CONTRACTOR REPORT

THRUST VECTOR CONTROL,
HEAT TRANSFER MODELING

by

A. Leitner

July, 1986

Approved for public release; distribution unlimited.

Prepared for:
Naval Postgraduate School
Monterey, California 93943
The work reported herein was carried out for the Naval Postgraduate School by Mr. Amiram Leitner under Contract Number N62271-86-M-0206. The work presented in this report is part of a project sponsored by Naval Weapons Center on heat transfer modeling and is under the cognizance of Professors Matthew D. Kelleher and Robert H. Nunn.

Reproduction of all or part of this report is authorized.

This report was prepared by:

MR. AMIRAM LEITNER
Contracted Research Associate

Reviewed by:

MATTHEW D. KELLEHER
Professor
Department of Mechanical Engineering

ROBERT H. NUNN
Professor
Department of Mechanical Engineering

Anthony J. Healey
Chairman
Department of Mechanical Engineering

Released by:

JOHN N. DYER
Dean of Science and Engineering
The report presents heat transfer modeling of Thrust Vector control systems using the PHOENICS computer code.

Simple two-dimensional wedge and blunt bodies have been examined in supersonic cold flow, for both laminar and turbulent flow cases.

The research presents a numerical solution of the supersonic compressible viscous two-dimensional flow field. Post calculations were done to estimate skin friction coefficient, surface heat flux, heat transfer coefficient and Stanton number distributions in both wedge and blunt cases.
Thrust Vector Control Heat Transfer Modeling

Abstract

The report presents heat transfer modeling of Thrust Vector control systems using the PHOENICS computer code.

Simple two-dimensional wedge and blunt bodies have been examined in supersonic cold flow, for both laminar and turbulent flow cases.

The research presents a numerical solution of the supersonic compressible viscous two-dimensional flow field. Post calculations were done to estimate skin friction coefficient, surface heat flux, heat transfer coefficient and Stanton number distributions in both wedge and blunt cases.
NOMENCLATURE

C_p: Specific heat [J/kg·K]
C_1, C_2, C_D: Constants used in turbulent model
C_f: Skin friction coefficient
h: Enthalpy [J/kg]
h_c: Heat Transfer coefficient [W/m²·K]
M: Mach number
P: Pressure
Pr: Prandtl number
q: Heat flux
R: Gas constant [J/kg·K]
Re: Reynolds number
St: Stanton number
t: Time [S]
T: Temperature

GREEK LETTER SYMBOLS

γ: Specific heat ratio
δ: Boundary layer thickness
μ: Dynamic viscosity [kg·m/s]
σ: General exchange coefficient
ρ: Density [kg/m³]

\( \tau_k, \tau_c \): Constants used in turbulent model

ϕ: Any property at the grid node

SUBSCRIPTS

comp: Compressible value
eff: Effective value
inc: Incompressible value
r  Recovery
lam  Laminar quantity
t  Turbulent quantity
stat  Static values
W  Wall value
z  Local value in the flow direction
∞  Free stream value
# Table of Contents

1. Introduction ........................................... 1
2. PHOENICS description .................................. 4
   2.1 The structural principle of PHOENICS .......... 5
   2.2 Numerical scheme ................................ 5
3. Geometry and Dimensions ............................... 6
4. Assumptions ........................................... 7
5. Governing equations ................................... 10
6. Input Variables ......................................... 12
7. Boundary conditions ................................... 14
   7.1 Inlet ........................................... 14
   7.2 Outlet .......................................... 14
   7.3 Free stream boundary ............................ 15
   7.4 Solid wall ....................................... 15
   7.5 Wall function ................................... 15
   7.6 Boundary conditions in PHOENICS ............... 16
8. Mesh generation ......................................... 18
9. Heat Transfer Analysis ................................ 20
10. Code and Computer ..................................... 23
11. Results discussion .................................... 24
12. Conclusions and Recommendations ................. 25
    List of References .................................. 26

Appendices

   A - Satellite listing .................................. 38
   B - Ground listing .................................... 57
   C - Digital results ................................. 57
List of Figures

1. NWC jet vane configuration 27
2. Jet vane approximation 28
3. Wedge vane domain and grid 29
4. Blunt vane domain and grid 30
5. Re\(_x\) No. along the vane 31
6. C\(_f\) in laminar flow 32
7. C\(_f\) in turbulent flow 33
8. S\(_t\) in laminar flow 34
9. S\(_t\) in turbulent flow 35
10. Coefficient of heat convection in laminar flow 36
11. Coefficient of heat convection in turbulent flow 37
1. **Introduction**

This report describes a numerical analysis of heat transfer of a typical jet vane configuration used for thrust vector control. The work was carried out under contract Nos. N62271-85-M-0443 and N62271-86-M-0206, for the Naval Postgraduate School.

The tasks to be accomplished under the first contract were:

**Task I:** Formulate the conservation equations of momentum energy for two-dimensional, supersonic flow in geometries typical of thrust vector control systems.

**Task II:** Formulate boundary conditions for these equations appropriate to thrust vector control systems.

The tasks to be accomplished under the second contract were:

**Task I:** Continue and update the formulation of thrust vector control geometries based on the input from the Naval Weapons Center (NWC).

**Task II:** Construct the computational model for implementation in the PHOENICS code, of the thrust vector control geometries and flow conditions provided by NWC.

**Task III:** Run the PHOENICS code for the previously formulated models. Analyze and interpret the PHOENICS results for surface temperature and heat flux.

Thrust vector control components such as jet vanes and jet tabs are exposed to high speed hot gases at the exit of a rocket nozzle.

Estimation of the heat transfer from the hot exhaust gases to the vane is a major consideration in the correct design of a vane, and its ability to survive during its mission.
The research work was done under the framework of the tasks. A brief survey of what has been done according to the task is given:

Task 1 (M-0443): Heat transfer modeling of thrust vector control vane requires supersonic compressible viscous flow analysis.

In order to meet the requirements, the conservation differential equations of mass momentum energy and the two k-ε turbulent equations were formulated, and additional algebraic formulas for the relations between pressure density and the equation of state for ideal gas.

Task II (M-0443): The physical dimensions of the flow field grid were chosen and the boundary conditions for the Navier-Stokes, energy and the two k-q turbulent model equations were given.

Task I (M-0206): Working on the task, the actual configuration of a jet vane that is presently being tested at NWC has been modeled. The geometry being used is a wedge which has the same half angle and dimensions as the NWC jet vane.

Task II (M-0206): BFC (Body fitted coordinate) version of PHOENICS code (Ref. 3) was used for calculating the flow-field and heat transfer over the model. Using the BFC, a better geometrical approximation to vane shape could be achieved.

Non-Uniform grids have been utilized in order to model complicated regions in the flow field. Relaxation parameters and false timesteps options were adjusted to enable efficient computer runs with good convergence.

Task III (M-0206): In carrying out this task, four major runs have been analyzed:

Two geometric configurations were used: wedge vane and blunt vane (see Figures 1, 2, 3, 4); each one in both laminar and turbulent flow conditions.
Numerical results for fluid field and thermodynamic properties of pressure, temperature, density, Mach number and velocities are given in appendix C.

Post-calculations of heat transfer coefficient, skin friction coefficient and Stanton number are given in Figures (6, 7, 8, 9, 10, 11).

The next chapters describe in more detail the process of building the model and the analysis of the results.
2. PHOENICS Description

The present work addresses the heat transfer modeling of thrust vector control systems. In this effort the Navier-Stokes approach is applied by using a computer code which is capable of simulating a large number of fluid flow, heat transfer and chemical reaction processes which arise in industry and elsewhere. This code is called PHOENICS, which is an acronym standing for: 'Parabolic, Hyperbolic or Elliptic Numerical Integration Code Series.' The name comes from the fact that the differential equations of fluid flow, etc. arise in forms classified by mathematicians as parabolic, hyperbolic or elliptic; and PHOENICS solves these equations, whatever their form.

Built into PHOENICS are the major conservation laws of physics (mass, momentum, and energy) applied to a large number of continuous subdomains called 'cells,' into which the domain of study is artificially divided. The number of cells can be few or many according to the requirements of the problem. Because of numerical stability considerations the restrictions on cell refinement can become particularly burdensome in the calculation of a turbulent boundary layer where a very fine mesh near the wall may be required.

When supplied with appropriate information concerning: the physical properties of the materials, the geometrical and other constraints, the inlet and/or initial conditions, PHOENICS computes the corresponding solutions to the relevant differential equations, expressing them as tables of numbers describing the field of velocity, temperature concentration, etc.

Detailed information about PHOENICS is given in [Ref. 3].
2.1 The Structural Principle of PHOENICS

The code consists of three major parts: Satellite subroutine, Ground subroutine and Earth library.

The satellite subroutine is the main input subroutine and should provide the answers to the questions:
- what kind of process is to be simulated
- what are the properties of the fluid
- what are the shape and size of the domain
- how fine is the grid to be employed
- to what degree of accuracy is the calculation to be continued
- and what output should be provided

Ground subroutine is active during the computing process and is used for updating properties which vary with time, temperature, etc. For example: viscosity depends on temperature or density depends upon pressure and temperatures, etc.

Earth library is the main solver generator. It is given as a binary library and does not enable the user access to the source code.

2.2 Numerical Scheme

The numerical scheme used by the code is the simpler (semi-implicit method for pressure-linked equations revised) (Ref. 9). The scheme was developed by Patankar, S. V. and Spalding, D. B.

The scheme requires an additional dependent variable, the pressure correction, which has no physical meaning but should take part in the process.

The value of the pressure correction should tend to zero in the convergence process.

Two additional differential equations are solved: for the pressure, and for the pressure correction.
3. **Geometry and Dimensions**

Symmetrical 2-D planar geometry, which is shown in Figure 2, was chosen to be the approximation of the MWC vane in Figure 1.

Two geometrical profiles were examined, one with wedge leading edge and the second with blunt leading edge.

The dimensions of the domain in Figure 3 and 4 satisfy aspect ratio of 10:1 in the vertical y coordinate. A high aspect ratio in the coordinate is important for the assumption of free stream conditions at the upper boundary.
4. Assumptions

Postulating the right or the wrong assumptions has the most influence on modeling process. The stage was carried out very carefully in order to make the most compatible model with reality.

4.1 Steady state:
The modeling assumes steady state physical phenomenon process.

\[ \frac{\partial}{\partial t} (\text{all properties}) = 0 \]

This is a valid assumption since the time constant for the convection process is much shorter than the time constant for the wall conduction.

By assuming the wall temperature to be constant, the two procedures are decoupled.

In hot flow it is important to run the code for a wide range of wall temperature which will take into account the influence of different temperatures on the heat convection process.

4.2 Cold Air Flow
Ambient temperature air flow which was utilized by NWC experiments is being used in the computations.

4.3 Ideal Gas
The gas is assumed to satisfy the ideal gas equation of state

\[ p = \rho RT \]

This is a fairly good assumption for nonreactive gas flow. In spite of the values of static temperature can decrease to 200[k], the density remain relatively low.
This assumption is an important simplification to the solution in Ref. 10 which used the isentropic relation between pressure and density instead

\[
\frac{\rho}{\rho_0} = \left(\frac{P}{P_0}\right)^{1/\gamma}
\]  

(4.2)

4.4 Constant Pr, γ:

Prandtl number and γ (ratio of specific heats) were found to have negligible variations in the temperature range of the model. (200k - 350k)

4.5 Varying Viscosity and Thermal Conductivity:

\( \mu \) and \( k \) are much more dependent on temperature especially very close to the solid wall where values of \( \mu \) and \( k \) influence strongly the shear and heat transfer mechanism. To account for the temperature dependence power law relations have been formulated for \( \mu \) and \( k \).

\[
\mu = \mu_0 \left(\frac{T}{T_0}\right)^{0.666}
\]

(4.3)

\[
k = k_0 \left(\frac{T}{T_0}\right)^{0.666}
\]

(4.4)

4.6 Parallel Flow

Gas flow at the exit of the exhaust nozzle is more likely to be a conic source flow than parallel flow.

If the half angle of the nozzle is small, \( (\alpha < 15^\circ) \), parallel flow is a good assumption

4.7 Negligible Radiation

Assessments that were done showed that heat convection is at least one order of magnitude greater than heat flux by radiation.
4.8 Laminar and Turbulent Solutions

In order to overcome lack of ability to predict transition, separated laminar and turbulent calculations were done for each case. The turbulent solution utilizes the (k-ε) eddy viscosity model Ref. 5.

4.9 Constant Wall Temperature

The vane wall is assumed to have constant temperature during the time of calculation.
5. Governing Equations

The conservation equations for the compressible flow of the mathematical model consists of a viscous, Newtonian perfect gas consisting of the following six differential equations:

Conservation of Mass:
\[
\frac{3}{3t} (\rho) + \nabla (\rho \nabla) = 0
\]  
(5.1)

Conservation of momentum:
\[
\frac{3}{3t} (\rho \nabla) + \nabla (\rho \nabla - \mu \nabla \nabla) \nabla \nabla
\]  
(5.2)

where \( \nabla \) is \( V \) or \( W \) velocity component for \( y \) and \( z \) direction.

Conservation of Energy
\[
\frac{3}{3t} (\rho h) + \nabla (\rho \nabla h - \frac{\mu}{\rho} \nabla h) = \frac{DP}{dt}
\]  
(5.3)

where \( h \) is the total enthalpy.

\[
h = C_p T_o
\]

where To is the total temperature

\[
T_o = T_{stat}(1 + \frac{\gamma - 1}{2} M^2)
\]

In the case of laminar flow the governing equations (5.1), (5.2), (5.3) are sufficient to determine a solution when proper boundary conditions are applied and the equation of state (4.1) is provided.

Turbulence Model:

In turbulent flow it is necessary to hypothesize a turbulence model relating the turbulent viscosity to the other problem variables.
The model used in PHOENICS is the eddy viscosity (k-ε) model [Ref. 3, Ref. 5]. In this model k, the turbulent kinetic energy and ε, the turbulence dissipation rate, are treated as properties of the flow and conservation equations are postulated for these properties. The two conservation equations are: one for k, the kinetic energy of turbulence:

\[
\frac{\partial}{\partial t} \left( \frac{\partial k}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\partial k}{\partial y} \right) + \frac{\partial}{\partial z} \left( \frac{\partial k}{\partial z} \right) = \frac{\partial}{\partial x} \left( \nu \frac{\partial k}{\partial x} \right) + \frac{\partial}{\partial y} \left( \nu \frac{\partial k}{\partial y} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial k}{\partial z} \right) - G_k - \varepsilon \tag{5.4}
\]

Second equation for ε, the dissipation rate of turbulence

\[
\frac{\partial}{\partial t} \frac{\partial \varepsilon}{\partial x} \left( \frac{\partial \varepsilon}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\partial \varepsilon}{\partial y} \right) + \frac{\partial}{\partial z} \left( \frac{\partial \varepsilon}{\partial z} \right) = \frac{\partial}{\partial x} \left( \nu \frac{\partial \varepsilon}{\partial x} \right) + \frac{\partial}{\partial y} \left( \nu \frac{\partial \varepsilon}{\partial y} \right) + \frac{\partial}{\partial z} \left( \nu \frac{\partial \varepsilon}{\partial z} \right) - \left( C_1 G_k - C_2 \varepsilon \right) \tag{5.5}
\]

where

\[
G_k = \nu \left[ \frac{\partial \bar{u}_i}{\partial y} + \frac{\partial \bar{u}_j}{\partial x} \right] \frac{\partial \bar{u}_i}{\partial y} \tag{5.6}
\]

\[
\nu_{\text{eff}} = \nu_{\text{lam}} + \mu_c \frac{k^2}{\varepsilon} \tag{5.7}
\]

c1, c2, \sigma_k, \sigma_\varepsilon, c_\mu are empirical constants which are provided in PHOENICS.

The reason for using the (k-ε) model is because it is the most verified model for engineering applications. It combines simplicity, universality, and realism of predictions in most cases.

Two additional differential equations are solved also in order to satisfy the SIMPLER algorithm as was mentioned in chapter 2.2. The description of the pressure and pressure correction equations is provided by Ref. 9.
6. **Input Variables**

The properties of mach no., stagnation presence and temperature of the gas were provided by NWC; additional properties were taken from air tables:

Mach number:
\[ M_{\infty} = 3.2 \]

Stagnation pressure:
\[ P_0 = 55.10^5 \,[\text{Pa}] \]

Stagnation temperature:
\[ T_0 = 555.55 \,[\text{K}] \]

Gas constant
\[ R = 287. \,[\text{J/kg} \cdot \text{k}] \]

Specific heat ratio
\[ \gamma = 1.35 \]

Laminar Prandtl Number
\[ Pr = 0.7 \]

Turbulent Prandtl Number
\[ Pr_t = 0.9 \]

Constant Pressure Specific Heat
\[ C_p = R/(1-1/\gamma) \,[\text{J/kg} \cdot \text{k}] \]

Laminar Viscosity
\[ \mu = 0.1716 \times 10^{-5} \times (T/273)^{0.666} \]

Thermal Conductivity
\[ k = \mu \, C_p / Pr \]

The gas properties in the inlet boundary are equivalent to the properties at nozzle exit. Inlet properties are calculated from the stagnation values in the combustion chamber. The calculation was done by assuming one dimensional
isentropic expansion from combustion chamber to the nozzle exit (inlet for the vane).

\[
\begin{align*}
\text{Pressure} & \quad P_1 = P_0 \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\gamma/\gamma-1} \\
\text{Temperature} & \quad T_1 = T_0 \left(1 + \frac{\gamma - 1}{2} M^2\right) \\
\text{Density} & \quad \rho_1 = P/RT \\
\text{Enthalpy} & \quad h_1 = C_p T_1 \\
\text{Sonic Velocity} & \quad C_1 = \sqrt{\frac{\gamma R T_1}{\gamma - 1}} \\
\text{Velocity} & \quad W_1 = C \cdot M
\end{align*}
\]

The subscript \(i\) signifies inlet property
7. Boundary Conditions

The flow field described in Figures 3, 4 has four boundaries, which can be named: inlet, outlet, freestream boundary and solid wall.

Super sonic flows have a hyperbolic mathematical nature. The field consists of influence zones, the flow at every point is governed only by its influence zone, basically by the upwind stream.

As a consequence from the discussion, it's obvious that the outlet boundary condition has no influence on the upstream flow. The boundary values that are given at the outlet are to satisfy some numerical needs only.

7.1 Inlet

Parallel uniform flow with known velocity, enthalpy, pressure, and density: equation (6.1), (6.3), (6.4), (6.6) are given at the left boundary of the grid. In PHOENICS this is specified as the LOW side of the first Z cell.

In turbulent flow, boundary conditions are supplied for k and ε. The values that are given are based on empirical values:

\[ k_1 = 0.0 \frac{w_1^2}{k^2} \]  
\[ \epsilon_1 = 0.16 \frac{k^{1.5}/(5*GH)}{k^2} \]

where GH is half the vane thickness

7.2 Outlet

As was mentioned previously, the outlet has negligible effect on the results. The only property that is specified at the outlet is the pressure.
7.3 **Freestream Boundary**

Assuming that the upper boundary is chosen to be far enough away, the default boundary condition option of PHOENICS is used. This implies a line of symmetry where all gradients are zero.

7.4 **Solid Wall**

Zero velocity and constant wall enthalpy (temperature) are assumed on the wall. In PHOENICS the wall is the SOUTH side of the first y cell. The high enthalpy and velocity gradients near the wall demands a refined grid close to the wall. Values of shear stress and heat flux are calculated to first order accuracy using:

\[
\tau_w = \mu \frac{\partial w}{\partial y} = \mu \frac{W_1}{\Delta y_1/2} \tag{7.1}
\]

\[
q_w = \frac{\mu}{Pr} \frac{\partial h}{\partial y} = \frac{\mu}{Pr} \frac{h_1 - h_w}{\Delta y_1/2} \tag{7.2}
\]

In turbulent flow, a wall function is used to provide the wall condition for velocity, enthalpy, k, and \( \varepsilon \).

7.5 **Wall Function**

The wall problem in the numerical computation of flows, especially in turbulent flow, is an old one and most authors have adopted similar techniques. In effect they "bridge over" the region very close to the wall by introducing special functions which are called wall functions. These are often empirical in origin. Accounts may be found in Ref. 11.

The problem arises as follows. Turbulence dies out, close to the wall, because the no slip condition and the rigidity of the wall make all the velocity components fall to zero. The consequence is that the effective
viscosity and other transport properties fall there to their laminar values and the result is a rapid variation with distance from the wall both of the \( \phi \)'s and of their gradients.

Where \( \phi \) signifies general dependent variable, it is possible to compute these variations in detail, by using a computer code such as PHOENICS on two conditions:

(i) the grid points must be packed into the region of steep gradient changes closely enough for sufficient numerical accuracy to be obtained

(ii) the functions appearing in the turbulence model equations must properly represent the influence of local Reynolds number on turbulence.

Under the conditions above, the wall function sequences in the program act as follows:

The Reynolds number is first evaluated, based on the resultant velocity parallel to the wall, on the distance from the wall to the grid node and on density and laminar viscosity. If this Reynolds number is less than 132.25 (the value at which the laminar and turbulent wall function intersect) a laminar wall function is used. If this Reynolds number turns out to be greater than 132.25 the velocity variation is logarithmic and the corresponding shear stress coefficient is evaluated. This corresponds to the commonly used "log law" wall function. [Ref. 4]

7.6 Boundary Conditions in Phoenics

PHOENICS utilizes source terms for creating boundary conditions. The form of the source term of each dependent variable \( \phi \) is:

\[
S_{\phi} = ([\text{m}] + C_\phi) (V_{\phi} - \phi_p)
\]

(7.3)

where: \( \text{m} \) - is mass flux source
\( \phi_p \) - is the value of the dependent variable at point near the boundary
\( C_\phi, V_{\phi} \) - two coefficients specified by the user. The source term for
mass flux is simply

\[ S_m = C_m (V_m - P_p) \]  \hspace{1cm} (7.4)

where: \( P_p \) is the pressure near the boundary and \( C_m, V_m \) are two coefficients.

The values of \( C_\phi \) and \( V_\phi \) for the dependent variables in SATELLITE are: At the Inlet:

\[ C_m = 2 \frac{\gamma}{\gamma-1} \frac{1}{W_1} \]  \hspace{1cm} (7.5)
\[ V_m = \frac{P_o \rho_i}{P_0} \]  \hspace{1cm} (7.6)
\[ C_w = C_h = C_k = C_\varepsilon = 0. \]  \hspace{1cm} (7.7)
\[ V_w = W_i \]  \hspace{1cm} (7.8)
\[ V_h = h_i \]  \hspace{1cm} (7.9)
\[ V_k = K_i \]  \hspace{1cm} (7.10)
\[ V_\varepsilon = \varepsilon_i \]  \hspace{1cm} (7.11)

At the Outlet:

\[ C_m = 1000 \times W_i \frac{P_i}{P_1} \]  \hspace{1cm} (7.12)
\[ V_m = P_1 \]  \hspace{1cm} (7.13)

At the Wall (laminar)

\[ C_w = \frac{\mu}{0.5 \Delta w_1} \]  \hspace{1cm} (7.14)
\[ V_w = 0 \]  \hspace{1cm} (7.15)
\[ C_h = \frac{\mu}{Pr/(0.5 \Delta w_1)} \]  \hspace{1cm} (7.16)
\[ V_h = C_p \times T_w \]  \hspace{1cm} (7.17)

At the Wall (turbulent)

\[ C_w = C_h = C_k = C_\varepsilon = \text{WALL} \]  \hspace{1cm} (7.18)
\[ V_w = V_k = V_\varepsilon = 0 \]  \hspace{1cm} (7.19)
\[ V_h = C_p \times T_w \]  \hspace{1cm} (7.20)
8. Mesh Generation

In this work a two-dimensional mesh is being used with 18 x 29 cells in the y and z coordinate respectively. A Nonuniform grid has been used for both directions. Figures 3 and 4 shows the grid in the z direction. A finer grid is used in the blunt region, \( IZ = (7 \div 17) \), and in the zone, where the inclined wall transitions to a straight wall, \( IZ = (23 \div 26) \).

In the y coordinate, except in the boundary layer region, the grid is uniform. To obtain a finer grid resolution in the boundary layer for the laminar flow case the first five cells in the y direction from the wall obey the following proportionality relationship:

\[
BYFRAC\left(IY\right) = \left(\frac{IY}{5}\right)^3 \left(\frac{\Delta_{max}}{10GH}\right)
\]  

(8.1)

Where \( BYFRAC(IY) \) is the distance from the south side to the north side of the cell of particular interest, divided by total length of the domain, \( IY \) is the cell number, \( \Delta_{max} \) is maximum allowable cell height, and \( GH \) is the half thickness of the TVC jet vane.

A fine grid resolution for the turbulent flow case is set up in the same way as laminar flow. The only difference comes from the selection of the first five cells in y direction. The following calculation shows the difference.

From the laminar solution and the given properties the following are known:

\[
\begin{align*}
\gamma &= 885.2 \text{[m/s]} \\
\mu_{lam} &= 1 \times 10^{-5} \text{[N.s/m]} \\
\mu &= 5.5 \times 10^6 \text{[Pa]} \\
P_{static} &= 1.048 \times 10^5 \text{[Pa]} \\
\gamma &= 1.35
\end{align*}
\]
\[ \rho = 1.835 [\text{kg/m}] \]

Using the values above and the length of vane, which is 0.095m, a corresponding Reynolds number was calculated:

\[
Re_z = \frac{\rho w_Z Z}{\mu_{\text{lam}}} = \left( \frac{1.835 \times 888.5 \times 0.095}{1 \times 10^{-5}} \right) = 1.54 \times 10^6
\]

Using a power law correlation for the boundary layer thickness:

\[
\frac{\delta}{z} = 0.37 \times Re_z^{-1/5}
\]

From equation (8.2) the boundary layer thickness at the high end of the domain has been calculated as \( \delta = 2 \times 10^{-3} [\text{m}] \)

With \( Re \) based on \( w_\infty \) the velocity parallel to the wall, \( \frac{\Delta Y}{2} \) the distance from the wall to the first grid node, \( \rho_\infty \) the density, and \( \mu_{\text{lam}} \) the laminar viscosity, \( \Delta y \) must satisfy the condition

\[
Re_\Delta = \frac{\rho_\infty w_\infty \Delta y}{2 \mu_{\text{lam}}} > 132.25 \quad \text{or} \quad \Delta y > 6.48 \times 10^{-6} [\text{m}]
\]

Therefore the interval of \( \Delta y \) is chosen such that

\[ 2 \times 10^{-3} [\text{m}] > \Delta y > 6.48 \times 10^{-6} [\text{m}] \]

In this effort using the relationship

\[ \text{BYFRAC(IY)} = \left( \frac{IY}{5} \right)^2 \left( \frac{\Delta y_{\text{max}}}{10^6} \right) \]

\( \Delta y \) has been calculated as \( \Delta y = 8 \times 10^{-5} [\text{m}] \) which is in the required interval.

For both the laminar and turbulent cases, cells in the z direction were adjusted so that the points where possible physical phenomena such as shock waves and expansion fans are expected, very fine cells were used. In the other parts of the domain larger cells were used.
9. Heat Transfer Analysis

Skin friction and heat transfer quantities were calculated in both laminar and turbulent cases and they are shown in Figures (6 + 11).

9.1 Laminar Calculation

In laminar flow fluxes can be derived directly from the gradients near the wall. The first cell is close "enough" to the wall and gradients of velocity and enthalpy do not change much in this region near the wall. The shear stress and heat flux in the laminar case will be:

\[ \tau_w = \mu \frac{W_1}{\Delta Y_1/2} \]  \hspace{1cm} (7.1)

\[ q_w = \frac{\mu}{Pr} \frac{h_1 - h_w}{\Delta Y_1/2} \]  \hspace{1cm} (7.2)

The skin friction coefficient and Stanton number will be:

\[ C_f = \frac{2 \tau_w}{\rho \infty W_1^2} \]  \hspace{1cm} (9.1)

\[ S_t = \frac{q_w}{\rho \infty \mu (h_r - h_w)} \]  \hspace{1cm} (9.2)

where \( h_r \) is the recovery enthalpy

\[ \frac{h_r}{h_0} = \frac{1 + \frac{\gamma - 1}{2} \frac{M^2}{M_\infty}}{1 + \frac{\gamma - 1}{2} \frac{M^2}{M_\infty}} \]  \hspace{1cm} (9.3)

\( r \) is the recovery factor

\[ r = \sqrt{Pr} \quad \text{(laminar flow)} \]  \hspace{1cm} (9.4)
The coefficient of heat transfer in convection was calculated using

\[ h_c = \rho_\infty U_\infty C_p S_t \]  \hspace{1cm} (9.5)

9.2 Turbulent Calculations

In turbulent flow the gradients of velocity and enthalpy near the wall are very steep and change rapidly with distance from the wall.

Direct calculation of flux gradients is not accurate in this case. The log law approach is used to calculate skin friction. In the calculations using PHOENICS flow field, the following relation has been used.

\[ C_f = \frac{2 \rho_w k_w}{\omega w^2 \rho_\infty 3.33} \]  \hspace{1cm} (9.6)

To obtain equation 9.6, the turbulent kinetic energy equation has been used as a starting point. [Ref. 5],

\[ \rho \frac{Dk}{Dt} = \frac{3}{\delta y} \left( \frac{\partial}{\partial y} \left( \frac{\partial k}{\partial y} \right) \right) \]  \hspace{1cm} (9.7)

\[ + k \left[ \frac{\partial u}{\partial y} \left( \frac{\partial u}{\partial y} \right)^2 - C_D \frac{\rho^2 k}{\mu_t} \right] \]

The source term of the turbulent kinetic energy equation should be zero near the wall which means

\[ \frac{\partial u}{\partial y} \left( \frac{\partial u}{\partial y} \right)^2 - C_D \frac{\rho^2 k}{\mu_t} = 0 \]  \hspace{1cm} (9.8)

therefore the shear stress on the wall can be defined as:

\[ \tau_w = C_D \frac{1}{2} \rho_w k_w \]  \hspace{1cm} (9.9)

where \( k_w \) is the turbulent kinetic energy on the wall, \( \rho_w \) is the density on the wall and \( C_D = 0.09 \) [Ref. 5], substituting the values above into the Blasius skin friction relation the \( C_f \) equation becomes:
The heat transfer quantities are evaluated from the Chilton-Colburn form of Reynolds analogy.

\[ C_f = \frac{2 T_w}{\rho_w W^2} = \frac{\rho_w}{\rho_w} \frac{2}{\mu_w^2} \frac{k_w}{3.33} \]  \hspace{1cm} (9.10)

\[ s_t = \frac{C_f}{2} * P \rho^{-2/3} \]  \hspace{1cm} (9.11)

\[ q_w = s_t * \rho_\infty U_\infty (h_\infty - h_w) \]  \hspace{1cm} (9.12)

where equation (9.3) is used to evaluate \( h_\infty \) with the recovery factor given as:

\[ r = P \rho^{1/3} \] \hspace{1cm} (turbulent flow) \hspace{1cm} (9.13)

The convective heat transfer coefficient is calculated by using equation (9.5)
10. **Code and Computer**

PHOENICS 81, Body Fitted Coordinate (BFC) version has been used in the computations (see Ref. 3). PHOENICS has been installed on NPS IBM 3033 MVS 1.3 computer. 400 sweeps per computer run provided a reasonable convergence in all runs except the turbulent blunt case continuity error of less than $4 \times 10^{-4}$ has been achieved in the three runs.

The continuity error is the total summation of the absolute mass imbalance in all cells divided by the inlet mass flux. CPU time consumption varies from case to case as follows:

- Laminar Wedge: 630 CPU Seconds
- Turbulent Wedge: 630 CPU Seconds
- Laminar Blunt: 630 CPU Seconds
- Turbulent Blunt: 1542 CPU Seconds for 1000 sweeps
11. Results and Discussion

The results of the calculations are available on appendix c. The tabular results include the values of pressure, velocities, enthalpy, temperature mach number, density, turbulent kinetic energy and rate of turbulent dissipation. The values are given in 18 x 29 cells points.

Skin friction and heat transfer results are shown in Figures (5-11). Laminar and turbulent skin friction and Stanton number in wedge flow show improvement compared to the results reported by Yukselen (Ref. 10). The lines are smoother and the oscillations at the end were eliminated. Basically the magnitudes are similar to those in Ref. 10.

Laminar blunt values are similar except near the beginning. The beginning, as expected in blunt zone, creates higher rates of heat transfer. Even though the blunt geometry used is a multi-wedge shape it should predict the correct values except for the stagnation point itself.

Turbulent blunt skin friction has different behavior. It has a very large value at the first point and then undershoots to values that are smaller than for wedge. It should also be kept in mind that the convergence of this case wasn't very successful.
12. Conclusions and Recommendations

1. PHOENICS was found to be a friendly code for simulating complicated mixed heat transfer fluid dynamics problems.

2. Derivation of heat transfer properties to a vane solid wall in laminar and turbulent flow has been installed in the code. It can be used for predictions of heat transfer rate in both cold and hot gas flow.

3. Two features have been added to the code in NPS: The restart option and the use of initial field, make it possible to simulate time dependent processes and solve the temperature variation in the vane itself.
LIST OF REFERENCES


All Dimensions in Millimeters

Figure 2: NWC Jet Vane Approximation
Figure 3: Wedge vane domain and grid
Figure 4: Blunt vane domain and grid.
Figure 5: $R_{ex}$ No. along the Vane
Figure 6: $C_f$ in Laminar flow.
Figure 7: $C_f$ in Turbulent flow

$M_\infty = 3.2$

- △ - Wedge Vane
- □ - Blunt Vane

$C_f = 12.62 \cdot 10^{-3}$
\[ M_\infty = 3.2 \]

- \( \triangle \) - Wedge Vane
- \( \square \) - Blunt Vane

Figure 8: \( S_t \) in Laminar flow
Figure 9: Turbulent stanton number.
Figure 10: Coefficient of heat convection in laminar flow.
Figure 11: Coefficient of heat convection in turbulent flow.
Appendix A

Satellite Listing

Two subroutines SATELLITE and GROUND had to be changed and improved. The full list is enclosed in Appendix A and B.

VAN4SAT and VANTSAT are the laminar and turbulent SATELLITE subroutines, the first has the blunt geometry and the second has the wedge (it can be changed easily from wedge to blunt and vice versa). VAN4GRD and VANTGRD are exactly the same. They are the GROUND subroutines, VANTGRD is given in Appendix B.
FILE: VANTSAT FORTRAN A1

C$DIRECTIVE**SATLIT AMI LEITNER C LAMINAR SOLUTION FOR NWCS NY=18 NZ=29 YN=OTH C LECSAT CONVERTED TO DIAMSAT C FILE NAME: MODBFCST.FTN C ABSTRACT: SATELLITE MODEL MAIN PROGRAM. THIS VERSION IS C FOR USE WITH THE BODY-FITTED COORDINATE SCHEME (SUMMER 1984 C VERSION) PROVIDED AS AN ATTACHMENT TO SPRING 1983 PHOENICS. C DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983) C WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT C SUMMER 1984). C AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY C SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C CHAPTER 1 COMMON BLOCKS AND USER'S DATA.

INCLUDE (CMNGUS)
INCLUDE (CMNGRF)
COMMON/CPI/IPWRIT,IDUM(243)
COMMON/GUSSEQ
COMMON/CPI/IPWRIT,IDUM(243)
COMMON/CPI/IPWRIT,IDUM(243)
DIMENSION GDTAPE(3),DFAULT(4)
DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)
LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST
INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE
INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
SC3,C4
REAL NORTH,LOW
LOGICAL BFC
EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)
EQUIVALENCE (ARRAY3(1),SPARE1(1)),(Ml,Rl),(M2,R2)
EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,L0GIC(88))
EQUIVALENCE (L0GIC(20),BFC)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.

C$DIRECTIVE**CMNBF1$$
C THIS FILE CONTAINS SATELLITE COMMON BLOCKS FOR BFC'S
C F1 MUST BE DIMENSIONED TO GREATER THAN OR EQUAL TO
C (NX+NY+17XNZ+2«XNXXNY+6X(NX+1)X(NY+1)+6XND). THE VALUE
C OF THE DIMENSION MUST BE SET AS NBFC IN GROUP 6 OF SATLIT.
COMMON/F0B/FK5000)
COMMON/CIB/ND/CIC/KOORD
COMMON/CID/KDBGG,KDBGMF,KDBGCD,KDBIND,KDBMFX,KDBCDT,KDBPCS,
& KDBGUV,KDBGV
COMMON/CIE/KDBGS,KDBINS
COMMON/CIF/IGEN/CIG/NCART

C THE FOLLOWING ARRAYS MUST BE EXACTLY DIMENSIONED FOR NXP1,
C NYP1 AND NZP1, BUT MAY BE OVER DIMENSIONED FOR ND.
C THOSE ARRAYS MUST BE DIMENSIONED TO ALLOW FOR SETTINGS
C IN SATLIT. THEY MAY BE OVER DIMENSIONED.
COMMON/CRA/XH(19,30,1)/CRB/XE(19,30,1)
& /CRC/YS(2,30,1)/CRD/YN(2,30,1)
& /CRE/ZL(2,19,1)/CRF/ZH(2,19,1)
& /CRC/RCON/CRH/DARY/CRI/BXFRAC(99)/CRJ/BYFRAC(99)
& /CRC/BZFRAC(99)
COMMON/CLA/STORSA(6),STORHD(6),STORP,STORPE,STORPH,
& STORPH,STOR1,STOR2,STOR3,STOUNV,PRTBFC,STOCR
COMMON/CIC/STORC(6),STORHD(6),STORP,STORPE,STORPH,
& STORPH,STOR1,STOR2,STOR3,STOUNV,PRTBFC,STOCR
C END

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C GRAFFIC ARRAYS DIMENSIONED AS NEEDED...
COMMON/GRAF1/PHI1(1) /GRAF2/PHI2(1)
COMMON/GRAP/DATA1/PN1(1,1,1),PHI1(1,1,1),PN1(1,1,1),PN1(1,1,1)
DIMENSION LS31(1),ISPDA(1),ISPD(1)
DIMENSION LS31(1),ISPDA(1),ISPD(1)
C USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCE ETC. HERE.
EQUIVALENCE(RAIR,RE(21)),(GAMA,RE(22)),(GSPN,RE(23))
& (GSPN,RE(23)),(GSPN,RE(23)),(GSPN,RE(23)),(GSPN,RE(23))
C USER PLACES HIS DATA STATEMENTS HERE.
DATA NLS31,NLS31,NSR/1,1,1/ CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
**CHAPTER 2**  SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.

Please do not alter, or re-set, any of the remaining statements of this chapter.

Data:

```
DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
  0.,1.2,3,4,5,6,7 /
```

```
DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
  C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
```

```
DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
```

```
DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
```

```
DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
```

```
DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
```

```
DATA NLDATA,NIDATA,NRDATA/309,194,421/
```

```
DATA NLCREG,NTCVRG/60,350/
```

```
DATA TITPP,TITC1,TITC2,TITC3/3HRH0,4HMACH,4HTEMP,4HCFST/
```

```
CALL TAPES(10,GDTAPE,3,1,4XNRDATA)
```

C Read default file if blockdata absent.

```
IF(INTGR1(29).NE.10) GO TO 2
```

```
CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.)
```

GO TO 3

2

```
CALL DEFLT
```

```
CD  2 CALL TAPESC1,DFAULT,4,2,4XNRDATA)
```

```
CD   CALL DATAI0(RD,1)
```

```
CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED. )
```

GO TO 3

3

```
L0GIC(89)=.TR'JE.
```

**CHAPTER 3**  DEFINE DATA FOR NRUN RUNS.

C Standard section 2 ends.

C User section 2 starts.

C Group 4: Multi-runs : RUN(1-30)<T.,29X.F.>

```
RUN(1)=.FALSE.
```

C Note: All runs are deactivated at this point - user should switch on one only of runs 1-4 in next statement.

```
RUN(1)=.TRUE.
```

C Standard section 2 ends.

C User section 2 starts.

C Standard section 3 ends.

C User section 3 starts.

C All integer variables are defaulted to 0, and real variables to 0.0 unless otherwise indicated.

C The default settings of all logical variables are always indicated, e.g. variable<.T.>, or variable<.F.>.

C

C Group 1. Flow type:

```
PARA<.F.>,CARTES<.T.>,ONEPHS<.T.>
```

C Group 2. Transience:

```
STEADY<.T.>,ATIME,STEP<1>,FSTEP<1>
```

```
LAST<.E10>,FRAC(1-30)<30X1.>
```

C Service subroutine for 'NT' power-law time steps.

```
CALL G1PNR(0,NT,LAST,POWER)
```

C Group 3. X-direction:

```
NX<1>,XULAST<1.0>,XFRC(1-30)
```

C Service subroutine for power-law grid.

```
CALL G2PNR(1,NX,XULAST,POWER)
```
FILE:  VANTSAT FORTRAN A1

C--- GROUP 4. Y-DIRECTION:
C NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWRC2(NY,YVLAST,POWER)
NY=18
C-----------------------------------------------

C--- GROUP 5. Z-DIRECTION:
C NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWR(3,NZ,ZWLAST,POWER)
NZ=29
C-----------------------------------------------

C--- GROUP 6. MOVING GRID OR DISTORTED (BODY-FITTED) GRID:
C MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2M1T
C BODY-FITTED GRID
C BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
C BXFRAC(1-NX)<1.0,NXM1X0.0>
C BYFRAC(1-NY)<1.0,NYM1X0.0>
C BZFRAC(1-NZ)<1.0,NZM1X0.0>
C SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1)
C ONLY):
CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
CALL XCYIZdZ (.TRUE.)
CALL UCURVF(IZ,.FALSE.)
CALL VCURVF(IZ,.FALSE.)
CALL WCURVF(IZ,.FALSE.)
CALL PRTBF.C,.FALSE.)
CALL DARCY,BFPLOT,.FALSE.)
RTBC<.F.>,DARKY,BFPLOT<.F.>
CYCLIC BOUNDARY CONDITIONS ARE DEFAULTED INACTIVE ;
C TO ACTIVATE THEM AT SELECTED IZ SLABS USE SERVICE SUBROUTINE:
CALL XCYIZCIZ,.TRUE.)
C SERVICE SUBROUTINE TO DEACTIVATE CURVATURE TERMS IN U, V
C AND W EQUATIONS ASSOCIATED WITH CURVATURE OF IX, IY, IZ
C GRID LINES RESPECTIVELY:
CALL VCURVF(.FALSE.)
CALL WCURVF(.FALSE.)
NCART<1>
XNOTES
A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.

--- BODY-FITTED GRID ---
BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
BXFRAC(1-NX)<1.0,NXM1X0.0>
BYFRAC(1-NY)<1.0,NYM1X0.0>
BZFRAC(1-NZ)<1.0,NZM1X0.0>
SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1)
ONLY):
CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
XWARNINGSII I I I I
A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.

--- MOVING GRID ---
MGRID,IZH1,IZH2,AZH2,BZH2,CZH2,PINT,ZH2M1T
--- MOVING GRID ---
BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
BXFRAC(1-NX)<1.0,NXM1X0.0>
BYFRAC(1-NY)<1.0,NYM1X0.0>
BZFRAC(1-NZ)<1.0,NZM1X0.0>
SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1)
ONLY):
CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
XWARNINGSII I I I I
A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.

--- MOVING GRID ---
MGRID,IZH1,IZH2,AZH2,BZH2,CZH2,PINT,ZH2M1T
--- MOVING GRID ---
BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
BXFRAC(1-NX)<1.0,NXM1X0.0>
BYFRAC(1-NY)<1.0,NYM1X0.0>
BZFRAC(1-NZ)<1.0,NZM1X0.0>
SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1)
ONLY):
CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
XWARNINGSII I I I I
A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.

--- MOVING GRID ---
MGRID,IZH1,IZH2,AZH2,BZH2,CZH2,PINT,ZH2M1T
--- MOVING GRID ---
BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
BXFRAC(1-NX)<1.0,NXM1X0.0>
BYFRAC(1-NY)<1.0,NYM1X0.0>
BZFRAC(1-NZ)<1.0,NZM1X0.0>
SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1)
ONLY):
CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
XWARNINGSII I I I I
A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.

--- MOVING GRID ---
MGRID,IZH1,IZH2,AZH2,BZH2,CZH2,PINT,ZH2M1T
--- MOVING GRID ---
BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
BXFRAC(1-NX)<1.0,NXM1X0.0>
BYFRAC(1-NY)<1.0,NYM1X0.0>
BZFRAC(1-NZ)<1.0,NZM1X0.0>
SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1)
ONLY):
CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
XWARNINGSII I I I I
A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.
FILE: VANTSAT FORTRAN A1

ALL OTHER BFC DATA MUST BE SET AFTER "STANDARD BFC SECTION 2.
C) NXP1, NYP1, NZP1 STORE NX+1, NY+1, NZ+1; THESE ARE AVAILABLE TO USER AFTER STANDARD BFC SECTION 2.
D) FOR IGEN=1 USE BXFRAC, BYFRAC & BZFRAC IN PLACE OF XFRAC, YFRAC & ZFRAC.

C-~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C STANDARD BFC SECTION 1 STARTS:
C DEFAULT SETTINGS:
  NCART=10
  BFC=.TRUE.
  IGEN=1
  ND=1
  NBFC=5000
C~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C STANDARD BFC SECTION 1 ENDS:
C USER SETS BFC, IGEN, ND AND NBFC HERE:
C~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C STANDARD BFC SECTION 2 STARTS.
CALL SB4I(NXP1,NX+1,NYP1,NY+1,NZP1,NZ+1,1,0)
IF(BFC) CALL BFCDFTNBFC,XE,XW,YN,YS,ZH,ZL,ND,NXP1,NYP1,
S NZP1,NZ)
C~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C STANDARD BFC SECTION 2 ENDS.
C USER SETS ALL OTHER BFC VARIABLES HERE.
C USING NONIFORM GRID 1-8
GTH=65.E-3
GTL=150.E-3
GBETA=4.
GBETA=GBETA*3.1415927/180
GTAB=TAN(GBETA)
DELMAX=2.E-3
GNBL=5.
GPWR=2.
64 BYFRAC(IY)=(FLOAT(IY)/GNBL)XXGPWRXDELMAX/GTH
BYFRACC6)=BYFRACC5)+3.E-3/GTH
DEL=C1.-BYFRACC6))/CFLATCNY)-GNBL-1)
DO 65 IY=7,NY
65 BYFRACCIY)=BYFRACCIY-1)+DEL
BZFRAC(1)=10.E-3
DO 66 IZ=2,5
66 BZFRAC(IZ)=10.E-3+BZFRAC(IZ-1)
BZFRAC(6)=BZFRAC(5)+5.E-3
DO 67 IZ=7,9
67 BZFRAC(IZ)=BZFRAC(IZ-1)+2.E-3
DO 68 IZ=10,10
68 BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
DO 77 IZ=11,14
77 BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
DO 78 IZ=15,15
78 BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
BZFRAC(16)=BZFRAC(15)+2.E-3
BZFRAC(17)=BZFRAC(16)+3.E-3
BZFRAC(18)=BZFRAC(17)+5.E-3
DO 69 IZ=19,22
69 BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
BZFRAC(23)=BZFRAC(22)+5.E-3
BZFRAC(24)=BZFRAC(23)+2.E-3
BZFRAC(25)=BZFRAC(24)+2.E-3
BZFRAC(26)=BZFRAC(25)+5.E-3
BZFRAC(27)=BZFRAC(26)+5.E-3
DO 71 IZ=28,NZ
71 BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
DO 72 IZ=1,NZ
72 BZFRAC(IZ)=BZFRAC(IZ)/GTL
CALL DOMAIN(1,1,NX,1,NY,1,NZ)
DO 61 IX=1,NXP1
61 Z(I)=0.0
DO 62 IY=1,NYP1
62 Z(1,1)=GTL
YZ(IX,IY,1)=GTH
FILE: VANTSAT FORTRAN A1

VAN02890
VAN02900
VAN02910
VAN02920
VAN02930
VAN02940
VAN02950
VAN02960
VAN02970
VAN02980
VAN02990
VAN03000
VAN03010
VAN03020
VAN03030
VAN03040
VAN03050
VAN03060
VAN03070
VAN03080
VAN03090
VAN03100
VAN03110
VAN03120
VAN03130
VAN03140
VAN03150
VAN03160
VAN03170
VAN03180
VAN03190
VAN03200
VAN03210
VAN03220
VAN03230
VAN03240
VAN03250
VAN03260
VAN03270
VAN03280
VAN03290
VAN03300
VAN03310
VAN03320
VAN03330
VAN03340
VAN03350
VAN03360
VAN03370
VAN03380
VAN03390
VAN03400
VAN03410
VAN03420
VAN03430
VAN03440
VAN03450
VAN03460
VAN03470
VAN03480
VAN03490
VAN03500
VAN03510
VAN03520
VAN03530
VAN03540
VAN03550
VAN03560
VAN03570
VAN03580
VAN03590
VAN03600

GROUP 7. BLOCKAGE: BLOCK<F>.IP, IPWRIT

GROUP 8. DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED:

GROUP 9. VARIABLE LABELS:

GROUP 10 PROPERTIES:
FILE: VANTSAT FORTRAN A1

C IEMU1<1>, EMU1<1.0>, EMULAM<1.E-10>
C IHSAT, H2SAT, PSATEX<1.0>
C SIGMA<1.0, 2.0, 1.1, E10, 1.1, E10, 1.1, E10,
C 4<1.0, 3.14, 1.0, 1.E10, 10<1.0>
C IRH01=-1
C PT0T=55.E5
C T0T=555.55
C RAIR=287.
C GAMA=1.38
C CP=RAIR/(1-1/GAMA)
C TH=323.
C HNALL=THWC
C HT0T=CPX T0T
C RHT0T=PT0T/T0T/RAIR
C SIGMAC =1-25X1.0, 2.0, 1.1, E10, 1.1, E10,
C 4X1.0, 1.314, 1.0, 1.E10,
C BRH01=1./GAMA

C TURBULENT OR LAMINAR
C IEMU1=2
C IEMU1=1
C JEMU1=IEMU1
C EMU1=1.E-5
C EMULAM=EMU1
C GEMU1=EMU1
C GPR=.7
C SIGMAC24XGPR
C SIGMAC14X.9

C— GROUP 11 INTER-PHASE TRANSFER PROCESSES :
C ICFIP, CFIPS, IMD0T, CMD0T, CA1<1.E6>, CA2<1.E6>
C— GROUP 12 SPECIAL SOURCES :
C ISPCS0U-25), AGRAVX, AGRAVY, AGRAVZ, ABOY, HREF
C— GROUP 13 INITIAL FIELDS :
C FIINIT(1-25X25X1.E-10>
C MACH NO. OF FREE STREAM
GMACH=3.2
A=1+(GAMA-1)/2XGMACHX2
TE=T0T/A
RHE=RHT0T/A**(1/(GAMA-1))
PSTAT=PT0T/A**((GAMA/(GAMA-1))
RH01=ARH01XPSTATXBRH01
SONIC=SQRT(GAMAXRAIRXTE)
WIN=SONICXRHE
RKEIN=0.01XWINX2
EPIN=0.16XRKEINX1.5/TH/2.
FIINIT(W1)=WIN
FIINIT(P1)=PSTAT
FIINIT(H1)=HT0T
FIINIT(KE)=RKEIN
FIINIT(EP)=EPIN

C— GROUP 14 BOUNDARY/INTERNAL CONDITIONS :
C ILOOP1, ILOPN, XCYCLE<.F.>, PBAR, REGION(1-10)<10X.T.>
C XN.B. ALL 10 REGIONS ARE DEFAULTED .TRUE.. THE USER SHOULD
C SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.
C DO 14 I=1,10
C 14 REGION(I)=.FALSE.

C— GROUP 15 TO 24; REGIONS 1 TO 10
C CALL PLACE(IREGION, TYPE, IXF, IXL, IYP, IYL, IZF, IZL) &
C CALL COVAL(IREGION, VARBLE, COEFF, VALUE)
C CALL PLACE(1, LOW, 1, NX, 1, NY, 1, 1)
C CALL COVAL(1, M1, FIXFLU, WINXRHE)
C CALL COVAL(1, M1, E20X1.E-20XRHE)
C GMEX=2XGAMA/WIN/(GAMA-1)
C GVM=PT0TXRHE/RHT0T
C CALL COVAL(1, M1, GVM, GMEX)
C CALL COVAL(1, M1, ONLYMS, WIN)
CALL COVAL(1,H1,ONLYMS,HTOT)
CALL COVAL(1,KE,ONLYMS,KEIN)
CALL COVAL(1,EP,ONLYMS,EPIN)
CALL PLACE(2,HIGH,1,NX,1,NY,NZ,NZ)
CALL COVAL(2,M1,FIXVAL,PSTATX0.)
CALL COVAL(2,M1,1000*HEX/RH/PSTAT,PSTAT)
CALL COVAL(2,H1,ONLYMS,HTOT)
CALL COVAL(2,H1,ONLYMS,HTOT)
CALL COVAL(2,M1,1000*X/RH/PSTAT,PSTAT)
CALL COVAL(2,H1,ONLYMS,HTOT)
WALL ALONG THE VANE IZ(11,NZ)
GCM=EMUI/C.5*BYFRAC(1)*XGTH)
DY1=BYFRAC(1)*XGTH
G0EFF=EMUI/(0.5*DY1)
G0EFF=EMUI/(0.5*DY1*SIGMA(2))
CALL PLACE(3,SOUTH,1,NX,1,1,12,NZ)
CALL COVAL(3,W1,GOEFF,0.)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,W1,WALL,0.)
CALL COVAL(3,H1,WALL,HWALL)
CALL COVAL(3,KE,WALL,0.)
CALL COVAL(3,EP,WALL,0.)
GROUP 25 GROUND STATION
GROSTA=.TRUE., NAMLST<.F.>
* NAMLST ACTIVATES NAMELIST IN GROUND.
GROSTA=.TRUE.
GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS
WHOLEP<.F.>, SUBPST<.F.>, DONACC<.F.>
WHOLEP=.TRUE.
GROUP 27 SWEEP AND ITERATION NUMBERS
FSWEEP=1, LSWEEP=1, LITHE<1>, LITKE<1>, LITH<1>,
LITER(1-25)<9.X1,-1.15X1>
LITER(1-25)<9.X1,-1.15X1>
LITER(1-25)<9.X1,-1.15X1>
GROUP 28 TERMINATION CRITERIA
ENDIT(1-25)<9.X1,E-10, 0.5, 15X1.E-10>
GROUP 29 RELAXATION
RLXP=1.0, DTFALS<3-25<23X1.E10>
DTFALS(H1)=1.0=E-5
DTFALS(V1)=1.0=E-5
DTFALS(KE)=1.0=E-5
DTFALS(EP)=1.0=E-6
RLXP=.3
GROUP 30 LIMITS
VELMAX<1.E10>, VELMIN<1.E10>, RHOMAX<1.E10>, RHOMIN<1.E-10>,
TKEV<1.E10>, TKEV<1.E-10>, EMUMAX<1.E10>, EMUMIN<1.E-10>,
EPSMAX<1.E10>, EPSMIN<1.E-10>, AMDTMX<1.E10>, AMDTMN<1.E-10>,
EPSMAX=1.0E10
GROUP 31 SLOWING DEVICES: SLORHO<1>, SLORHU<1>
GROUP 32 PRINT-OUT OF VARIABLES
PRINT(1-25)<.T.>,F., 23X.T.>, SUBWGR<.F.>
PRINT(C1)=.TRUE.
PRINT(C2)=.TRUE.
FILE: VANTSAT FORTRAN A1

PRINT(C3)=.TRUE.
PRINT(PP)=.TRUE.

C----- GROUP 33 MONITOR PRINT-OUT :
C  IXMON<1>, IYMON<1>, IZMON<1>, NPRMON<1>, NPMNT<1>
NPRMON=10
IYMON=2
IZMON=12

C----- GROUP 34 FIELD PRINT-OUT CONTROL :
C  NPRINT<100>, NTPRIN<100>, NXPRIN<1>, NYPRIN<1>, NZPRIN<1>,
  IZPRF<1>, ISTPRF<1>, ISTPR2<1>, IZPRL<1000>, ISTPRL<10000>
NPRMON<100>, KOUTPT
NPRINT=LSWEEP

C----- GROUP 35 TABLE CONTROL :
C  TABLES<.F.>, NTABLE, NTABVR, LINTAB, NPRTAB, NMON,
  ITAB(1-8), MTABVR(1-8)

C----- GROUP 36 ARE NOT DOCUMENTED IN THE INSTRUCTION
C  MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY

C----- GROUP 37 DEBUG PRINT-OUT SLAB AND TIME-STEP :
C  IXPRL<1>, IYPRL<1>, ISTPRL<1>, ISTPR2<1>

C----- GROUP 38 DEBUG SHEEP AND SUBROUTINES :
C  KEMU, KMAIN, KINDEX, KGEOM, KINPUT, KSOAT, KCMPF, KSOINE, KSOVR,
  KSOVL1, KSOVL2, KSOVL3, KSOCP, KADST, KLFLUX, KSHIFT, KDF, KCOMPU,
  KCOMPV, KCOMPW, KCOMPR, KHALF, KDBRHO<1>, KDBEXP, KDBMDT

C----- GROUP 39 MONITOR, TEST, AND FLAG :
C  MONITR<.F.>, FLAG<.F.>, TEST<.T.>, KFLAG<1>

C----- END OF MAINTENANCE-ONLY SECTION

C----- GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
C  IERRP<1000>, RESREFN(1,3-24)<25X1.>, RESMAP<.F.>,
  RESIDN(1-25)<2X.F.>, 23X.*T.>, KOUTPT
  RESREF(1)=WINXRHE
  RESREF(7)=WINXRESREF(1)
  RESREF(5)=WINXRESREFN(1)*0.1
  RESREF(H1)=HTOTXRESREF(1)
  RESREF(K1)=KEINXRESREF(1)
  RESREF(E1)=EPINXRESREF(1)
  IERRP=LSWEEP/20
  KOUTPT=LSWEEP/20

C----- GROUP 40 SPECIAL DATA : LOGIC(1..10), INTGR(1..10), RE(21..30),
C  NLSP<1>, NISP<1>, NRSP<1>, SPDATN<.F.>, LSFPDA(1), ISFPDA(1), RPSPA(1)
C  USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C  TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C  SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
C  PROVISION SET SPDATN = .T., AND DIMENSION ARRAYS LSFPDA,
C  RPSPA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE

C----- GROUP 41 RESTARTS AND DUMPS :
C  SAVEM<.F.>, RESTRT<.F.>, KINPUT
C  SAVEM=.TRUE.
C  RESTRT=.TRUE.

C----- GROUP 43 GRAPHIC :
C  GRAPHS<.F.>, ORTHOG<.T.>, ANTSYM, NPRIT<1>, ITITL<5X4X***>
C  FOR A GRAPHIC RUN, DIMENSION PHI & PHI2 AS FOLLOWS:
C  PHI(NX*NY*NZ*NM)
C  PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)), WHERE
C  NM=NO. OF VARIABLES STORED & DENSITY<IES>
C  IBLK=0 IF BLOCK=.FALSE.,=4 IF A 3D RUN,
C =3 IF A 2D.YZ RUN,
C  IBLK=100 IF IBLK<3, IN ALL RUNS

--- IF(IRUN.EQ.1) GO TO 900
900 CONTINUE
C----- ALL RUNS
C(----- WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC... ----C)

IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)
IF(BFC) CALL WRTBFC(14,NBFC,XE,XH,YN,YS,ZH,ZL,ND,NX+1,NY+1,NZ+1,NZ,PRTBFC)

OLD PRACTICES RETAINED FOR REFERENCE:

IF(SPDATA) CALL SPCDAT(IRUN)
IF(BLOCK) CALL PORDAT(IRUN)
IF(GRAPHS) CALL SORT(IRUN)
IF(RESTRT) GO TO 902

DO 901 INDVAR=1,25
IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901
CALL FLDDAT(IRUN)
GO TO 902

901 CONTINUE

902 CALL DATAIO(WRT,10)
IF(MONITR) CALL DATAIO(WRT,-6)

999 CONTINUE
STOP
END

C*** IGEN=1 SO BFCXYZ NOT REQUIRED.
C*** COMMENT OUT BOTH VERSIONS.

SUBROUTINE BFCXYZ(NXP1,NYP1,NZP1)
RETURN
END
FILE: VAN4SAT FORTRAN A1

C+DIRECTIVE+SATLIT AMI LEITNER
C LAMINAR SOLUTION FOR NWCS NY=18 NZ=29 YN=GTH
C LECAT CONVERTED TO DIAMSAT
C FILE NAME: MOBFCST.FTN
C
C ABSTRACT: SATELLITE MODEL MAIN PROGRAM. THIS VERSION IS
C FOR USE WITH THE BODY-FITTED COORDINATE SCHEME (SUMMER 1984
C VERSION) PROVIDED AS AN ATTACHMENT TO SPRING 1983 PHOENICS.
C DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C (SUMMER 1984).
C
C AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY
C SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C
C 1 ENDS.

C INCLUDE (CMNGUS)
C INCLUDE (CMNGRP)
C COMMON/CPI/IPHRIT, IUM(243)
C DIMENSION DOTAPE(3), DFAULT(4)
C DIMENSION ARRAY1(309), ARRAY2(194), ARRAY3(421)
C LOGICAL ARRAY1.LSPDA, WRT, RD, NAMLST
C INTEGER ARRAY2, XPLANE, YPLANE, ZPLANE
C INTEGER P1, PP, U1, U2, V1, V2, R1, R2, RS, EP, H1, H2, H3, C1, C2,
C & C3, C4
C REAL NORTH, LOW
C LOGICAL BFC
C EQUIVALENCE (ARRAY1(I), CARTES), (ARRAY2(I), NX)
C EQUIVALENCE (ARRAY3(I), SPARE1(I), (M1, R1), (M2, R2)
C EQUIVALENCE (LSTRUN, INTEGR(12)), (NAMLST, LOGIC(88))
C EQUIVALENCE (LOGIC(20), BFC)
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.

C   END

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:

C USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.

C EQUIVALENCE (RAIR, RE(21)), (GAMA, RE(22)), (GSWP, RE(23))
C &(GPR, RE(24)), (TW, RE(25)), (GEMU1, RE(26)), (JEMU1, INTEGR(1))

C USER PLACES HIS DATA STATEMENTS HERE.

DATA NLSP, NISP, NRSP/1, 1, 1/
C
C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
FILE: VAN4SAT FORTRAN A1

C------------------------------------------------------------------------
CHAPTER 2  SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
C  PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C  STATEMENTS OF THIS CHAPTER.
DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
& 0.,1.,2.,3.,4.,5.,6.,7. /
DATA R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,
& R11,R12,R13,R14,R15,R16,R17,R18,R19,R20/
DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA.4HULT.,4HDTA/,1HG/
DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
DATA NLDATA,NIDATA,NRDATA/309,194,421/
DATA NLCREG,NTCVRG/60,350/
DATA TITPP,TITC1,TITC2,TITC3/3HRH0,4HMACH,4HTEMP,4HCFST/
CALL TAPESC10,GDTAPE,3,1,4*NRDATA)
RE
AD DEFAULT FILE IF BLOCKDATA ABSENT
IF(INTGR1(29).NE.10) GO TO 2
CALL WRIT40C40HDATA ESTABLISHED IN BLOCK DATA.
GO TO 3
2 CALL DEFLT
2
CALL TAPES(1,DFAULT,4,2,4XNRDATA)
CALL WRIT40C40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C
3
CALL WRIT40C40HFILE MODSTL.FTN IS THE SATLIT USED.
LOGIC(89)=.TRUE.
C------------------------------------------------------------------------
CHAPTER 3   DEFINE DATA FOR NRUN RUNS.
C  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
C  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS.
C— GROUP  41MULTI-RUNS i RUN(1-30)<.T.,29*. F.>
RUN(1)=.FALSE.
C  NOTE: ALL RUNS ARE DEACTIVATED AT THIS POINT - USER SHOULD
C  ===  SWITCH ON ONE ONLY OF RUNS 1-4 IN NEXT STATEMENT.
RUN(1)=.TRUE.
C  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
C  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS.
DO 10 IRUN=1,30
IF(.NOT.RUN(IRUN)) GO TO 10
NRUN=NRUN+1
LSTRUN=IRUN
10 CONTINUE
DO 999 IRUN=1,LSTRUN
IF(.NOT.RUN(IRUN)) GO TO 999
INTGR(11) = IRUN
C  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
C  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS.
C— ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C  TO 0.0, UNLESS OTHERWISE INDICATED.
C  E.G. BY VARIABLE<10.>, OR <10.0> AS APPROPRIATE.
C  THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C  INDICATED, E.G. VARIABLE<.T.>, OR VARIABLE<.F.>.
C—
C  RUN1
C—
C— GROUP 1. FLOW TYPE:
C  PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
C—
C— GROUP 2. TRANSIENCE:
C  STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C  TLAST<1.E10>,TFRACT<30>30W1>
C  SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
C  CALL GRDPWRC1(NX,XULAST,POWER)
C—
C— GROUP 3. X-DIRECTION:
C  NX<1>,XULAST<1.0>,XFRACT<1-30>
C  SERVICE SUBROUTINE FOR POWER-LAW GRID:
C  CALL GRDPWR(1,NX,XULAST,POWER)
C—
GROUP 4. Y-DIRECTION:
NY<1>, YVLAST<1.0>, YFRAC(1-30), RINNER, SNALFA
CALL GRDPHR(2, NY, YVLAST, POWER)
NY<18

GROUP 5. Z-DIRECTION:
NZ<1>, ZHLAST<1.0>, ZFRAC(1-30)
SERVICE SUBROUTINE FOR POWER-LAW GRID:
CALL GRDPWR(3, NZ, ZHLAST, POWER)
NZ<29

GROUP 6. MOVING GRID OR DISTORTED (BODY-FITTED) GRID:
--- MOVING GRID ---
MGRID, IZW1, IZW2, AZW2, BZW2, CZH2, PINT, ZH2MIT
--- BODY-FITTED GRID ---
BFC<.T.>, IGEN<1>, ND<1>, NBFC<5000>, KOORD, RCON
BFRAC(1-NX)<1.0, NXMI<1.0>
BYFRAC(1-NY)<1.0, NYMI<1.0>
BZFRAC(1-NZ)<1.0, NZMI<1.0>
SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1 ONLY):
CALL DOMAIN(ID, IIS, IISL, IISF, IISLF, ISTORU, ISTORH, ISTORPN, ISTORPE, ISTORPH, ISTRU, Проблема с выводом), ISTRU, Проблема с выводом)
CALL UCURVE(IZ, FALSE.)
CALL VCURVE(IZ, FALSE.)
CALL WCURVE(IZ, FALSE.)
NCART<1>

**warnings:**
A) WHEN USING BFCs STOVAR(H3), STOVAR(C4), STOVAR(21) ARE AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY COMPONENTS.
B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C) MOVING GRID, TWO-PHASE AND PARABOLIC OPTIONS ARE NOT AVAILABLE WITH BFC OPTION.
D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.

**Notes:**
---
A) THE STANDARD VELOCITY-FIELD PRINTOUT FOR THE VELOCITY RESOLUTES IS ACTIVATED IN THE USUAL WAY. AN ADDITIONAL OPTION EXISTS FOR PRINTING THE CARTESIAN VELOCITY-COMPONENTS WHICH MAY BE ACTIVATED BY SETTING THE FOLLOWING LOGICALS:
STOVAR(U2)=.T. FOR U-COMPONENT (CARTESIAN)
STOVAR(V2)=.T. FOR V-COMPONENT (CARTESIAN)
STOVAR(W2)=.T. FOR W-COMPONENT (CARTESIAN)
SIMILARLY PRINTOUT OF NON-ORTHOGONAL VELOCITY COMPONENTS MAY BE ACTIVATED AS FOLLOWS:
STOVAR(C4)=.T. FOR U-COMPONENT (NON-ORTHOG)
STOVAR(H3)=.T. FOR V-COMPONENT (NON-ORTHOG)
STOVAR(21)=.T. FOR W-COMPONENT (NON-ORTHOG)
B) BFC (TO ACTIVATE THE BFC OPTION), IGEN (THE CODE FOR METHOD OF GRID SPECIFICATION), ND (NUMBER OF SUB-DOMAINS) AND NBFC (THE FI ARRAY DIMENSION), MUST BE SET BEFORE "STANDARD BFC SECTION 2".

---
FILE: VAN4SAT FORTRAN A1

C  ALL OTHER BFC DATA MUST BE SET AFTER *STANDARD BFC
C  SECTION 2.  ==
C  C) NXP1, NYP1, NZP1 STORE NX+1, NY+1, NZ+1; THESE ARE
C      AVAILABLE TO USER AFTER STANDARD BFC SECTION 2.
C  D) FOR IGEN=1 USE BXFRAC,BYFRAC & BZFRAC IN PLACE OF
C     XFRACYFRAC & ZFRAC.
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 STARTS.
C DEFAULT SETTINGS:
NCART=10
BFC=.TRUE.
IGEN=1
NB=1
NBFC=5000
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 ENDS.
C   XUSER SETS BFC, IGEN, ND AND NBFC HERE.
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 STARTS.
CALL SB4I(NXP1,NX+1,NYP1,NY+1,NZP1,NZ+1,1,0)
IF(BFC) CALL BFCDFT(NBFC,XE,XW,YN,YS,ZH,ZL,ND,NXP1,NYP1,
& NZP1,NZ)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 ENDS.
C   XUSER SETS ALL OTHER BFC VARIABLES HERE.
C   XUSING NONIFORM GRID 1-8
GTH=65.E-3
GTL=150.E-3
GBETA=4.
GBETA=GBETA*3.1415927/180
GTAB=TAN(GBETA)
DELMAX=2.E-3
GNBL=5.
GPWR=4.
DO
64
IY=1,5
64  BYFRAC(IY)=(FLOAT(IY)/GNBL)*GPWR*DELMAX/GTH
BYFRAC(6)=BYFRAC(5)+3.E-3/GTH
DEL=(1.-BYFRAC(6))/(FLOAT(NY)-GNBL-1)
DO 65 IY=7,NY
65  BYFRAC(IY)=BYFRAC(IY-1)+DEL
BZFRAC(1)=10.E-3
DO 66 IZ=2,5
66  BZFRAC(IZ)=10.E-3+BZFRAC(IZ-1)
BZFRAC(6)=BZFRAC(5)+.E-3
DO 67 IZ=7,9
67  BZFRAC(IZ)=BZFRAC(IZ-1)+2.E-3
DO 68 IZ=10,11
68  BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
BZFRAC(12)=BZFRAC(11)+.E-3
DO 70 IZ=13,14
70  BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
DO 71 IZ=15,15
71  BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
BZFRAC(16)=BZFRAC(15)+.E-3
BZFRAC(17)=BZFRAC(16)+2.E-3
BZFRAC(18)=BZFRAC(17)+.E-3
DO 69 IZ=19,22
69  BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
BZFRAC(23)=BZFRAC(22)+3.E-3
BZFRAC(24)=BZFRAC(23)+.E-3
BZFRAC(25)=BZFRAC(24)+2.E-3
BZFRAC(26)=BZFRAC(25)+3.E-3
BZFRAC(27)=BZFRAC(26)+5.E-3
DO 72 IZ=28,NZ
72  BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
DO 73 IZ=1,NZ
73  BZFRAC(IZ)=BZFRAC(IZ)/GTL
CALL DOMAIN(1,1,NX,1,NY,1,NZ)
DO 61 IX=1,NXP1
DO 62 IY=1,NYP1
ZL(IX,IY,1)=GTL
DO 63 IZ=1,NZP1
51
 FILE: VAN4SAT FORTRAN A1

YN(IX,IZ,1) = GTH
63 YS(IX,IZ,1) = 0.0
C YS(IX,13,1) SHOULD COME AFTER
DO 662 IZ = 15, 25
C BL DO 662 IZ = 16, 25
662 YS(IX,IZ,1) = (BZFRAC(IZ-1) - BZFRAC(3)) * GTB * GTL
C BL DO 663 IZ = 13, 15
C BL GZ12 = (BZFRAC(IZ-1) - BZFRAC(11)) * GTL - 0.5E-3
C BL 663 YS(IX,IZ,1) = SQRT(YS(IX,16,1) * GZ12 * 2 - GZ12 ** 2)
DO 664 IZ = 26, NZ
664 YS(IX,IZ,1) = YS(IX,25,1)
CONTINUE
STORSA(IFIX(LOW)) = .TRUE.
STORSA(IFIX(HIGH)) = .TRUE.
STORSA(IFIX(SOUTH)) = .TRUE.
STORWD(IFIX(SOUTH)) = .TRUE.
STORP = .TRUE.
PRTBFC = .TRUE.
CDAR
C

C—— GROUP 7. BLOCKAGE: BLOCK<.F.>, IPLANE, IPWRIT
XSET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
CALL CONPOR(IR, TYPE, VALUE, IXF, IXL, IYF, IYL, IZF, IZL), WHERE:
IR = RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION; 'TYPE' = EAST,
WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE' = WANTED POROSITY
OVER REGION IXF,...IZL.
XDIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
PC(NX,NY,NZ) ABOVE.
XFOR FULLY-BLOCKED
CELLS 'VALUE' = 0.0) USER NEED SET ONLY
THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
AUTOMATICALLY ZERODED.
XFOR SATETILITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
XFOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
IN THIS CASE, THE USER MUST SET ALL ELEMENTS OF
ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
CALL CR(PARRAY, VALUE, IXF, IXL, IYF, IYL, IZF, IZL, NX, NY, NZ)
ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
XCONPOR MUST NOT BE USED IN CONJUNCTION WITH EXPLICIT
SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).

C—— GROUP 8. DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
SOLVAR(1-25)<25X.F.>, ST0VAR(1-25)<25X.F.>, CONC(1-9)<6X.T.>
USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
P1, PP, U1, U2, V1, V2, W1, W2, RS, KE, EP, H1, H2, H3, C1, C2, C3, C4.
SOLVAR(P1)=.TRUE.
SOLVAR(PP)=.TRUE.
SOLVAR(V1)=.TRUE.
SOLVAR(W1)=.TRUE.
SOLVAR(H1)=.TRUE.
SOLVAR(KE)=.TRUE.
SOLVAR(Ep)=.TRUE.
SOLVAR(V2)=.TRUE.
ST0VAR(W2)=.TRUE.
ST0VAR(C1)=.TRUE.
ST0VAR(C2)=.TRUE.
ST0VAR(C3)=.TRUE.

C—— GROUP 9. VARIABLE LABELS :
TITLE(1-25X2HP1, 2HPP, 2HU1, 2HU2, 2HV1, 2HV2, 2HW1, 2HW2, 2HR1,
2HR2, 2HRS, 2HKE, 2HEP, 2H1, 2H2, 2H3, 2MC1, 2MC2,
2MC3, 2HC4, 2HRX, 2HRY, 2HRZ, 2XHXXXX>
TITLE(C1)= TITC1
TITLE(C2)= TITC2
TITLE(C3)= TITC3
TITLE(PP)= TITPP

C—— GROUP 10 PROPERTIES:
IRH01<1>, IRH02<1>, RH01<1.0>, RH02<1.0>,
CDAR
C

C ——— FILE: VAN4SAT FORTRAN A1

52
FILE: VANSAT FORTRAN A1

C ARH01<1.0>,BRH01<1.0>,CRH01<1.0>
C IEMU1<1>,EMU1<1.0>,EMULAM<1.E-10>
C SIGMA(1-25)<1.0,2.0,1.,1.E10,1.,1.E10,1.,1.E10,
C 4X1.0,1.314,1.0,1.E10,1.E10,1.,1.E10>
C IRH01=-1
PT0T=55.E5
T0T=55.55
RAIR=287.
GAMA=1.35
CP=RAIR/(1-1/GAMA)
TW=323.
HWALL=TW*CP
HTOT=CP*TOT/RAIR
LOGIC(87)=.TRUE.
ARH01=RHT0T/PT0T*GAMA
C TURBULENT OR LAMINAR
IEMU1=-1
C     IEMU1=1
JEMU1=IEMU1
EMU1=1.E-5
EMULAM=EMU1
GEMU1=EMU1
GPR=.7
SIGMA(24)=GPR
SIGMA(14)=GPR
C-- GROUP 11 INTER-PHASE TRANSFER PROCESSES :  
C ICFIP,CFIPS,IMD0T,CMDOF,CA1<1.E6>,CA2<1.E6>
C-- GROUP 12 SPECIAL SOURCES :  
C ISPCSOC1-25),AGRAVX,AGRAVY,AGRAVZ,ABUOY,HREF
C-- GROUP 13 INITIAL FIELDS :  
C FINIT(1-25)<25X1.E-10>
C MACH NO. OF FREE STREAM
GMACH=3.2
A=1+(GAMA-1)/2*GMACH**2
TE=TOT/A
RHE=RHTOT/A**(1/(GAMA-1))
PSTAT=PTOT/A**(GAMA/(GAMA-1))
RH01=ARH01*PSTAT*BRH01
SONIC=SQRT(GAMA*RAIR*TE)
WIN=SONIC*GMACH
RKEIN=0.1*WIN*2
EPIN=0.16*RKEIN**1.5/GTH/2.
FINIT(WI)=WIN
FINIT(P1)=PSTAT
FINIT(HI)=HTOT
FINIT(KE)=RKEIN
FINIT(EP)=EPIN
C-- GROUP 14 BOUNDARY/INTERNAL CONDITIONS :
C ILOOP1,ILOOPN,XCYCLE=.FALSE.,PBAR,REGION(1-10)<10*X.T.>
C *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE.. THE USER SHOULD
C SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.
DO 14 I=1,10
14 REGION(I)=.FALSE.
C-- GROUP 15 TO 24: REGIONS 1 TO 10
C ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE
C USER, PREFERABLY BY WAY OF:
C CALL PLACE(IREGON,TYPE,IXF,IYL,IZF) &
C CALL COVAL(IREGON,VARBLE,COEFF,VALUE)
C CALL PLACE(1,LW1,1,NX,1,NY,1,1)
C CALL COVAL(1,M1,FMXFLU,WIN*RHE)
CDAR CALL COVAL(1,M1,1.E-20,1.E+20*WIN*RHE)
GCM=2*GAMA/(GAMA-1)
GVM=PTOT*RHE/RHTOT
CALL COVAL(1,M1,GCM,GVM)
CALL COVAL(1,H1,ONLYMS,WIN)
CALL COVAL(1,H1,ONLYMS,HTOT)
CALL COVAL(1,KE,ONLYMS,RKEIN)
CALL COVAL(1,EP,ONLYMS,EPIN)
CALL PLACE(2,HIGH,1,NX,1,NY,NZ,NZ)
CALL COVAL(2,H1,ONLYMS,HTOT)
CALL COVAL(2,H1,ONLYMS,HTOT)
CALL COVAL(2,H1,ONLYMS,HTOT)
CALL COVAL(2,H1,ONLYMS,HTOT)
C WALL ALONG THE VANE IZ(I1,NZ)
GCM=EMUL/(0.5*BYFRAC(1)*GTH)
DYL=BYFRAC(1)*GTH
GEOFF=EMUL/(0.5*DYL*SIGMA(24))
GEOFH=EMUL/(0.5*DYL*SIGMA(24))
CALL PLACE(3,SOUTH,1,NX,1,1,4,NZ)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWALL)
CALL COVAL(3,H1,GOEFF,HWAL
FILE: VAN4SAT FORTRAN A1

C---- GROUP 33 MONITOR PRINT-OUT
C IXMON<1>, IYMON<1>, IZMON<1>, NPRINTM<1>, NPRINTN<1>
NPRMON=5
IYMON=2
IZMON=12
C----
C---- GROUP 34 FIELD PRINT-OUT CONTROL
C NPRINT<100>, NPRINT1<100>, NPRINT2<1>, NPRINT3<100>, NPRINT4<1000>
C IZPRF<1>, ISTPRF<1>, IZPRL<10000>, ISTPRL<10000>
C NUMCLS<10>, KOUTPT
NPRINT=LSWEEP
C----
C---- GROUP 35 TABLE CONTROL
C TABLES<.F.>, NTABLE, NTABLEVR, NTABVR, NMON,
ITAB<1-8>, NTABVR<1-8>
C----
C---- GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
C AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
C----
C---- GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP
C IZPRK1<1>, IZPRK2<1>, ISTPR1<1>, ISTPR2<1>
C----
C---- GROUP 37 DEBUG SWEEP AND SUBROUTINES:
C KEMU, KMAIN, KINDEX, KGEOM, KINPUT, KSOQAT, KCOMPP, KSORCE,
KSOVL1, KSOVL2, KSOVL3, KCOMPP, KADJST, KFLUX, KSHFT, KDIFF,
KCOMPP, KCOMPV, KCOMPW, KCOMPR, KALL, KDBRHO<1>, KDBEXP, KDBMDT
C----
C---- GROUP 38 MONITOR, TEST, AND FLAG
C MONITR<.F.>, FLAG<.F.>, TEST<.T.>, KFLAG<1>
C----
C---- END OF MAINTENANCE-ONLY SECTION
C----
C---- GROUP 39 ERROR AND RESIDUAL PRINT-OUT
C IERRP<100>, RESREF<1, 3-24<25*1.>, RESMAP<.F.>,
RESID<1-25><25*1, KOUTPT
RESREF(1)=WINKRHE
RESREF(7)=WINXRESREF(1)
RESREF(5)=WINXRESREF<1><0.1
RESREF<1)=HTOT*RESREF(1)
RESREF<1)=RKEINXRESREF<1>
IERRP=LSWEEP/10
KOUTPT=LSWEEP/10
C----
C---- GROUP 40 SPECIAL DATA
C LOGICC<1-10>, INTGR<1-10>, RE<21. . 30>,
NLSP<1>, NISP<1>, NRSP<1>, SPDATA<.F.>, LSPDA<1>, ISPDA<1>, RSPDA<1>
C USE FIRST 10 ELEMENTS OF ARRAYS LOGIC S INTGR AND 21ST
TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA
FROM SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C----
C---- GROUP 41 RESTARTS AND DUMPS
C SAVEM<.F.>, RESTART<.F.>, KINPUT
SAVEM=.TRUE.
BFPLOT=.TRUE.
RESTART=.TRUE.
C----
C---- GROUP 42 GRAPHIC
C IGRAPHS<.F.>, ORTHOG<T.>, ANTSYM, NPRINT<1>, ITITL<5*4H****>
C FOR A GRAPHIC RUN, DIMENSION PH1 & PH1 AS FOLLOWS:
PH1(NXNNY*NZNM)
PH12((NX+2)*(NY+2)*(NM+2)*(NM+IBLK)), WHERE
NM=NO. OF VARIABLES STORED + DENSITY<IES>
IBLK=0 IF BLOCK=.FALSE., =4 IF A 3D RUN,
=5 IF A 2D, YZ RUN.
C----
C---- IF(TRUN.EQ.1) GO TO 900
900 CONTINUE
C ALL RUNS
CXXXAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBB USER SECTION 3 ENDS.
CXXXAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBB STANDARD SECTION 4 STARTS;
C----
C WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...
C
IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
IF(BLOCK) CALL WRTBLOCK(PBPN,PHPC,NX,NY,NZ,IPLANE)
IF(BFC) CALL WRTBFC(14,NBFC,XE,XW,YN,YS,ZH,ZL,
&ND,NX+1,NY+1,NZ+1,NZ,PRTBFC)
C OLD PRACTICES RETAINED FOR REFERENCE:
C        IF(SPDATA) CALL SPCDAT(IRUN)
C        IF(BLOCK) CALL PORDAT(IRUN)
IF(GRAPHS) CALL SORT(IRUN)
IF(RESTRT) GO TO 902
DO 901 INDVAR=1,25
IF(IFIX(FINIT(INDVAR)+0.1).NE.10101) GO TO 901
CALL FLDDAT(IRUN)
GO TO 902
901 CONTINUE
902 CALL DATAIO(WRT,10)
IF(MONITR) CALL DATAIO(WRT,-6)
999 CONTINUE
STOP
END  
Cxes   IGEN=1 SO BFCXYZ NOT REQUIRED.
Cxes  COMMENT OUT BOTH VERSIONS.
C-----------------------------------------------
SUBROUTINE BFCXYZ (NXP1,NYP1,NZP1)
RETURN
END
Appendix B

Ground Listing
FILE: VANTGRD FORTRAN A1

C$DIRECTIVE$**$MAIN* AMI LEITNER
C FILE LAST GEO. NZ=27 NY=18 LAMINAR FLOW
C INCLUDE DED SUBROUTINES: THE MODELS OF MAIN, GROUND & STRIDE.
C DOCUMENTATION; PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C (SUMMER 1984).
C $Satellite file name: MODBFCGD.FTN
C xINCLUDE DED SUBROUTINES, THE MODELS OF MAIN, GROUND & STRIDE.
C DOCUMENTATION. PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C (SUMMER 1984).
C $Satellite file name: MODSTL.FTN

C0MM0N/ISHIFT/IIK57), NFMAX
C SET F-ARRAY DIMENSION AS NEEDED, & SET NFMAX ACCORDINGLY.
C FOR BFC'S ALSO SET F-ARRAY DIMENSION AS NEEDED, AND SET
C NF1MAX ACCORDINGLY.

C COMMON/FOB/FOB(0),
C COMMON/NFOB/NF1MAX
C COMMON F(25000)
C NFMAX=25000
C NF1MAX=10000
C CALL MAIN1
C STOP
C END

C$DIRECTIVE$**$GROUND
C SUBROUTINE GROUND, ICHAP, ISTP, ISWP, IZED, INDVAR.
C INCLUDE (CMNGUS)
C INCLUDE (GUSSEQ)
C INCLUDE NMLIST
C LOGICAL BFC
C EQUIVALENCE (L0GIC(20),BFC)
C
C+++++MEANING OF SUBROUTINE ARGUMENTS
C IRN=RUN NUMBER; ICHAP=CHAPTER CALLED; ISTP=TIME STEP;
C ISWP=SOLUTION SWEEP; IZED=Z-SLAB; INDVAR SEE CHAPTERS BELOW.
C+++++USER-INTRODUCED VARIABLES & ARRAYS
C TO AVOID CONFLICT WITH VARIABLE NAMES USED IN COMMON, ALL
C VARIABLES INTRODUCED BY THE USER SHOULD HAVE NAMES STARTING
C WITH 'G' IF REAL, 'J' IF INTEGER, AND 'G' OR 'J' IF LOGICAL.
C THUS GDZ(IZ) MIGHT BE A Z-INTERVAL ARRAY;
C GWIKY.IX) A 2-D ARRAY FOR AXIAL VELOCITY; ETC.
C USER-GENERATED SUBROUTINES SHOULD BE NAMED CORRESPONDINGLY, EG
C SUBROUTINE GVISC(GTEMP,GCNC,GVSC), FOR COMPUTING VISCOSITY
C+++++GROUND-TO-EARTH CONNECTING SUBROUTINES:
C XUSE GET(NAME,GARRAY,NY,NX) TO PUT VALUES OF VARIABLE NAMED
C 'NAME' INTO ARRAY 'GARRAY' DIMENSIONED GARRAY(NY,NX).
C XUSE SET(NAME,IXF,IXL,IYF,IYL,GARRAY,NY,NX) TO SET VARIABLE
C 'NAME' TO GARRAY(IY,IX) OVER THE REGION: IXF-IXL & IYF-IYL.
C XUSE PRNSLB(NAME) TO PRINT VARIABLE 'NAME' OVER X-Y PLANE.
C XUSE ADD(NAME,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C TO ADD SOURCE TO VARIABLE NAMED 'NAME' (SEE CHAPTER 5).
C XUSE READIZ(IZED) IN CHAPTERS 1, 2, 8, 9 TO ACCESS PI,...DM
C & VOL,...AHDZ. (SEE FOOTNOTE TO LEGALITY TABLE)
C XUSE GET1D(NAME,GARRAY,NDIM) TO PUT VARIABLE NAMED 'NAME' IN
C ONE-D ARRAY 'GARRAY' DIMENSIONED NDIM, THUS:
C CALL GET1D(NAME,GNX,NX) FOR XC...DXG & DIMENSION GNX(NX);
C CALL GET1D(NAME,GNY,NY) FOR YG,...RV & DIMENSION GNY(NY);
C CALL GET1D(NAME,GNZ,NZ) FOR ZG,...WGRID & DIMENSION GNZ(NZ).
C+++++LEGALITY TABLE FOR USE OF EARTH-CONNECTING SUBROUTINES:
C ENTRIES IN TABLE GIVE CHAPTERS IN WHICH SUBROUTINES CAN BE
C USED FOR VARIABLES IN LEFT-HAND COLUMN. (SUBROUTINE
C STRIDE IS REGARDED AS BEING IN CHAPTER 3)
C
C| VARIABLE | GET & | SET | ADD | READIZ | GETID |
C|----------|------|-----|-----|--------|------|
C| P1 | RZ | ALL | 6 & 7 | 5 | 1,2,8,9 | NONE |
C| P10 | RZH | 3-7, 10-16 | 3 | NONE | NONE |
C| VOL | ANDZ | ALL | 3 | NONE | 1,2,8,9 | NONE |
C| D1DP | | NONE | 10 | NONE | NONE |
C| D2DP | | NONE | 11 | NONE | NONE |
C| MUL,MUH | 5,13-16 | 12 | NONE | NONE |
C| EXCOL,H | | 13 | NONE | NONE |
C| CFP | | 5 | 14 | NONE | NONE |
FILE: VANTGRD FORTRAN A1

C    MDT  :  5  :  15  :  NONE  :  NONE  :  NONE  :  VAN00730
C    HST1,HST2 :  5 & 15  :  16  :  NONE  :  NONE  :  NONE  :  VAN00740
C    XG - NGRID :  None  :  None  :  None  :  ALL  :  VAN00750

NOTES ON ABOVE TABLE:
C
* IN CHAPTERS 1, 2, 8, & 9 VARIABLES P1...DM & GEOMETRY
VOL, VHM, AMDZ CAN BE ACCESSED BUT ONLY IN CONJUNCTION WITH
USE OF READIZ, THUS:
DO 1 IZ=1,NZ
CALL READIZ(IZ)
1 CALL GETIZCIZED)

1 CALL GET(...) AS REQUIRED...

* GEOMETRY ACCESSED BY READIZ IS THAT AT INITIAL TIME.

C    IDIF & D2DP ONLY ACCESSIBLE IN UNSTEADY FLOWS.

+++++ GROUND SERVICE SUBROUTINES:
C
* USE CONTUR(NAME,IPLANE,ILOC,NINT,II,12,J1,J2,GARRAY,NDIM) FOR
LINE-PRINTER PLOTS OF CONTOURS. *NAME* = U1,...C4;
*IPLANE* = XPLANE, YPLANE, OR ZPLANE; ILOC SETS IX, IY, OR
*2 LOCATION OF IPLANE; II, J1, & J2 SET FIRST & LAST
CELLS IN HORIZ. & VERT. ON PLOT; GARRAY IS 1-D WORKING ARRAY
OF DIMENSION NX*NY, NY*NZ, OR NY*NX DICTATED BY IPLANE; &
NDIM SETS VALUE OF DIMENSION OF GARRAY.
C
* USE FLDZDA(TITLE,GARRAY,NX,NY) TO PRINT ANY ARRAY DIMENSIONED
GARRAY(NX,NY); SET *TITLE* TO REQUIRED NAME ( 4 HOLLERITH
CHARACTERS ONLY).
C
* USE FLDSDA(TITLE,GARRAY,NX,NY,NZ,IPLANE,ILOC) TO PRINT ANY
ARRAY DIMENSIONED GARRAY(NX,NY,NZ) IN PLANE SPECIFIED BY
*IPLANE* & *ILOC* AS FOR CONTUR ABOVE; SET *TITLE* AS FOR
C
*FLDZDA.

VARIABLE NAMES FOR USE IN GROUND:
COMMON/TYPE/CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME,WALL
COMMON/VAR/P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS
&EP,EP,ML,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
COMMON/VAROLD/P10,PP0,U10,U20,V10,V20,W10,W20,R10,R20,RS0
&EP0,ML0,H30,C10,C20,C30,C40,RX0,RY0,RZ0,S10,S20
COMMON/VARLOW/P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RS,L
&EP,L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
COMMON/VARNX/XG,XU,DXU,DXG
COMMON/VARNY/YG,YV,DYV,DYG
COMMON/VARNZ/ZG,ZW1,DZW,DZG
COMMON/GDMSCL/XPLANE,YPLANE,ZPLANE,ITNO
COMMON/GDMSCL/LSLAB,MSLAB,HSLAB,LAMMU
REAL NORTH,LOW
LOGICAL LSLAB,MSLAB,HSLAB,LAMMU,LSPDA
EQUIVALENCE (ML,R1),(M2,R2)
EQUIVALENCE (IRUN,INTGRDL))
C
SATLIT-EQUIVALENT IRUN:
EQUIVALENCE (IRUN, INTGRDL))
C
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
C
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS.
C
ARRAY ( DIMENSIONED NY,NX ) FOR USE WITH 'ADD':
DIMENSION CVAR(1,1), CVAR(1,1), CM(1,1), VM(1,1), ZER0(1,1)
DIMENSION GP(30,1), GM(30,1), GD(30,1), GV(30,1), GM(30,1)

59
FILE: VANTGRD FORTRAN A1

1. GMACH(30,1), GTEMP(30,1), GVISC(30,1), GWH(30,1), GWM(30,1)
2. GKE(30,1), GY(30,1), GYY(30,1), GYY(30,1), GZ(30,1)

C SPECIAL-DATA ARRAYS DIMENSIONED & DIMENSION VALUES SET HERE.
DIMENSION LSDDA(1), ISDDA(1), RSPDA(1)

C USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
EQUIVALENCE
(RE1, RE2), (GAMA, RE2), (GSWP, RE2),
(RE2, RE2), (GPR, RE2), (GTW, RE2), (GEMU1, RE2), C(JEMU1, INGR(1))
DATA NLSP, NSIP, NSP /1, 1, 1/
DATA CVAR, VVAR, CM, VM, ZERO /5*0.0/

C USER PLACES HIS DATA STATEMENTS HERE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS.
C    PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C    STATEMENTS OF THIS SECTION.
DATA NUMCH4 / 0 /

&CALL RDCM4(IRN, INGR(12), LSDDA, NSIP, RSPDA, NSP)
CALL CMCUTY(ICHAP, ICHAP, IZED, INDVAR)
IF (ICHAP.EQ.-5) GO TO 10
IF (ICHAP.LE.0.OR.ICHAP.GT.16) RETURN
GO TO (100, 200, 300, 4999, 500, 600, 700, 800, 900, 1000, 1100, 1200,
1300, 1400, 1500, 1600), ICHAP
RETURN
4999 NUMCH4 = NUMCH4 + 1
IF (MOD(NUMCH4, 2).EQ.1) GO TO 400
RETURN

CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS.

C---- CHAPTER 0: MODIFY SATLIT DATA, AT START OF EACH IRN.
C    IF (.NOT. NAMLST) RETURN
C    IF (IRN.EQ. RUN) DATFIL = .FALSE.
C--- READ SATLIT DATA NAMELIST HERE
C CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 1 TO 24 )
C READ(20,G1G24)
C CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 25 TO 42 )
C READ(20,G25G42)
RETURN
C---------------------------

C    CHAPTER 1: CALLED AT THE START OF EACH TIME STEP.
C    SET 'DT' HERE WHEN TLAST SET NEGATIVE IN BLOCK DATA.
C    'ATIME + DT' GIVES THE END TIME OF THE CURRENT TIME STEP.
C    NOT ACCESSED IF STEADY OR PARABOLIC.
C------
100 CONTINUE 
RETURN
C------

C    CHAPTER 2: CALLED AT THE START OF EACH SWEEP.
C------

C------

C    CHAPTER 3: CALLED AT THE START OF EACH SLAB;
C NOT ACCESSED IF PARABOLIC, BUT 'STRIDE' IS.
C------
300 CONTINUE 
RETURN
C------

C    CHAPTER 4: CALLED AT THE START OF EACH RE-CALCULATION OF
C VARIABLES P1,...C4 AT CURRENT SLAB. ITNO= ITERATION NUMBER.
C------
400 CONTINUE 
RETURN
C------

C    CHAPTER 5: GROUND CALLED WHEN SOURCE TERM IS COMPUTED.
C INDVAR GIVES DEPENDENT VARIABLE IN QUESTION IE. U1,...C4.
C TO ADD SOURCE TO DEPENDENT VARIABLE Cl(SAY) FOR IX=IXF,IXL
C AND If=ITY.F.ITYL INSERT STATEMENT:
C IF(INDVAR.EQ.C1)
FILE: VANTGRD FORTRAN A1

C   &CALL ADD(INDVAR,IXF,IXL,IYF,IYL,TYPE,CM,VM,CMVAR,VVAR,NY,NX)
VAN02170
C   NOTES ON 'ADD'.
VAN02180
** SOURCE= (CVAR(IY,IX)+AMAX1(0.0,MASFLO))X(VVAR(IY,IX)-PHI),
VAN02190
** 'MASFLO'= CM(IY,IX)X(VM(IY,IX)-P),
VAN02200
** WHERE 'PHI' IS IN-CELL VALUE OF VARIABLE IN QUESTION,
VAN02210
** 'P' IS THE IN-CELL PRESSURE.
VAN02220
** FOR INDVAR= M1, OR =M2, SOURCE ADDED IS 'MASFLO' ONLY,
VAN02230
** EXCEPT IF ONEPHS=F. & MASFLO < 0.0 (IE. OUTFLOW) WHEN
VAN02240
** 'CM'(IY,IX) IS MULTIPLIED BY R1XD1 (FOR M1) 8 R2XD2 (FOR M2).
VAN02250
** BOTH 'CVAR' & 'CM' ARE MULTIPLIED BY CELL-GEOMETRY QUANTITY
VAN02260
** DICTATED BY SETTING OF 'TYPE' (=CELL, EAST AREA,..VOLUME).
VAN02270
** TYPE-SPECIFIED AREAS ARE CALCULATED AS IF BLOCKAGE ABSENT,
VAN02280
** BUT 'VOLUME' WITH ACCOUNT FOR ITS PRESENCE.
VAN02290
C   FOR ALL SOLVED VARIABLES, INCLUDE DING M1 ( & M2 WHEN ONEPHS=F),
VAN02300
C    IF 'CM'> 0.0 CALL 'ADD'; FOR M1 8 M2 ALTHOUGH 'CVAR' 8 'VAR'
VAN02310
C    HAVE NO SIGNIFICANCE THEY MUST BE ENTERED AS ARGUMENTS.
VAN02320
C   'CVAR', 'VM', 'CM' & 'VM' MUST BE DIMENSIONED NY,NX.
VAN02330

500 CONTINUE
VAN02340
RETURN
VAN02350

C     CHAPTER 6 CALLED AT THE END OF EACH VARIABLE-RECALCULATION
VAN02360
C    CYCLE COMMENCED AT CHAPTER 4. ITNO = ITERATION NUMBER.
VAN02370

600 CONTINUE
VAN02380
RETURN
VAN02390

C     CHAPTER 7 CALLED AT END OF EACH SLAB-WISE CALCULATION.
VAN02400

700 CONTINUE
VAN02410
IF(FLOAT(ISWP).LT.GSWP)
VAN02420
RETURN
VAN02430
CALL GET(P1,GP,NY,NX)
VAN02440
CALL GET(H1,GH,NY,NX)
VAN02450
CALL GET(D1,GD,NY,NX)
VAN02460
CALL GET(V1,GV,NY,NX)
VAN02470
CALL GET(W1,GW,NY,NX)
VAN02480
CALL GET(KE,GKE,NY,NX)
VAN02490
C    CALL GET1D(YG,GYG,NY)
VAN02500
CALL GRED1(39,IZED,GYG,NY,NX)
VAN02510
CALL GRED3(57,IZED,GXX,GYY,GZZ,NY,NX)
VAN02520
GCP=RAIR/(1.-1/GAMA)
VAN02530
DO 701 1=1,NY
VAN02540
GSON = SQRT(GAMAXGP(I,1)/GD(I,1))
VAN02550
GAV=SQRT(GV(I,1)XX2+GW(I,1)XX2)
VAN02560
GMACH(I,1)=GAV/GSON
VAN02570
701
VAN02580
GTEMP(I,1)=GP(I,1)/GD(I,1)/RAIR
VAN02590
C 701 GTEMP(I,1) = (GH(I,1)-GW(I,1)XX2/2.-GV(I,1)XX2/2.)/GCP
VAN02600
CALL SET(C1,1,NX,1,NY,GMACH,NY,NX)
VAN02610
CALL SET(C2,1,NX,1,NY,GTEMP,NY,NX)
VAN02620
C------CALCULATE D1 CF ST H(CONVECTIVE COEF.) Q TAU TR
VAN02630
IF(JEMU1.NE.2) GOTO 702
VAN02640
C------TURBULENT VALUES
VAN02650
GCF=2./GW(NY,1)XX2*GKE(1,1)/3.33*GD(1,1)/GD(NY,1)
VAN02660
C7 GCF=GCF*GD(NY,1)/GD(1,1)*GTEMP(NY,1)/GTEMP(1,1)*GP(1,1)/GP(NY,1)
VAN02670
GST=GCF/2./GPRXX.666
VAN02680
GHH=GD(NY,1)*GPRXGHN(1,1)*GST
VAN02690
GR=GPRXX.533
VAN02700
GTR=GTEMP(NY,1)*GMACH(1,1)*GTRX(GAMA-1.)/2.XGMACH(NY,1)XX2
VAN02710
GQ=GHXGTRXGTH
VAN02720
GOTO 703
VAN02730
C------LAMINAR VALUES
VAN02740
200 CONTINUE
VAN02750
IF(JEMU1.EQ.-1) GEMU1=GVISC(1,1)
VAN02760
GQ=GVIS*GDH(1,1)XX2*GTRXXGTH
VAN02770
GTR=GTEMP(NY,1)*GMACH(1,1)/2.*GMACH(NY,1)XX2
VAN02780
GQ=GHXGTRXGTH
VAN02790
GOTO 703
VAN02800
C------END OF FILE
VAN02810

61
FILE: VANTGRD FORTRAN A1

703 GC3(1,1)=GYG(1,1)
704 GC3(2,1)=GCF
705 GC3(3,1)=GST
706 GC3(4,1)=GCF/2./GST
707 GC3(5,1)=GHH
708 GC3(6,1)=GQ
709 GC3(7,1)=GTAU
710 GC3(8,1)=GTR
711 GC3(9,1)=GTR-GTW
712 GC3(10,1)=GD(NY,1)*GW(NY,1)*GZZ(1,1)/GEMU1
713 GC3(11,1)=GZZ(1,1)
714 GC3(12,1)=GEMU1
715 GC3(13,1)=GD(NY,1)*GW(NY,1)*GYG(1,1)/GEMU1*SQRT(ABS(GCF/2.))
716 CALL SET((G3,1,NX,1,NY,GC3,NY,NX)
717 RETURN
718
719 C-----------------------------------------
720 C   CHAPTER 8. CALLED AT THE END OF EACH SHEEP;
721 C   NOT ACCESSED IF PARABOLIC.
722 C-----------------------------------------
723 800 CONTINUE
724 RETURN
725
726 C-----------------------------------------
727 C   CHAPTER 9. CALLED AT THE END OF EACH TIME STEP;
728 C   NOT ACCESSED IF PARABOLIC.
729 C-----------------------------------------
730 900 CONTINUE
731 RETURN
732
733 C-----------------------------------------
734 C   CHAPTER 10. SET PHASE 1 DENSITY HERE WHEN IRH01=-1 IN DATA.
735 C   SET CURRENT-Z 'SLAB' DENSITY, D1, IF MSLAB=.T.,
736 C   EG. IF(MSLAB) CALL SET(D1,1,NX,1,NY,GD1,NY,NX).
737 C   SET NEXT LARGER-Z 'SLAB' DENSITY, D1H, IF HSLAB=.T. & PARAB=F
738 C   EG. IF(HSLAB) CALL SET(D1H,1,NX,1,NY,GD1H,NY,NX).
739 C   SET D(LN(D1))/DP (IE. D1DP) FOR UNSTEADY FLOW,
740 C   EG. IF(MSLAB) CALL SET(D1DP,1,NX,1,NY,GD1DP,NY,NX).
741 C-----------------------------------------
742 1000 CONTINUE
743 IF (MSLAB) GO TO 101
744 JP1=P1H
745 JH1=H1H
746 JD1=D1H
747 JW1=W1H
748 JV1=V1H
749 GO TO 102
750
751 101 JP1=P1
752 JH1=H1
753 JD1=D1
754 JW1=W1
755 JV1=V1
756 GO TO 102
757
758 102 CALL GET(JP1,GP,NY,NX)
759 CALL GET(JH1,GH,NY,NX)
760 CALL GET(JD1,GD1,NY,NX)
761 CALL GET(JW1,GW,NY,NX)
762 CALL GET(JV1,GV,NY,NX)
763 IF(IZED.EQ.1) GOTO 105
764 IF(IZED.EQ.NZ) GOTO 109
765 C-----------------------------------------
766 DO 103 IX=1,NX
767 DO 103 IY=1,NY
768 IF(MSLAB) GOTO 104
769 GHA=GWH(IY,IX)+GVMH(IY,IX)/2.
770 GVMH(IY,IX)=GWH(IY,IX)
771 GOTO 115
772 104 GHA=(GWH(IY,IX)+GVMH(IY,IX))/2.
773 GVMH(IY,IX)=GWH(IY,IX)
774 115 GHS=GHIY,IX)-(GVMH(IY,IX)+GWH(IY,IX)/2.
775 IF(GHS.EQ.1.E5) GHS=1.55
776 103 GDH(IY,IX)= GP(IY,IX)/(1./GAMA)/GHS
777 GOTO 115
778 C-----------------------------------------
779 DO 106 IX=1,NX
780 DO 106 IY=1,NY
781 GHS=GH(IY,IX)-(GWH(IY,IX)+GVMH(IY,IX))/2.
782 105
FILE: VANTGRD FORTRAN A1

IF(GHS.LE.1.E5) GHS=1.E5
GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS
IF(HSLAB) GOTO 107
GWM(IY,IX)=GPM(IY,IX)
GOTO 106
107 GWH(IY,IX)=GPM(IY,IX)
106 CONTINUE
GOTO 113
113 CONTINUE
CONTINUE
CALL SET(ID1,1,NX,1,NY,GD,NY,NX)
RETURN
C------------------------IZED=NZ
109 DO 110 IX=1,NX
DO 110 IY=1,NY
IF(HSLAB) GOTO 111
GHS=GPM(IY,IX)-(GPM(IY,IX)**2+GV(IY,IX)**2)/2.
IF(GHS.LE.1.E5) GHS=1.E5
GWM(IY,IX)=GPM(IY,IX)
GOTO 112
111 GHS=GV(IY,IX)**2+GV(IY,IX)**2)/2.
C IF(GHS.LE.1.E5) GHS=1.E5
GWM(IY,IX)=GP(IY,IX)/(1-1/GAMA)/GHS
110 CONTINUE
C------------------------
113 CONTINUE
CALL SET(D1,1,NX,1,NY,GD,NY,NX)
RETURN
C------------------------CHAPTER 11: SET PHASE 2 DENSITY HERE WHEN IRH02=-1 IN DATA.
C SET CURRENT-Z 'SLAB' DENSITY, D2, IF MSLAB=.T.,
C EG. IF(MSLAB) CALL SET(D2,1,NX,1,NY,GD2,NY,NX).
C SET NEXT LARGER-Z 'SLAB' DENSITY, D2H, IF HSLAB=.T. & PARAB=F
C EG. IF(HSLAB) CALL SET(D2H,1,NX,1,NY,GD2H,NY,NX).
C SET D(LD(D2))/(DP FOR UNSTEADY FLOW,
C EG. IF(MSLAB) CALL SETD(D2DP,1,NX,1,NY,GD2DP,NY,NX).
1100 CONTINUE
RETURN
C------------------------CHAPTER 12: SET PHASE 1 VISCOSITY HERE WHEN IEMU1=-1 IN DATA.
C SET CURRENT-Z 'SLAB' VISCOSITY (MUL), IF MSLAB=.T.
C EG. IF(MSLAB) CALL SETM1,1,NX,1,NY,GVISC,NY,NX).
C SET NEXT LARGER-Z 'SLAB' VISC. (MULH), IF HSLAB=.T. & PARAB=F
C EG. IF(HSLAB) CALL SETM1H,1,NX,1,NY,GVISCH,NY,NX).
C CHAPTER ALSO ACCESSED WHEN EMULAM=-1.0 IN DATA, SO THAT THE
C LAMINAR VISCOSITY WHICH APPEARS IN WALL FUNCTIONS & IN THE
C KE-EP TURBULENCE MODEL (IEMU1=2) MAY BE SET NON-CONSTANT.
C SET CURRENT-Z 'SLAB' VALUE (MULAM) WHEN LAMMU=.T.,
C EG. IF(LAMMU) CALL SETM1AM,1,NX,1,NY,GVISCL,NY,NX).
1200 CONTINUE
GCP=RAIR/(1.-1/GAMA)
IF (HSLAB) GOTO 122
CALL GET(H1,GH,NY,NX)
CALL GET(W1,GW,NY,NX)
CALL GET(V1,GV,NY,NX)
GOTO 123
122 CALL GET(H1H,GH,NY,NX)
CALL GET(W1H,GW,NY,NX)
CALL GET(V1H,GV,NY,NX)
123 CONTINUE
DO 121 IX=1,NX
DO 121 IY=1,NY
GTMP=(GH(IY,IX)-GW(IY,IX)**2/2.-GV(IY,IX)**2/2.)/GCP
IF(GTMP.LT.150.) GTMP=150.
IF (HSLAB) CALL SETM1M(1,NX,1,NY,GVISC,NY,NX)
IF (HSLAB) CALL SETM1M(1,NX,1,NY,GVISC,NY,NX)
IF (LAMMU) CALL SETM1AM(1,NX,1,NY,GVISCL,NY,NX)
RETURN
C------------------------CHAPTER 13: SET EXCHANGE COEFFICIENT (E.C.) FOR VARIABLE

63
INDVAR WHEN SIGMA(INDVAR)=-1.0 IN DATA.

SET CURRENT-Z 'SLAB' E.C. (EXCO) IF MSLAB=.T.,

E.G. IF(MSLAB) CALL SET(EXCO,1,NX,1,NY,GEXCO,NY,NX).

SET NEXT SMALLER-Z 'SLAB' E.C. (EXCOL) IF LSLAB=.T.,

E.G. IF(LSLAB) CALL SET(EXCOL,1,NX,1,NY,GEXCOL,NY,NX).

SET NEXT LARGER-Z 'SLAB' E.C. (EXCOH) IF HSLAB=.T.,

E.G. IF(HSLAB) CALL SET(EXCOH,1,NX,1,NY,GEXCOH,NY,NX).

NOTE: FOR MSLAB, INDVAR=U1,...C4; FOR LSLAB, INDVAR=U1L,...C4L

& FOR HSLAB, INDVAR=U1H,...C4H. IF PARAB=.T. SET MSLAB ONLY.

1300 CONTINUE

RETURN

CHAPTER 14: SET INTER-PHASE FRICTION COEFFICIENT (CFP) HERE

WHEN ICFP = -1 IN DATA; ITS UNITS = FORCE / (CELL * RELATIVE

SPEED OF PHASES).

1400 CONTINUE

RETURN

CHAPTER 15: SET INTER-PHASE MASS-TRANSFER RATE PER CELL (MDT)

HERE WHEN IMDOT = -1 IN DATA.

1500 CONTINUE

RETURN

CHAPTER 16: SET HERE PHASE 1 & 2 SATURATION ENTHALPIES

(HST1 & HST2) WHEN IHSAT = -1 IN DATA.

1600 CONTINUE

RETURN

END
<table>
<thead>
<tr>
<th>No.</th>
<th>Distribution</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Library, Code 0142</td>
<td>Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>2.</td>
<td>Research Administration</td>
<td>Code 012, Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>3.</td>
<td>Department Chairman, Code 69</td>
<td>Department of Mechanical Engineering, Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>4.</td>
<td>Professor Matthew D. Kelleher, Code 69Kk</td>
<td>Department of Mechanical Engineering, Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>5.</td>
<td>Professor Robert H. Nunn, Code 69Nn</td>
<td>Department of Mechanical Engineering, Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>6.</td>
<td>Professor David Salinas, Code 69Sa</td>
<td>Department of Mechanical Engineering, Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>7.</td>
<td>A. Leitner</td>
<td>Visiting Research Associate, Department of Mechanical Engineering, Navy Postgraduate School, Monterey, CA 93943</td>
</tr>
<tr>
<td>8.</td>
<td>Dr. R. Dillinger</td>
<td>Naval Weapon Center, China Lake, CA 93555</td>
</tr>
<tr>
<td>9.</td>
<td>Dr. C. Porter</td>
<td>Naval Weapon Center, China Lake, CA 93555</td>
</tr>
<tr>
<td>10.</td>
<td>Mr. A. Danielson</td>
<td>Naval Weapon Center, China Lake, CA 93555</td>
</tr>
</tbody>
</table>