlet diameter on the right side of 9 feet is used as the reference length.

Some thought must go into fixing a proper grid to the problem domain. A grid is constructed of a 50x50 quadrilateral mesh. Regions of high velocity, expected where the flow converges into and within the hood area, require a denser grid. Fewer nodes can support the solution gradients in the right half of the domain where the uniform slow velocity enters. Wall surfaces require node clustering, therefore there must be a macro region boundary along wall surfaces to progress discretization non-uniformly from the wall to the domain interior. Three macros in each direction are used to construct a grid, which are shown in Figure 15. PREPROS execution follows.

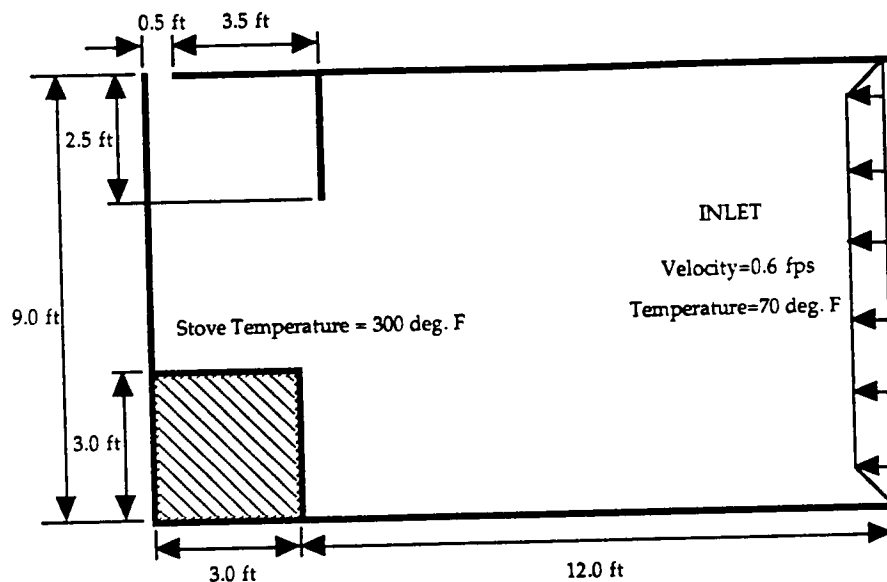


Figure 14. Kitchen Hood Problem.

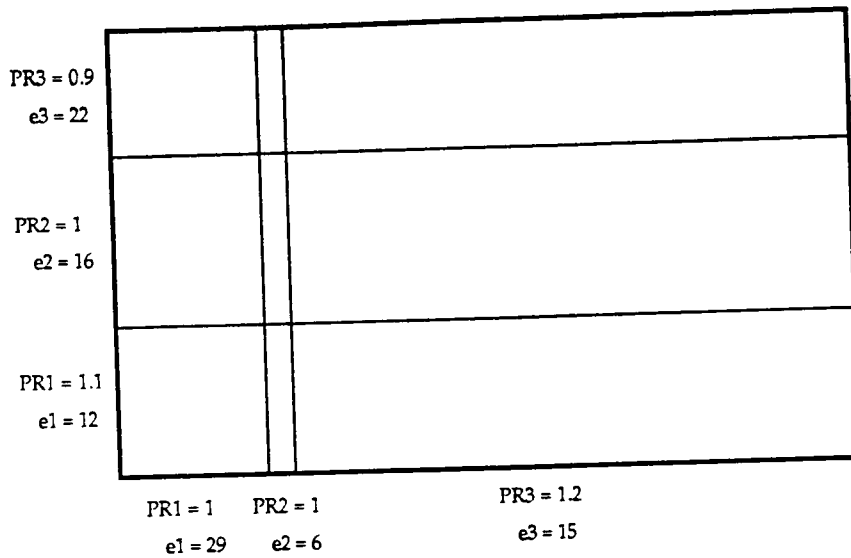


Figure 15. Kitchen Hood Macro Construction.

\$ RUN PREPROS

THIS PROGRAM ALLOWS THE USER TO INTERACTIVELY CREATE AN INPUT DATA FILE FOR EXECUTION OF A 2-DIMENSIONAL FLUID/THERMAL CFD CODE.

- POSITIVE REPLIES TO QUESTIONS REQUIRE AN UPPER-CASE Y, NEGATIVE REPLIES REQUIRE ONLY A RETURN.

NAME OF FILE TO WRITE DATA TO ?
HOOD.DAT

SPECIFY TERMINAL TYPE 1) TEK 4115
 2) TEK 4107
 3) VT 240
 4) IMAGEN
 5) UNIX WORKSTATION

4 ***** TITLE (TITL) *****

WHAT IS THE TITLE OF THE PROBLEM?
KITCHEN HOOD 50X50

***** ALGORITHM AND PROBLEM PARAMETERS (TYPE) *****

SPECIFY COMPUTATIONAL METHOD TO USE:
 1) STREAMFUNCTION-VORTICITY
 2) CONTINUITY CONSTRAINT

1
FINITE ELEMENT BASIS DEGREE: 1
DIMENSIONAL ORDER OF PROBLEM: 2
NUMBER OF DEGREES OF FREEDOM / NODE: 2

ELEMENT TYPE:

QUADRILATERALS

IS THIS PROBLEM ISOTHERMAL?

DOES THIS PROBLEM HAVE TRANSPORT OF AN INERT SPECIES?

IS THIS PROBLEM AXISYMMETRIC?

***** PHYSICAL PARAMETERS (PARM) *****

MATERIAL PROPERTIES OF AIR (70 DEG. F, 1ATM):

1 GRAVITATIONAL ACCELERATION....	0.3220E+02	(FT/SEC**2)
2 KINEMATIC VISCOSITY.....	0.1591E-03	(FT^2/SEC)
3 DENSITY.....	0.7559E-01	(LBM/FT**3)
4 SPECIFIC HEAT.....	0.2370E+00	(BTU/(LBM*DEG F))
5 THERMAL CONDUCTIVITY.....	0.4176E-05	(BTU/(FT*SEC*DEG F))
6 COEFF. OF THERMAL EXPANSION...	0.8182E-02	(1/DEG R)

DO YOU WANT TO CHANGE ANY OF THESE VALUES?

REFERENCE LENGTH..... 1.0000

DO YOU WANT TO CHANGE THE REFERENCE VALUE?

Y

ENTER VALUE OF REFERENCE LENGTH

9.

REFERENCE LENGTH..... 9.0000

DO YOU WANT TO CHANGE THE REFERENCE VALUE?

***** GRID CONSTRUCTION (GRID) *****

DO YOU WANT TO 1) USE AN OLD GRID
 2) DEFINE A NEW GRID

2

DEFINE DOMAIN GEOMETRY: 0 = RECTANGULAR QUADRILATERAL
 1 = GENERAL QUADRILATERAL
 2 = CURVED-SIDED QUADRILATERAL

1

NAME OF FILE TO WRITE NEW GRID TO ?

HOOD.GRID

NUMBER OF MACRO DOMAINS IN ETA1 DIRECTION?

3

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 1

1

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 1

29

ENTER PROGRESSION RATIO FOR MACRO DOMAIN 2

1

ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 2

6
 ENTER PROGRESSION RATIO FOR MACRO DOMAIN 3
 1.2
 ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 3
 15

NUMBER OF MACRO DOMAINS IN ETA2 DIRECTION?
 3
 ENTER PROGRESSION RATIO FOR MACRO DOMAIN 1
 1.1
 ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 1
 12
 ENTER PROGRESSION RATIO FOR MACRO DOMAIN 2
 1
 ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 2
 16
 ENTER PROGRESSION RATIO FOR MACRO DOMAIN 3
 .9
 ENTER NUMBER OF ELEMENTS IN MACRO DOMAIN 3
 22

MACRO VERTEX NUMBERING BEGINS IN THE LOWER LEFT CORNER AND CONTINUES
 IN THE ETA1 DIRECTION FIRST, THEN ALONG THE ETA2 DIRECTION.
 FOR THE MACRO DOMAINS YOU HAVE INPUT, THE VERTICES ARE NUMBERED:

```

13-----14-----15-----16
|         |         |         |
9-----10-----11-----12
|         |         |         |
5----- 6----- 7----- 8
|         |         |         |
1----- 2----- 3----- 4

```

FOR EACH VERTEX NODE, ENTER X AND Y COORDINATES ON THE SAME LINE,
 SEPARATED BY A COMMA OR A SPACE

INPUT MACRO VERTEX 1
 0 0

INPUT MACRO VERTEX 2
 3 0

INPUT MACRO VERTEX 3
 3.5 0

INPUT MACRO VERTEX 4
 15 0

INPUT MACRO VERTEX 5
 0 3

INPUT MACRO VERTEX 6
 3.5 3

INPUT MACRO VERTEX 7
 3.6 3

INPUT MACRO VERTEX 8
 15 3

INPUT MACRO VERTEX 9

0 6.5

INPUT MACRO VERTEX 10
3 6.5

INPUT MACRO VERTEX 11
3.5 6.5

INPUT MACRO VERTEX 12
15 6.5

INPUT MACRO VERTEX 13
0 9

INPUT MACRO VERTEX 14
3 9

INPUT MACRO VERTEX 15
3.5 9

INPUT MACRO VERTEX 16
15 9

NODES ARE NUMBERED IN THE ETA1 DIRECTION

WOULD YOU LIKE TO SEE A PLOT OF THE GRID?
Y

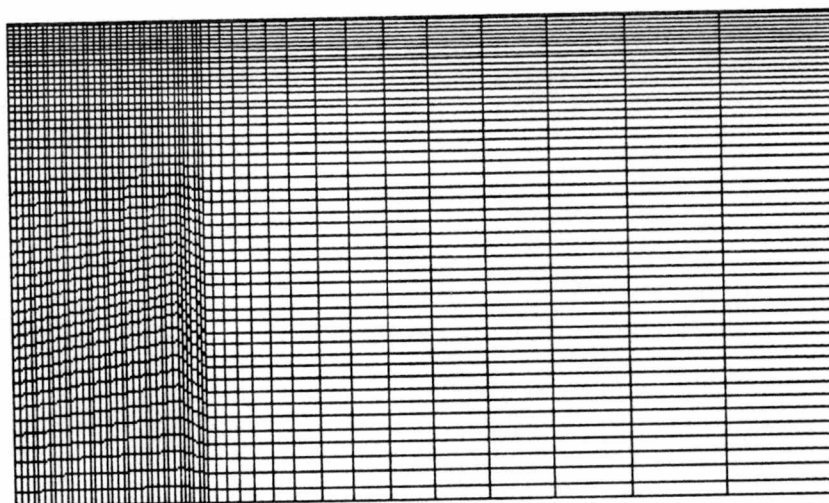


Figure 16. Incorrect Kitchen Hood Grid.

THE VERTICES OF THE MACRO DOMAINS ARE

2551	2580	2586	2601
1429	1458	1464	1479
613	642	648	663
1	30	36	51

WOULD YOU LIKE A FILE CONTAINING THE NODE MAP OF THE MACRO VERTICES?

DO YOU WANT TO CONTINUE WITH THIS GRID?
N

THE MACRO VERTEX NODES ARE NUMBERED AS FOLLOWS:

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

THEIR COORDINATES ARE:

VERTEX	X	Y
1	0.00000	0.00000
2	3.00000	0.00000
3	3.50000	0.00000
4	15.00000	0.00000
5	0.00000	3.00000
6	3.00000	3.60000
7	3.50000	3.00000
8	15.00000	3.00000
9	0.00000	6.50000
10	3.00000	6.50000
11	3.50000	6.50000
12	15.00000	6.50000
13	0.00000	9.00000
14	3.00000	9.00000
15	3.50000	9.00000
16	15.00000	9.00000

DO YOU WANT TO CHANGE THE CORNER NODE COORDINATES?
Y

WHICH VERTEX?

6

ENTER X AND Y COORDINATES.

3 3

THE MACRO VERTEX NODES ARE NUMBERED AS FOLLOWS:

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

THEIR COORDINATES ARE:

VERTEX	X	Y
1	0.00000	0.00000
2	3.00000	0.00000
3	3.50000	0.00000
4	15.00000	0.00000
5	0.00000	3.00000
6	3.00000	3.00000
7	3.50000	3.00000
8	15.00000	3.00000
9	0.00000	6.50000
10	3.00000	6.50000
11	3.50000	6.50000
12	15.00000	6.50000
13	0.00000	9.00000
14	3.00000	9.00000

15 3.50000 9.00000
16 15.00000 9.00000

DO YOU WANT TO CHANGE THE CORNER NODE COORDINATES?

NODES ARE NUMBERED IN THE ETA1 DIRECTION

WOULD YOU LIKE A FILE CONTAINING THE NODE MAP OF THE MACRO VERTICES?

Y

THEY ARE CONTAINED IN A FILE CALLED VERTEX.MAP, WHICH CAN BE ACCESSED NOW.

WOULD YOU LIKE TO SEE A PLOT OF THE GRID?

Y

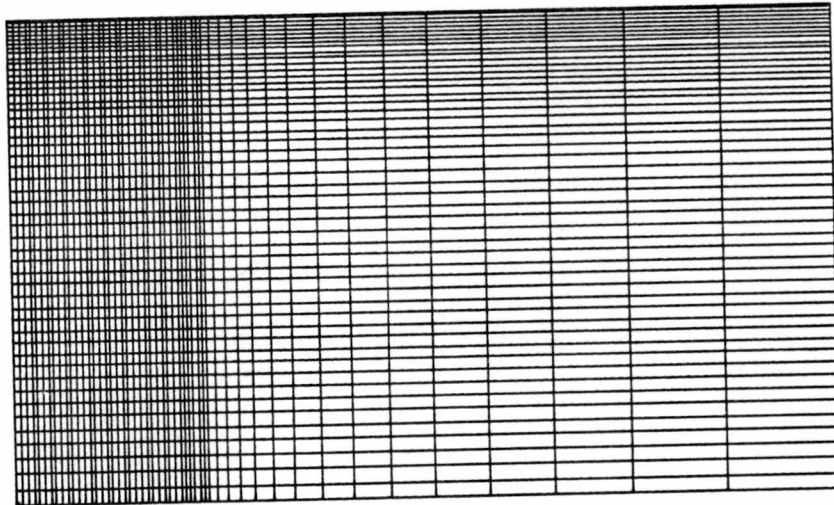


Figure 17. Corrected Kitchen Hood Grid.

DO YOU WANT TO CONTINUE WITH THIS GRID?

Y

***** INITIAL CONDITION (INIT) *****

SPECIFY AN INITIAL CONDITION: 1 ZERO (COLD START)
 2 SPECIFY UNIFORM VALUE
 3 READ FROM DATA FILE

1

DO YOU WANT TO SPECIFY THE REFERENCE TEMPERATURE AS THE UNIFORM INITIAL
TEMPERAT

URE CONDITION?

Y

DO YOU WANT THE SOLUTION TO BE OUTPUT IN A FILE THAT CAN BE INTERPOLATED
ONTO AN
OTHER GRID?

***** BOUNDARY CONDITIONS (DIRI, NEUM, ROBN) *****

SPECIFY BOUNDARY CONDITIONS

RECALL THAT NODES ARE NUMBERED IN THE ETAL DIRECTION

* * * * * INLETS * * * * *

INLETS MUST BE LOCATED ON THE DOMAIN PERIMETER.
INLETS ARE SPECIFIED BY THE NODE ON EACH SIDE OF THE INLET.
INPUT THE INITIAL AND FINAL NODES IN A CCW DIRECTION.

HOW MANY INLETS ARE THERE?

1

ENTER THE INITIAL NODE FOR INLET 1

51

ENTER THE FINAL NODE FOR INLET 1

2601

CHOOSE INLET VELOCITY DISTRIBUTION: 1) UNIFORM
2) NON-UNIFORM
3) PARABOLIC
4) HALF-PARABOLIC

1

INFLOW MUST BE NORMAL TO THE INLET BOUNDARY

ENTER INFLOW VELOCITY (ALL INFLOW IS POSITIVE)

.6

ENTER TEMPERATURE OF INLET STREAM (DEG. F)

70

INLET DEFINED IS:

INLET 1	NODE	INFLOW VELOCITY
	51	0.0000000
	102	0.6000000
	153	0.6000000
	204	0.6000000
	255	0.6000000
	306	0.6000000
	357	0.6000000
	408	0.6000000
	459	0.6000000
	510	0.6000000
	561	0.6000000
	612	0.6000000
	663	0.6000000
	714	0.6000000
	765	0.6000000
	816	0.6000000
	867	0.6000000
	918	0.6000000
	969	0.6000000

1020	0.6000000
1071	0.6000000
1122	0.6000000
1173	0.6000000
1224	0.6000000
1275	0.6000000
1326	0.6000000
1377	0.6000000
1428	0.6000000
1479	0.6000000
1530	0.6000000
1581	0.6000000
1632	0.6000000
1683	0.6000000
1734	0.6000000
1785	0.6000000
1836	0.6000000
1887	0.6000000
1938	0.6000000
1989	0.6000000
2040	0.6000000
2091	0.6000000
2142	0.6000000
2193	0.6000000
2244	0.6000000
2295	0.6000000
2346	0.6000000
2397	0.6000000
2448	0.6000000
2499	0.6000000
2550	0.6000000
2601	0.0000000

TEMPERATURE OF INLET STREAM= 70.00000 (DEGREES F)
 ARE THESE DATA CORRECT?
 Y

* * * * * WALLS * * * * *

WALLS ARE INPUT BY THE BEGINNING AND ENDING NODES OF EACH SEGMENT IN A COUNTERCLOCKWISE DIRECTION STARTING WITH THE FIRST INLET OR NODE.

1) INPUT INITIAL NODE OF SEGMENT.

2) INPUT FINAL NODE OF EACH CONTIGUOUS WALL SEGMENT.

3) WHEN A BREAK IS ENCOUNTERED (I.E., INLET OR OUTLET), INPUT A ZERO FOR FINAL
 NODE, AND PROCEED AS IN 1 AND 2.

4) WHEN THERE ARE NO MORE SEGMENTS, INPUT A ZERO FOR FINAL AND INITIAL
 NODES.

DO YOU WANT TO SPECIFY ANY WALL-SEGMENTS WITH A NON-ZERO TANGENTIAL
 VELOCITY?

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1
 2601

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 1
 2585

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
 2 - FIXED TEMPERATURE
 3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 1 NODE

2601
2600
2599
2598
2597
2596
2595
2594
2593
2592
2591
2590
2589
2588
2587
2586
2585

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 2

1463

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
 2 - FIXED TEMPERATURE
 3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 2 NODE

2585
2534
2483
2432
2381
2330
2279
2228
2177
2126
2075
2024
1973
1922
1871
1820
1769
1718
1667
1616
1565
1514
1463

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 3

1461

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 3 NODE

1463
1462
1461

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 4

2481

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 4 NODE

1461
1512
1563
1614
1665
1716
1767
1818
1869
1920
1971
2022
2073
2124
2175
2226
2277
2328
2379
2430
2481

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 5

2458

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 5 NODE

2481
2480
2479
2478
2477
2476
2475
2474
2473
2472
2471
2470
2469
2468
2467
2466
2465
2464
2463
2462
2461
2460
2459
2458

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 6

2560

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 6 NODE

2458
2509
2560

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 7

0

IS THIS DATA CORRECT?

Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 7
2551

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 7

613

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
2 - FIXED TEMPERATURE
3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 7 NODE

2551
2500

2449
 2398
 2347
 2296
 2245
 2194
 2143
 2092
 2041
 1990
 1939
 1888
 1837
 1786
 1735
 1684
 1633
 1582
 1531
 1480
 1429
 1378
 1327
 1276
 1225
 1174
 1123
 1072
 1021
 970
 919
 868
 817
 766
 715
 664
 613

ARE THESE DATA CORRECT?
 Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 8

NO-SLIP WALL SEGMENT 8	NODE	TEMPERATURE
	613	300.00000
	614	300.00000
	615	300.00000
	616	300.00000
	617	300.00000
	618	300.00000
	619	300.00000
	620	300.00000
	621	300.00000
	622	300.00000
	623	300.00000
	624	300.00000
	625	300.00000
	626	300.00000
	627	300.00000
	628	300.00000
	629	300.00000
	630	300.00000

631	300.00000
632	300.00000
633	300.00000
634	300.00000
635	300.00000
636	300.00000
637	300.00000
638	300.00000
639	300.00000
640	300.00000
641	300.00000
642	300.00000

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 9

30

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
 2 - FIXED TEMPERATURE
 3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 9 NODE

642
 591
 540
 489
 438
 387
 336
 285
 234
 183
 132
 81
 30

ARE THESE DATA CORRECT?

Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 10

51

ENTER WALL SEGMENT TEMPERATURE CONDITION: 1 - ADIABATIC
 2 - FIXED TEMPERATURE
 3 - FIXED HEAT FLUX

1

NO-SLIP WALL SEGMENT 10 NODE

30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40

41
42
43
44
45
46
47
48
49
50
51

ARE THESE DATA CORRECT?
Y

ENTER FINAL NODE IN CCW DIRECTION FOR WALL SEGMENT 11
0
IS THIS DATA CORRECT?
Y

ENTER INITIAL NODE IN CCW DIRECTION FOR WALL SEGMENT 11
0
DO YOU WANT TO DEFINE A BOUNDARY AS A SYMMETRY PLANE?

THE FOLLOWING SEGMENTS OF THE BOUNDARY ARE OUTLETS.

OUTLET 1	NODE
	2560
	2559
	2558
	2557
	2556
	2555
	2554
	2553
	2552
	2551

DO YOU WANT TO ASSIGN THIS OUTLET A TEMPERATURE?

DO YOU WANT A NODE TO BE SET AT A REFERENCE VALUE OF STREAMFUNCTION?

DO YOU WANT TO SPECIFY A UNIFORM TEMPERATURE ALONG SOME NODE SEGMENT?

DO YOU WANT TO SPECIFY A HEAT SOURCE?

DO YOU WISH TO SEE A PLOT
OF THE SPECIFIED BOUNDARIES?
Y

COLOR CODES:

INLETS

OUTLETS

NO-SLIP WALLS

GRID LINES

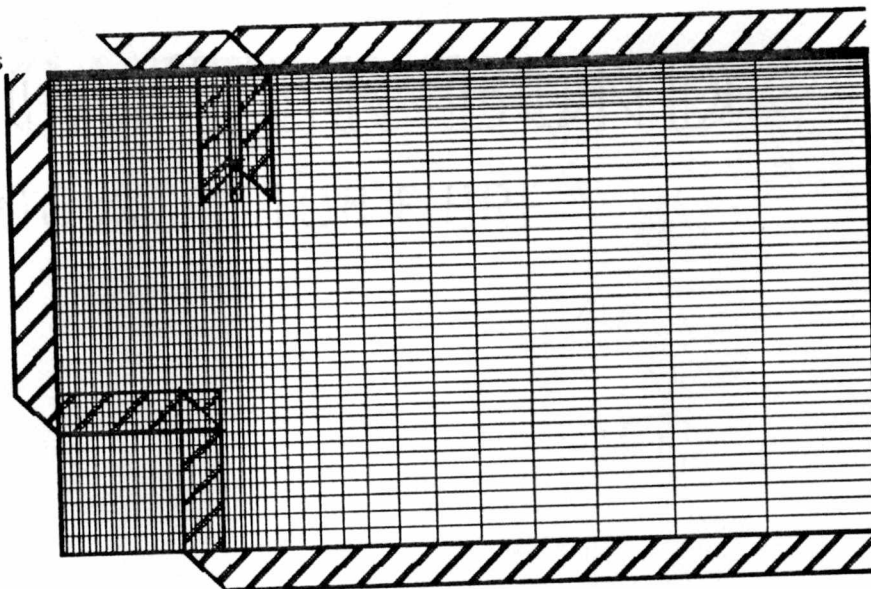


Figure 18. Kitchen Hood Boundary Plot.

***** CONTINUATION PARAMETERS (CONT) *****

GR/RE² (ARCHIMEDES NO.) SHOULD BE NEAR 1 FOR OPTIMAL SOLUTION STABILITY.
DO YOU WANT TO CHANGE THIS?

ENTER NUMBER OF EFFECTIVE REYNOLDS NOS. (IF ZERO INPUT, ONLY THE PHYSICAL
RE.NO.
IS USED)

3

ENTER EFFECTIVE REYNOLDS NOS., ONE PER LINE

10

100

500

EFFECTIVE REYNOLDS NUMBERS ARE 10.000 100.00 500.00
DO YOU WANT TO CHANGE THESE?

SHOULD EFFECTIVE PECLET NUMBERS BE CALCULATED FROM EFFECTIVE
REYNOLDS NUMBER TIMES PRANDTL NUMBER? (IF NOT, YOU WILL HAVE TO INPUT
THE PECLET NUMBERS.)

Y

FOR EFFECTIVE REYNOLDS NUMBER 10.00 ,
ENTER NUMBER OF CONTINUATION TEMPERATURE DIFFERENCES.

5

ENTER TEMPERATURE DIFFERENCES, ONE PER LINE.

10

50

100

160
230

FOR EFFECTIVE REYNOLDS NUMBER 100.0 ,
ENTER NUMBER OF CONTINUATION TEMPERATURE DIFFERENCES.
1
ENTER TEMPERATURE DIFFERENCES, ONE PER LINE.
230

FOR EFFECTIVE REYNOLDS NUMBER 500.0 ,
ENTER NUMBER OF CONTINUATION TEMPERATURE DIFFERENCES.
1
ENTER TEMPERATURE DIFFERENCES, ONE PER LINE.
230

FOR EFFECTIVE REYNOLDS NUMBER 10.00 ,
TEMPERATURE DIFFERENCES ARE:
10.000 50.000 100.00 160.00 230.00

FOR EFFECTIVE REYNOLDS NUMBER 100.0 ,
TEMPERATURE DIFFERENCES ARE:
230.00

FOR EFFECTIVE REYNOLDS NUMBER 500.0 ,
TEMPERATURE DIFFERENCES ARE:
230.00
DO YOU WANT TO CHANGE THESE?

ARE THERE ENERGY SOURCES?

DO YOU WANT TO USE ARTIFICIAL DISSIPATION CONTINUATION?

***** INTEGRATION PARAMETERS (INTE) *****

CHOOSE TYPE OF SOLUTION: 1) ASYMPTOTIC STEADY-STATE
2) TIME ACCURATE UNSTEADY

1
SHOULD VELOCITY BE POST-PROCESSED FROM 1) VORTICITY
2) STREAMFUNCTION
2

INTEGRATION CONTROL PARAMETERS

1 TSTART = TIME STATION AT WHICH SOLUTION IS STARTED..... 0.000000E+00
2 TSTOP = TIME STATION AT WHICH INTEGRATION IS STOPPED.. 0.100000E+0
3 DTZERO = INITIAL INTEGRATION STEP SIZE..... 0.500000E-02
4 DTMAX = MAXIMUM INTEGRATION STEP SIZE 0.500000E+01
5 THETA1 = IMPLICITNESS FACTOR..... 0.100000E+01
6 THETA2 = ENERGY IMPLICITNESS FACTOR..... 0.100000E+01
7 NITMAX = MAXIMUM NUMBER OF TOTAL ITERATIONS 100
8 EPS = ITERATION CONVERGENCE REQUIREMENT..... 0.500000E-02
9 DQMAX = INITIAL MAXIMUM VARIABLE CHANGE PER ITERATION 0.100000E+00
10 NPRINT = TIME STATION INTERVAL FOR PRINTING SOLUTION... 100

DO YOU WANT TO CHANGE ANY OF THESE PARAMETERS?

THE DATA FILE HOOD.DAT IS NOW READY FOR EXECUTION.

FORTRAN STOP

For the constructed hood problem, the velocity and temperature solutions are shown in Figure 19 and 20 respectively. Oscillations due to numerical dissipation are noticeably absent, indicating that the grid is sufficient to support the gradients of the solution variables. An important aspect of the grid construction needing elucidation is the 2-node wall suspended from the ceiling between the hood and the outlet. The reason for this construction is the boundary condition assumed for the outlet, which is vanishing normal derivative. Thus, if this drop ceiling were not there and the outlet was tangent with the ceiling, the flow along the wall would be required to sharply turn the corner and be unchanging in the direction of outlet flow. Satisfying these conditions is extremely difficult and can lead to the numerical problems in the vicinity of the outlet. Thus, knowledge of the boundary conditions is beneficial in creating the proper specifications to forestall numerical difficulties.

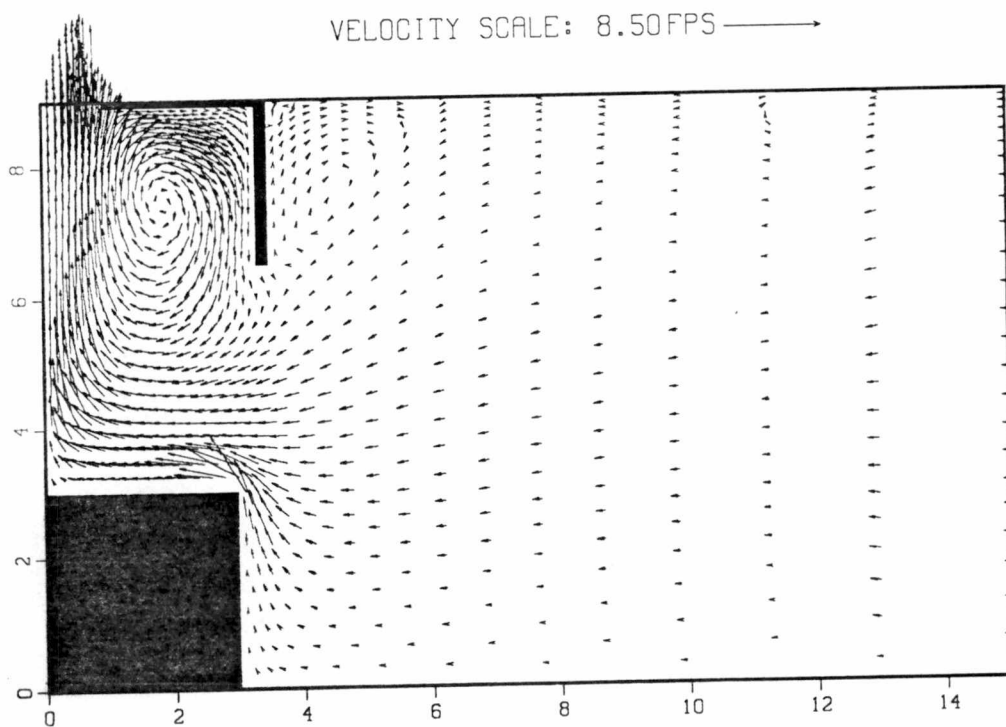


Figure 19. Hood Velocity Distribution at $Re=1000$.

CONTOUR INTERVAL = 5.75 DEG.

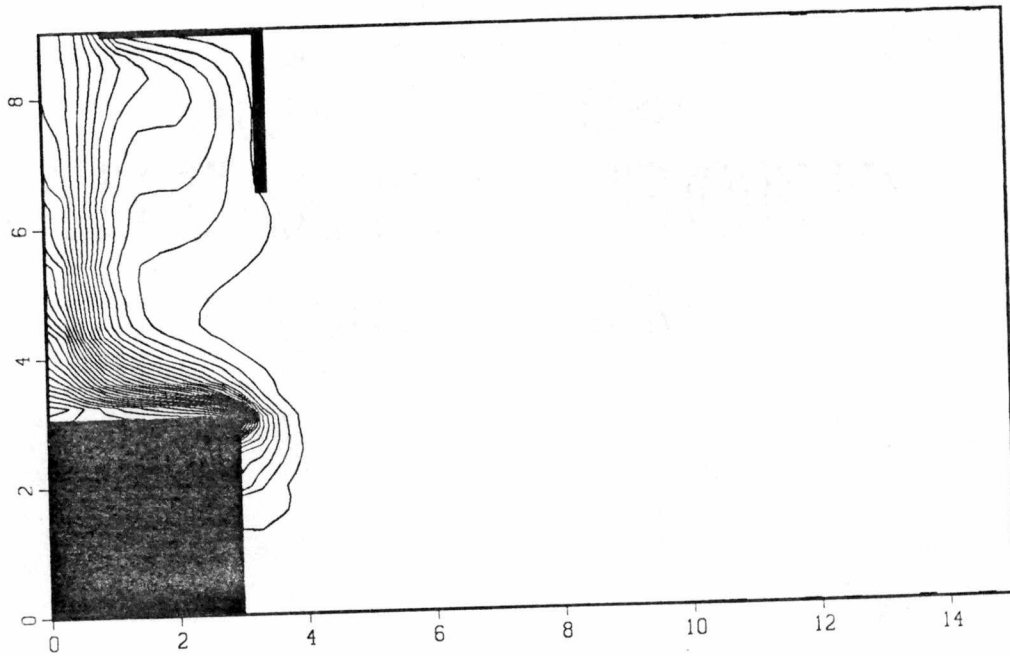


Figure 20. Hood Temperature Distribution at $Re=1000$.

CHAPTER 8

CONCLUSIONS

By integrating several CFD codes with a preprocessor that constructs the needed data file, their usability is enhanced, additionally enabling them to be used as a teaching tool. Neophyte users can learn the basic techniques of fluid and thermal computation, while veterans of the subject can construct and solve meaningful engineering problems in fluid/thermal analysis.

This utility would be greatly enhanced by adapting PREPROS to the graphical workstations becoming so common in academic and commercial organizations. Grid input could be simplified, and boundary conditions input more flexibly. The potential for a "seamless" computing environment is now within the grasp of the scientist/engineer, offering a great improvement in computing efficiency.

LIST OF REFERENCES

1. A.J. Baker, R.M. Kelso, *Analysis of Computational Fluid Dynamics Procedures Applicable to Predicting Room Air Motion, Part I*, ESM Report CFDL-89-2.1, University of Tennessee, 1989.
2. A.J. Baker, W.P. Noronha, P.T. Williams, R.M. Kelso, *Analysis of Computational Fluid Dynamics Procedures Applicable to Predicting Room Air Motion, Part II*, ESM Report CFDL-89-2.2, University of Tennessee, 1989.
3. A.J. Baker, *Finite Element Computational Fluid Mechanics*, 1983, Hemisphere, New York.
4. R.L. Panton, *Incompressible Flow*, 1984, Wiley-Interscience, New York.
5. E.M. White, *Viscous Fluid Flow*, 1974, McGraw-Hill, New York.

VITA

Leif E. Wilkening was born in Baltimore, Maryland on June 11, 1961. He attended local public schools and was graduated from Dulaney Senior High School in June 1979. The following September he entered the University of Delaware, and in June 1983 received a Bachelor of Science degree in Mechanical and Aerospace Engineering. That July, he accepted an engineering position with the AAI Corporation of Hunt Valley, Maryland. After taking part time graduate courses at the University of Maryland, he entered a Master's program in August, 1988, at the University of Tennessee. In August, 1991, he received a Master of Science degree in Engineering Science and Mechanics. He currently resides in Lake Lure, North Carolina with his wife Susan.