TURBULENT FLUID FLOW, HEAT TRANSFER AND ONSET OF NUCLEATE BOILING IN ANNULAR FINNED PASSAGES

By

Sang Yong Shim

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree of Doctor of Philosophy

Department of Mechanical and Industrial Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

February 1997

0-612-23663-3

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

COPYRIGHT PERMISSION

TURBULENT FLUID FLOW, HEAT TRANSFER AND ONSET OF NUCLEATE BOILING IN ANNULAR FINNED PASSAGES

BY

SANG YONG SHIM

A Thesis/Practicum submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

I hereby declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

ABSTRACT

Fins are often used in the energy industry for nuclear fuel or compact heat exchanger tubes to enhance the heat transfer rate. Information on turbulent fluid flow and heat transfer in finned passages is rather limited in the literature. This research was motivated to produce a theoretical means of predicting the pressure drop, heat transfer rate and onset of nucleate boiling (ONB) in finned flow passages.

A finite element model was formulated to solve the governing conservation equations of momentum and energy. The finite element method was chosen for ease of representing accurately the irregular geometry under consideration. The turbulence model used is based on a classical mixing length theory which was extended to be applicable for finned geometry.

The numerical model simulated experiments and analyses for annuli and finned annuli available in the literature. This was to show the accuracy of the numerical model and the validity of the turbulence model to the finned annulus geometry. The validated model was then applied to predict the ONB in finned annuli and to study the geometric effects of fin height and number of fins.

Agreement of the present analysis with available experiments and analyses is quite reasonable for fully developed turbulent flow and heat transfer conditions in both annuli and internally finned annuli. The predicted ONB results in conjunction with the Davis and Anderson criterion show good agreement qualitatively and quantitatively with the Atomic Energy of Canada Limited (AECL) data. Both the measured and predicted

ONB occurred at the sheath midway between fins. The predicted ONB followed the trends of the measured data such that the ONB power increases with increasing flow velocity, subcooling or pressure.

The parametric study shows that heat transfer in finned annuli is generally more effective than that in the unfinned annuli for nealy all cases. However, an exception was seen with a tall 8-fin geometry such that heat transfer is slightly less effective than the unfinned annulus, particularly for high flows. The pressure drop increased with the increase of fin height or number of fins for a given mass flow rate (or for a given flow velocity). The ONB for the finned annuli was found to occur at higher powers than that for the unfinned counterparts for the same flow conditions. The ONB heat flux increased with increasing fin height or number of fins. The increase of the ONB heat flux was found more pronounced with low flows.

ACKNOWLEDGMENT

I would like to convey my sincere appreciation to my advisors at the University of Manitoba, Professors Soliman, H.M. and Sims, G.E. for their guidance and encouragement throughout the course of the work.

I would like to thank Professor Britton, M. of the University of Manitoba, and Dr. Krishnan, V.S. and Mr. Richards, D.J. of Atomic Energy of Canada Limited for their keen interest and for providing useful comments through the Advisory Committee. I would also like to thank Dr. Kowalski, J.E. of Atomic Energy of Canada Limited for providing experimental data for the finned annulus geometries.

The support of the Atomic Energy Control Board is gratefully acknowledged.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

FIGURE **PAGE**

LIST OF TABLES

NOMENCLATURE

A	$=2\sigma T_{sat}/(\lambda\rho_v)$
A_l	inter-fin flow area bounded between the sheath and the fin tip, m^2
A_t	total flow area, m^2
A^+	van Driest damping constant
a_1, a_2, a_3	mixing length coefficients in Equation (2-52)
b_1, b_2, b_3	mixing length coefficients in Equation (2-40)
C	$=1+\cos\phi$
C_D	constant used in turbulent kinetic energy equation, Equation (2-23)
C_f	friction coefficient
C_p	specific heat at constant pressure, $J/(kg\cdot K)$
C_μ	constant used in turbulent kinetic energy equation, Equation (2-28)
c_1, c_2, c_3	mixing length coefficients in Equation (2-41)
D	diameter, m
D_f	van Driest damping factor in Equation (2-35)
D_h	hydraulic diameter, m
H	fin height, m
h	heat transfer coefficient, $W/(m^2\cdot K)$
h_{ave}	area-averaged heat transfer coefficient, $W/(m^2\cdot K)$
K	constant in Equation (3-1)
k	turbulent kinetic energy, m^2/s^2

k_l	thermal conductivity, W/(m·K)
k_s	thermal conductivity of solid, W/(m·K)
k_t	turbulent thermal conductivity, W/(m·K)
L	length of the heater, m
L_c	mixing length influenced by channel, m
L_i	mixing length influenced by inner wall in annulus, m
L_o	mixing length influenced by outer wall in annulus, m
L_p	mixing length influenced by pipe, m
l	mixing length, m
l_c	mixing length influenced by channel in Equation (2-38), m
l_p	mixing length influenced by pipe in Equation (2-37), m
m	number of nodes
$\dot{m}$	mass flow rate, kg/s
N	number of fins
n	normal distance from the wall, m
n_x	unit vector in the x direction
n_y	unit vector in the y direction
P	power, W
P_{ht}	heated perimeter, m
Pr	molecular Prandtl number, $\mu C_p/k$
Pr_t	turbulent Prandtl number, $\mu_t C_p/k_t$
P_{wet}	wetted perimeter, m

p	pressure, Pa
Q	heat generation rate, W
q	heat flux, W/m^2
q_{ave}	area-averaged heat flux, W/m^2
q_{gen}	heat generation rate per unit volume, W/m^3
q_{wi}	ONB heat flux, W/m^2
R	gas constant, J/(kg·K)
Re	Reynolds number, $\rho W D_h/\mu$
Re_t	turbulent Reynolds number, $k^2/(\nu\epsilon)$
r	radius, m
r_i	inner radius, m
r_m	the radius of maximum velocity, m
r_o	outer radius, m
s_o	half distance between fins, $r\theta_o$, m.
s^+	dimensionless distance from the wall defined in Equation (2-55)
T	local temperature, °C
T_b	bulk fluid temperature, °C
T_{ft}	fin tip temperature, °C
T_{sat}	saturation temperature, K
T_{sh}	sheath surface temperature between two fins, °C
T_w	wall temperature, °C
T_∞	fluid temperature at $y=\delta$, °C

U	velocity component used in Equation (2-21), m/s
u	time-averaged velocity component in the x direction, m/s
u'	fluctuating velocity component in the x direction, m/s
V_f	heater or fuel volume, m^3
Vol	volume, m^3
v	time-averaged velocity component in the y direction, m/s
v'	fluctuating velocity component in the y direction, m/s
v_t	characteristic velocity scale, m/s
W	cross-sectional average flow velocity in the z direction, m/s
w	time-averaged velocity component in the z direction, m/s
w	test function used in Equation (3-3)
w'	fluctuating velocity component in the z direction, m/s
w^*	friction velocity, $(\tau_w/\rho)^{1/2}$, m/s
w^+	dimensionless velocity, W/w^*
x	Cartesian coordinate in the x direction
y	distance from the wall, m
y	Cartesian coordinate in the y direction
y^+	dimensionless distance from the wall defined in Equation (2-54)
z	Cartesian coordinate in the z (axial) direction

Greek symbols

Γ	boundary
Δp	differential pressure, Pa
δs	distance along the finned periphery, m
δz	axial heated length, m
δ	limiting thermal layer thickness in Equation (5-1), m
ϵ	dissipation of turbulent kinetic energy, m^2/s^3
ε_H	eddy diffusivity for heat transfer, m^2/s
ε_M	eddy diffusivity for momentum, m^2/s
η	master element coordinate
θ	half angle between fins, rad
θ	circumferential direction in a cylindrical coordinate system
θ_{sat}	$=T_{sat}-T_\infty$, °C
θ_w	$=T_w-T_\infty$, °C
θ_{wo}	$=T_{wo}-T_\infty$, °C
κ	von Kármán constant in the interfin region
κ_i	von Kármán constant inside the location of maximum velocity
κ_o	von Kármán constant outside the location of maximum velocity
λ	latent heat of vaporization, J/kg
μ_l	dynamic viscosity, Pa·s
μ_t	turbulent eddy viscosity, Pa·s

ν	kinematic viscosity, μ/ρ, m^2/s
ξ	master element coordinate
ρ	density, kg/m^3
σ_k	constant used in turbulent kinetic energy equation, Equation (2-25)
σ_ϵ	constant used in turbulent kinetic energy dissipation equation, Equation (2-26)
σ	surface tension of liquid with respect to its vapour, N/m
Ω	$=r_m/r_i$
τ	shear stress, Pa
τ_i	shear stress at the inner wall, Pa
τ_o	shear stress at the outer wall, Pa
ϕ	angle of bubble surface with respect to horizontal, rad
ϕ	dependent variable in Equation (3-1)
ψ	approximation function

Subscripts

ave	average
b	bulk
c	channel
f	fuel
ft	fin tip
fs	fin side

i	inside
l	laminar
m	maximum
max	maximum
o	outside
p	pipe
s	solid
sat	saturation
sh	sheath
t	turbulent
w	wall

Superscripts

e	element

CHAPTER 1

INTRODUCTION

1.1 Purpose

Fins are often used in the energy industry for a nuclear fuel or compact heat exchanger tubes to enhance the heat transfer rate. Details of the flow and heat transfer behaviour are necessary to be able to optimize the geometry of fins for given constraints of a particular application. Unlike conventional geometries of a tube and annulus, information on turbulent fluid flow and heat transfer in finned passages is rather limited in the literature. This research was motivated to produce a theoretical means of predicting the pressure drop, heat transfer rate and onset of nucleate boiling (ONB) in finned flow passages. The main advantage of a theoretical approach is that it may not have to rely on experiments every time the geometry and/or operating conditions deviate from the reference design conditions.

Analytical solutions (in a closed form) for pressure drop and heat transfer rate for turbulent flows may be feasible for simple geometries such as a tube or annulus provided that the terms resulting from a turbulence closure are in a simple integrable form. There have been analytical derivations of the friction coefficients and Nusselt numbers for flow in an annulus using a turbulence closure based on the mixing length model [1,2]. However, when fins are attached to an annulus making it a finned annulus, the problem becomes intractable by a strict analytical means. Thus, numerical treatment becomes necessary to be able to consider the complexity of the geometry and flow behaviour.

Therefore, the primary goals of the present study are to:

(1) Develop a computational procedure using a finite element method to solve the governing equations for fully developed, incompressible fluid flow and heat transfer in finned passages. The solution domain will include both the wall and fluid, i.e., the conjugate problem,

(2) Validate the numerical model,

(3) Use the model to predict the heat transfer rate and pressure drop for a single-phase flow up to the onset of nucleate boiling (ONB) in internally finned annuli, and

(4) Perform a parametric study on the effects of fin geometry on pressure drop, heat transfer and onset of nucleate boiling.

1.2 Literature Review

A finned annulus may be considered as an extension from its reference geometry of a simple annulus produced by attaching longitudinal rectangular fins. Thus, the first step of the present study is to review the previous works on turbulent flow and heat transfer for smooth annuli, and then to extend the review to the more complex geometry of finned annuli.

Smooth Annuli

For annuli, a large number of experimental and analytical studies [1-15] have been performed to investigate the characteristics of turbulent fluid flow and heat transfer. Some [2,6,7] of the studies dealt with developing flow near the entrance region, but most of the studies dealt with fully developed flow. Most of the studies dealt with concentric annuli and one [6] dealt with eccentric annuli. Lee and Kim [9] studied the transverse curvature effects to the external flow along a cylindrical body that are relevant to the inner tube of an annulus. Most of the studies dealt with various inner radii and ratios between the inner and outer radii. The studies were usually made in a narrow range of Reynolds number. Most of the studies were done using air and a few [13-15] using water, thus they cover a very limited range of Prandtl number.

The present study concentrates on fully developed, turbulent flow conditions. Turbulent flow is more complex than a laminar flow as the steepest gradients of velocity and temperature take place within a very short distance from the wall. This thin layer governs the pressure loss and heat transfer rate in turbulent flow. This layer in turbulent flow is influenced by Reynolds number and Prandtl number. In addition, unlike flow in tubes or between parallel plates where the maximum velocity occurs at its center, the location of maximum velocity is not stationary in space and moves depending on the ratio between the inner and outer radii. As a result, the flow inside the location of maximum velocity cannot be described by the universal log-law velocity profile in a tube, although the flow outside the location of maximum velocity is similar to tube flow.

The equations governing fully developed flow conditions are the conservation laws of momentum and energy (see details in Sections 2.2 and 4.1.1). With the use of the eddy viscosity concept, the turbulent fluxes of momentum and heat are reduced to the terms containing the turbulent viscosities and diffusivities. The turbulent Prandtl number is used to relate the turbulent viscosities of momentum and diffusivities of heat. It is usually given as a constant or as a function of distance from the wall. The complexity of solutions depends on how the turbulent viscosities are modelled. The previous studies [1-11] based on the Prandtl concept of mixing length [16] have shown this concept able to predict well the characteristics of turbulent flow and temperatures in annuli. The studies differ in the choice of the expressions for turbulent viscosities and the simplifying assumptions used.

To obtain the temperature profile and thus heat transfer rate in a concentric annulus, Lee [2] used the relationship for turbulent eddy viscosity based on a given velocity profile and the shear stress variation. The velocity profile inside the radius of maximum velocity was based on his data since the universal velocity profile is not adequate for this region. The shear stress was obtained from a force balance on an annular fluid element. Similarly, Kays and Leung [1] used a given velocity profile together with given eddy diffusivity expressions from their experimental data to integrate the energy equation.

The velocity profiles both inside and outside the location of maximum velocity were obtained using given expressions for the turbulent viscosities for both inner and outer walls. Lee and Park [7] used the Deissler expression [17] for the region close to both inner and outer walls and Reichardt's expression [18] for the region remote from both walls. Shigechi et al. [8] used the van Driest expression [19] in the sublayer and Reichardt's expression in the fully turbulent layer for eddy diffusivities. Patankar et al. [20] used the mixing length relations to obtain the turbulent viscosities of momentum. The velocity profile was then used to obtain the temperature profiles from the energy equation.

The van Driest expression [19] for the velocity-shear stress relationship was equated with the stress equations based on a force balance in terms of the location of maximum velocity (Wilson and Medwell [10], Quarmby [11]). The location of maximum velocity was obtained by matching the value of maximum velocity given by the inner and outer velocity distributions.

To take into account the transverse curvature effects, Lee and Kim [9] applied a new mixing length model of Hornby et al. [21] to the external flow along a cylindrical body. Lee and Kim [9] obtained the mixing length distribution as a function of wall distance and duct geometry. They considered that the van Driest model is basically for a flat plate and does not recognize the influence of the duct shape.

It is apparent from the review of previous studies on annuli that a more systematic method is needed to model a wide range of the geometry (radius ratio) and heating conditions (constant temperature, constant heating rate, heating on inner, outer, or both walls), and the flow conditions of Re and Pr. The systematic method means less reliance on empiricism and fewer modelling assumptions. Most of the studies necessitated prior knowledge of the location of maximum velocity, the sublayer thickness, and/or the velocity profiles in the inner and outer regions (with respect to maximum velocity) to obtain the heat transfer rate.

In the present study, the location of maximum velocity is numerically determined from the continuous velocity profile in the annulus. The sublayer thickness is calculated based on the van Driest model [19]. Instead of using the wall function frequently used, a fine grid within the thin layers from both inner and outer walls was used to capture the sharp gradients of velocity and temperature. The present study simulated fully developed flow and temperature profiles with constant axial heat rate for a wide range of inner radius, radius ratio, Reynolds number and Prandtl number.

Finned Annuli

For internally finned tubes, turbulent flows were experimentally investigated by Trupp et al. [22,23] and Edwards et al. [24], who studied the local flow structure in internally finned tubes. Both studies dealt with fully developed isothermal air flows by varying Reynolds number. The finning configuration consisted of longitudinal rectangular fins equally spaced around the periphery of a tube. Trupp et al. used tall fins (H/r_o=0.67, defined as fin height/tube radius) while Edwards et al. used short fins (H/r_o=0.17 and 0.33) and varied the number of fins for the same tube radius. Both studies reported details of flow structures such as axial velocity distribution, secondary velocities, friction factor and local shear stress distribution along the tube and fin surface. Said and Trupp [25] used a high-Reynolds number k-ϵ model with the wall functions to predict detailed flow and heat transfer structures under fully developed turbulent flow conditions. Kim and Webb [26] developed an approximate solution to predict the friction factor and Nusselt number for turbulent flow in internally finned tubes. The model assumed a uniform wall shear stress around the fin and the logarithmic velocity and temperature profiles in the core and interfin regions of the flow.

In contrast, for internally finned annuli, very few experimental studies [27-31] have been performed. De Lorenzo and Anderson [28] presented heat transfer coefficients and friction factors for low Reynolds numbers up to 4000 for three finned annuli having different number of fins. Longitudinal fins were attached to the outside of the inner pipe. They indicated the transition between laminar and turbulent flow at Reynolds number of 400. Atomic Energy of Canada Limited (AECL) [27, 29-31] has performed experimental studies for the problem of finned annuli. They measured the inner wall midway between fins and fin tip temperatures under single-phase and two-phase (water/steam) flow conditions to determine the heat transfer coefficients. The data indicated that high internal heat generation produced highly nonuniform surface temperature distribution on the finned surface. The effects of hydraulic diameter and fin geometry were also investigated for a variety of flow conditions. However, detailed flow and temperature measurements were not made in these studies [27-31].

An analytical model was presented by Patankar et al. [20] for fully developed turbulent air flow in internally finned tubes and annuli. Their model assumed zero-thickness fins, uniform wall temperature and no secondary flow. It used the thermal boundary condition of constant axial heat input, and is based on a mixing-length turbulence model. Ivanovic [32] compared the variation of local heat transfer coefficients along the tube wall and fin height using a mixing length approach and a low-Reynolds-number k-ϵ model. The results of heat transfer coefficients around the finned surface from both models were found to be nearly identical. These analytical studies did not consider the actual interaction between wall heat conduction and fluid convection, and assumed infinite wall conductance (i.e., uniform wall temperature).

Based on the limited information on finned annuli, it is evident that a more realistic model is needed to deal with a fully conjugated problem where the heat is internally generated, is conducted in the finned sheath, and leaves from the finned surface to the fluid by turbulent convection. The continuity of temperature and heat flux at the surface/fluid interface determines the heat transfer rate. The heat transfer rate is influenced not only by the finning configuration and the finning material, but also the flow conditions (Re and Pr) and internal heat generation rate. Therefore, the present study deals with a fully conjugated problem of heat conduction in the solid and convection in the fluid for finned annuli.

The Onset of Nucleate Boiling (ONB)

The onset of nucleate boiling is defined as the condition at which the first bubble appears on a heating surface.

Hsu [33] developed a theory for the ONB which related the size range of active nucleation sites on a heating surface to the superheat required to initiate nucleate boiling in the liquid. He related this superheat equation to a transient heat conduction equation and derived the effective cavity sizes to be the ones which take a finite waiting period for the liquid to attain superheat to grow into a bubble. The model is based on the thermodynamic equilibrium criterion (the Clausius-Clapeyron equation) and the Gibbs equation for surface tension. These two equations are combined to give the superheat equation at which nucleate boiling will occur. The Hsu criteria gave the maximum and minimum sizes of effective cavities as a function of subcooling, pressure, physical properties and the thickness of the superheated layer. His theory requires the knowledge of the thermal layer thickness within which a bubble nucleus can develop. Hsu defined a thermal layer thickness δ similar to a laminar sublayer within which the transport is a result of molecular action only ($y<\delta$). Outside the thermal layer ($y\geq\delta$), turbulent motion is presumed sufficiently strong so that the liquid temperature remains constant at T_∞.

Han and Griffith [34] proposed an analysis similar to Hsu's. Bergles and Rohsenhow [35] adapted the Han and Griffith analysis to develop a criterion for the ONB for a system with a wide range of cavity sizes. Assuming a linear temperature profile in the vicinity of a hemispherical bubble nucleus and using an equilibrium theory to describe the superheat needed for equilibrium of the bubble, they developed a graphical technique for predicting the ONB. They established an empirical design equation for the heat flux required to initiate nucleate boiling in water.

A model proposed by Davis and Anderson [36] is also based on the thermodynamic equilibrium criterion (the Clausius-Clapeyron equation) and the Gibbs equation for surface tension. These two equations were combined to give the superheat equation at which nucleate boiling will occur. They equated the slope of the superheat equation with the temperature profile at the wall and solved for the critical distance from the wall required to initiate nucleate boiling provided that cavities of the size corresponding to this critical distance exist.

In this study, detailed flow and temperature profiles are predicted from the present model for the finned annulus geometry. The predicted temperature and heat flux distributions are used in conjunction with the ONB criteria of Hsu [33] and Davis and Anderson [36] to determine the ONB.

1.3 Scope of Present Study

In the present study, a mathematical model is formulated to study turbulent fluid flow and heat transfer in finned passages. The governing equations are solved numerically using a finite element method to represent finned geometries accurately. The numerical results are compared with available experimental data, analytical solutions and other numerical results for fully developed turbulent flow and heat transfer conditions for annuli and finned annuli. The analysis covers a wide range of Prandtl numbers, Reynolds numbers and geometries for turbulent flow in annuli and finned annuli. The analysis is then extended to predict the ONB and to study geometric effects of fin height and number of fins.

The present analysis includes improvements over previously published analyses as follows:

(1) A generalized computational procedure based on a finite-element model was developed for solving a conjugate heat transfer problem whose governing equations may be a number of nonlinear, coupled, partial differential equations. The model may be extended to simulate flows in a more complex geometry (e.g., nuclear reactor fuel subchannels).

(2) The present analysis of finned annulus geometries represents the geometry and boundary conditions of the problem accurately. Thus, the simplifying assumptions used in the previous analysis of Patankar et al. [20] became unnecessary, e.g., thin (zero-thickness) fins, a uniform finned surface temperature and a correlation for the radius of maximum velocity. The geometry is modelled accurately and so the effects of the fin space which was neglected in Patankar et al. on the flow and temperature fields are fully considered. The heat generation rate is specified in the heater or the fuel rather than assuming a uniform wall temperature along the finned surface. Thus, the effects of nonuniform fluid and heater temperature distributions on the heat transfer rate are captured. The numerically determined locations of maximum velocity are used.

(3) Results were obtained of the heat transfer and flow behaviour in finned/unfinned annuli for a wide range of geometries (e.g., annulus radius ratio, fin height, number of fins), Reynolds number (10^4 to 10^6), Prandtl number (0.7 to 10). The analyses of Patankar et al. [20] covered fully developed turbulent air flow in internally finned tubes and annuli for Re of 10^4 to 10^5 and Pr of 0.7. The effects of variable fluid properties were also considered in the present analysis.

(4) The analysis was extended to predict the ONB in internally finned annuli. The results were compared with experimental data obtained at AECL.

(5) Geometric studies using the model were performed by varying fin height and number of fins.

1.4 Outline of Presentation

The present study is presented in the following order:

(1) The governing equations with the turbulence closure in Chapter 2,

(2) The numerical procedure of the finite element formulation in Chapter 3,

(3) Validation tests against available experimental data and previous analytical studies for single-phase flow in annuli and finned annuli, and the parametric study of fins on the heat transfer and pressure drop in Chapter 4,

(4) An extension of the model to the ONB prediction, and the geometric effect study of fins on the ONB in Chapter 5, and

(5) Conclusion and recommendations for future studies in Chapter 6.

CHAPTER 2

MATHEMATICAL FORMULATION

2.1 Description of Problem

An internally finned annulus is considered as a base geometry in the present study. Figure 2.1 shows the cross section of a finned annulus which has eight internal fins. This is one of the geometries used in the AECL experimental study [27]. The eight longitudinal, rectangular fins are equally spaced around the inner wall.

A one-sixteenth part of the cross section is used due to the symmetry. The calculation domain consists of three distinct regions:

(1) the heater tube (Region 1),

(2) the sheath and fin (Region 2), and

(3) the fluid (Region 3).

The present study is also able to consider other geometries of annulus or thin fins. This is achieved by applying appropriate boundary conditions and physical properties as required by the governing equations and the underlying assumptions. The details of modelling are described in the analysis sections.

The following boundary conditions are applied (see Figure 2.1):

$$\frac{\partial w}{\partial \theta}=0 \text{ , } \textit{at the symmetry planes} \tag{2-1}$$

$$\frac{\partial T}{\partial \theta}=0 \text{ , } \textit{at the symmetry planes} \tag{2-2}$$

$$w=0 \text{ , } \textit{at the walls} \tag{2-3}$$

$$T=\textit{specified} \text{ , } \textit{at a node} \tag{2-4}$$

$$q_{gen}=\textit{specified} \text{ , } \textit{in region } 1 \tag{2-5}$$

2.2 Governing Equations

In all theoretical studies of the turbulent motion of a viscous fluid, it is assumed that the Navier-Stokes and energy equations are valid for the actual irregular motion. However, in view of the complexity of the paths of fluid particles in turbulent motion, the solution of the appropriate Navier-Stokes and energy equations is complicated and impracticable. Therefore, the main problems of turbulent fluid motion are to obtain the time-averaged velocity and temperature fields. The equations governing such mean fields are obtained by time-averaging the dependent variables of the conservation equations for the actual motion.

The Navier-Stokes and energy equations governing laminar flows remain valid for turbulent flows. The only difference between the two sets of equations is that the dependent variables (*u*, *v*, *w*, *T* and *p*) for turbulent flows become instantaneous quantities

($\phi=\Phi+\phi'$, where ϕ is an instantaneous value, Φ is an time-averaged value of ϕ, and ϕ' is the fluctuation).

In the present study, the governing equations are formulated in Cartesian coordinates. The finite element method used in the study can describe a curvature accurately given that a sufficient number of elements are provided. It was Reynolds who first derived the system of averaged turbulent-motion equation in 1895 (Buleev [37]). The equations of motion in Cartesian coordinates for an incompressible fluid are (Kays and Crawford [38])

$$\frac{\partial\rho u}{\partial t}+\frac{\partial\rho u^2}{\partial x}+\frac{\partial\rho vu}{\partial y}+\frac{\partial\rho wu}{\partial z}=-\frac{\partial p}{\partial x}+\mu_t\nabla^2 u-\frac{\partial\rho\overline{u'u'}}{\partial x}-\frac{\partial\rho\overline{v'u'}}{\partial y}-\frac{\partial\rho\overline{w'u'}}{\partial z} \tag{2-6}$$

$$\frac{\partial\rho v}{\partial t}+\frac{\partial\rho uv}{\partial x}+\frac{\partial\rho v^2}{\partial y}+\frac{\partial\rho wv}{\partial z}=-\frac{\partial p}{\partial y}+\mu_t\nabla^2 v-\frac{\partial\rho\overline{u'v'}}{\partial x}-\frac{\partial\rho\overline{v'v'}}{\partial y}-\frac{\partial\rho\overline{w'v'}}{\partial z} \tag{2-7}$$

$$\frac{\partial\rho w}{\partial t}+\frac{\partial\rho uw}{\partial x}+\frac{\partial\rho vw}{\partial y}+\frac{\partial\rho w^2}{\partial z}=-\frac{\partial p}{\partial z}+\mu_t\nabla^2 w-\frac{\partial\rho\overline{u'w'}}{\partial x}-\frac{\partial\rho\overline{v'w'}}{\partial y}-\frac{\partial\rho\overline{w'w'}}{\partial z} \tag{2-8}$$

The continuity equation is

$$\frac{\partial\rho}{\partial t}+\frac{\partial\rho u}{\partial x}+\frac{\partial\rho v}{\partial y}+\frac{\partial\rho w}{\partial z}=0 \tag{2-9}$$

The fluid energy equation is

$$\frac{\partial T}{\partial t}+\frac{\partial uT}{\partial x}+\frac{\partial vT}{\partial y}+\frac{\partial wT}{\partial z}=\frac{k_l}{\rho C_p}\nabla^2 T-\frac{\partial \overline{uT'}}{\partial x}-\frac{\partial \rho\overline{vT'}}{\partial y}-\frac{\partial \overline{wT'}}{\partial z} \tag{2-10}$$

The solid energy equation is

$$\frac{\partial \rho T}{\partial t}=\rho k_s \nabla^2 T+Q \tag{2-11}$$

where *u, v, w* and *T* are the time-averaged velocity components and temperature, x and y denote the *x* and *y* coordinate at a cross section, z denotes the axial coordinate in the flow direction, and *u', v', w'* and *T'* are the fluctuating velocity components and temperature.

The turbulence models most widely used in applications have been based on the Boussinesq eddy viscosity concept (Kays and Crawford [38]). The turbulent fluxes of momentum and energy in Equations (2-6) to (2-8) and Equation (2-10) are related to the mean-velocity and mean-temperature gradients via a turbulent viscosity and a thermal diffusivity, respectively.

$$-\overline{u'w'}=\varepsilon_M \frac{\partial w}{\partial x} \tag{2-12}$$

$$-\overline{v'w'}=\varepsilon_M \frac{\partial w}{\partial y} \tag{2-13}$$

$$-\overline{u'T'}=\varepsilon_H \frac{\partial T}{\partial x} \tag{2-14}$$

$$-\overline{v'T'}=\varepsilon_H \frac{\partial T}{\partial y} \tag{2-15}$$

where $\varepsilon_M(=\mu_t/\rho)$ and $\varepsilon_H(=k_t/(\rho c_p))$ are the eddy diffusivity for momentum and that for heat transfer, respectively. Both ε_M and ε_H were assumed to be isotropic within the flow domain.

The problem under consideration deals with a steady, incompressible, constant cross-sectional duct flow. Therefore all of the derivatives with respect to time, t, are neglected. With the assumption of no secondary flow (i.e., u=0, v=0), Equation (2-8) is sufficient for determining the axial velocity profile in the x and y plane. The last term in Equation (2-8), which is the derivative of a turbulent stress in the direction of flow, is generally found to be negligible (Kays and Crawford [38]). Velocity, w, is invariant with respect to the direction of flow, z, for a fully developed flow profile (i.e., $\partial w/\partial z$=0). Therefore, Equations (2-8) and (2-10) for steady, two-dimensional, fully-developed turbulent flow of an incompressible fluid reduce to:

$$\frac{\partial}{\partial x}\left[(\mu_l+\mu_t)\frac{\partial w}{\partial x}\right]+\frac{\partial}{\partial y}\left[(\mu_l+\mu_t)\frac{\partial w}{\partial y}\right]-\frac{dp}{dz}=0 \tag{2-16}$$

$$\frac{\partial}{\partial x}\left[(k_l+k_t)\frac{\partial T}{\partial x}\right]+\frac{\partial}{\partial y}\left[(k_l+k_t)\frac{\partial T}{\partial y}\right]-\rho C_p w\frac{dT}{dz}=0 \tag{2-17}$$

where p is pressure, μ_l dynamic viscosity, μ_t turbulent eddy viscosity, k_l thermal conductivity, k_t turbulent thermal conductivity and q_{gen} heat generation rate per unit volume.

Equation (2-11) with the assumption of no axial conduction (in the z direction) reduces to:

$$\frac{\partial}{\partial x}\left[k_s\frac{\partial T}{\partial x}\right]+\frac{\partial}{\partial y}\left[k_s\frac{\partial T}{\partial y}\right]+q_{gen}=0 \tag{2-18}$$

2.3 Review of Turbulence Models

Turbulent flows are very important in practical applications. However, turbulence modelling is the most uncertain feature of theoretical predictions for turbulent forced convection. It is difficult to make the turbulence closure applicable for all turbulent flows. Although considerable effort has been devoted to the development and evaluation of turbulence models, to date no model has been found to be both accurate and general.

For a suitable characteristic length scale l and velocity scale v_t, the use of dimensional reasoning suggests that the turbulent viscosity may be evaluated as

$$\mu_t=\rho\ v_t\ l \tag{2-19}$$

Closure through the Boussinesq assumption can be considered as specifying suitable expressions for v_t and l. Models based on the Boussinesq assumption are called turbulent viscosity models (or algebraic models).

Although experimental evidence indicates that the turbulent viscosity models are reasonably valid in many flow problems, there are exceptions. A class of models has been developed that effects closure without this assumption. These generally require the solution of transport partial differential equations for the Reynolds stresses known as Reynolds stress models. Turbulence models are often classified according to the number of partial differential equations that must be solved in order to supply the modelling parameters.

A detailed review and evaluation of turbulence models is beyond the scope of the present study, but brief descriptions of existing models are given below. A good review of various turbulence models may be found in References 39 to 42.

Zero Equation Models

In an algebraic model, following Prandtl [16], the characteristic velocity of turbulence is obtained from $l|\partial u/\partial y|$ and l is evaluated from the local geometry of the flow, i.e., distance from the wall and the boundary layer thickness. The mixing length physically means the distance over which a fluid particle travels before exchanging momentum with fluid particles of different layers. The mixing length is small in comparison with the channel dimensions. Algebraic models have proven to be accurate and reliable for relatively simple flows but need to be modified to predict flows with complicating features such as modifications to account for low Reynolds number effects, surface roughness, wall blowing and suction, strong pressure gradients and streamwise curvature (Pletcher [39]).

Prandtl assumed that

$$-\overline{u'w'}=\varepsilon_M\frac{\partial w}{\partial x} \tag{2-20}$$

The Prandtl mixing length hypothesis can be written in a generalized form [32]

$$\nu_t=l^2\left[\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j}\right]^{\frac{1}{2}} \tag{2-21}$$

Algebraic models have been criticized for their lack of generality. The adjustments needed to accommodate special effects have no physical basis and the constants in the models are changed to handle different classes of flow problems. Closures of all models suffer from these shortcomings to a certain degree, but some advantage in generality can be obtained through the use of more complex models. Algebraic models have continued to be used, especially for the calculation of flows that demand large computing times due to multi-dimensionality or geometric complexity.

Additional details of this model can be found in Section 2.4 as it forms the basis of the turbulence model used in the present study.

One-Equation Models

The most common one-equation model follows the suggestions of Prandtl and Kolmogorov made in the 1940s to let v_t be proportional to the square root of the turbulent kinetic energy k

$$\mu_t=\rho c_\mu k^{\frac{1}{2}} l \qquad (2\text{-}22)$$

where c_μ is a constant, usually taken as 0.09, and l a turbulent length scale.

A transport partial differential equation for the turbulent kinetic energy can be derived from the Navier-Stokes equations but the terms representing diffusion, generation and dissipation of k introduce additional unknowns involving higher momentums of fluctuating quantities. These are determined through additional assumptions.

The one-equation model appears at least more physical since it gives $k \neq 0$ for $\partial w/\partial r=0$ whereas the zero-equation model indicates $k=0$ for $\partial w/\partial r=0$.

Two-Equation Models

Two-equation models permit the determination of both a characteristic velocity v_t and a length scale l from the solution of transport partial differential equations. One of these transport equations is for the determination of turbulent kinetic energy k. Although a second transport equation can be developed for a length scale, a transport equation is solved for a length scale related parameter rather than the length scale itself. One of the most widely used two-equation models is the k-ϵ model first proposed by Harlow and Nakayama [43] and further developed by Jones and Launder [44]. The parameter ϵ is the turbulence dissipation rate and is assumed to be related to the length scale through

$$\epsilon = C_D \frac{k^{\frac{3}{2}}}{l} \tag{2-23}$$

The turbulent viscosity is evaluated in terms of k and ϵ by

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{2-24}$$

Launder [45] suggests that, for accuracy and wider applicability, a fine-grid low-Reynolds number treatment be employed near the wall in place of wall functions, despite the attractive simplicity of the latter approach. The local heat transfer coefficient is determined to a very large extent by the variation of the effective diffusivity within the immediate vicinity of the wall. This observation is applied even more strongly where the fluid is water or one with an even higher Prandtl number.

A low-Reynolds number k-ϵ turbulence model has been widely used and has shown good results for boundary layer flows in pipes (References 45 to 47). Jones and Launder [44] extended the application of a high-Reynolds k-ϵ model to low-Reynolds number turbulent flows as

$$\frac{Dk}{Dt}=\frac{1}{\rho}\frac{\partial}{\partial x_j}\left[(\mu+\frac{\mu_t}{\sigma_k})\frac{\partial k}{\partial x_j}\right]+\nu_t\frac{\partial u_i}{\partial x_j}\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right]-\epsilon-2\nu\left[\frac{\partial k^{\frac{1}{2}}}{\partial x_j}\right]^2 \tag{2-25}$$

$$\frac{D\epsilon}{Dt}=\frac{1}{\rho}\frac{\partial}{\partial x_j}\left[(\mu+\frac{\mu_t}{\sigma_\epsilon})\frac{\partial \epsilon}{\partial x_j}\right]+C_1\nu_t\frac{\epsilon}{k}\frac{\partial u_i}{\partial x_j}\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right]-C_2\frac{\epsilon^2}{k}+2\nu\nu_t\left(\frac{\partial^2 u_i}{\partial x_j\partial x_i}\right)^2 \tag{2-26}$$

The constants C_1=1.44, σ_k=1.0, and σ_ϵ=1.3 are used.

$$\mu_t=C_\mu\frac{\rho k^2}{\epsilon} \tag{2-27}$$

$$C_\mu=0.09\exp\left[-\frac{2.5}{\left(1+\frac{Re_t}{50}\right)}\right] \tag{2-28}$$

$$C_2=1.92[1-0.3\exp(-Re_t^2)] \tag{2-29}$$

$$Re_t=\frac{k^2}{\nu\epsilon} \tag{2-30}$$

The difference from the high-Reynolds number k-ϵ models is the last term in Equations (2-25) and (2-26) added for the low-Reynolds number model. Various forms in place of these terms were used in the literature (References 49 and 50). The last term in Equation (2-25) was introduced for computational rather than physical reasons. Because of the difficulty of specifying ϵ at the wall as a boundary condition, ϵ was set to zero at the wall and the dissipation energy rate in the vicinity of the wall is compensated. The last term in Equation (2-26) is included in the ϵ equation to produce satisfactory variation of k with the distance from the wall.

The main weakness of the model is that it is based on the concept of an isotropic eddy viscosity. For this reason, predictions are often poor for flows with recirculation, streamline curvature, and buoyancy effects unless the constants in the model are adjusted [39].

Reynolds Stress Models

Although two-equation models have a reasonable degree of flexibility, they are restricted by the assumption of an isotropic turbulent viscosity and the assumption that the stresses are proportional to the rate of mean strain. Reynolds stress models are free of these restrictions. Transport partial differential equations are solved for the Reynolds stresses and heat fluxes.

The transport equations can be derived in an exact form but contain terms that must be approximated to close the system. Several closure schemes have been proposed. One that is widely used follows from the work of Launder et al. [48].

The Reynolds stress models contain the greatest number of partial differential equations and constants. They can, in principle account for effects such as buoyancy, curvature and rotation without ad hoc adjustments. On the other hand, the determination of the optimum modelling formulation and values of constants is not easy. The computational effort required by the Reynolds stress models is significantly greater than that for the less complex models and to date they have received limited use in engineering predictions. A description of many of the improvements and applications can be found in the reviews by Rodi [42], Nallasamy [49] and Patel et al. [50].

Rodi [51] has proposed a useful algebraic simplification to the Reynolds stress model. He assumed that the transport of Reynolds stresses was proportional to the transport of turbulent kinetic energy k. The result is an algebraic relationship between the stresses and k, ϵ, and derivatives of mean flow quantities. Transport equations are solved for k and ϵ so that the algebraic Reynolds stress model can be considered as an extended k-ϵ model. The model appears attractive for accounting for effects of buoyancy, rotation and streamline curvature in an economical fashion. It is not, however, equivalent to a full Reynolds stress model because of the additional assumptions made to convert the expressions for Reynolds stresses to an algebraic form.

2.4 Present Turbulence Model

Since turbulence modelling is the most important link in the predictive procedure, improvements in predictive capability come mainly through the development and verification of improved turbulence models. It is not clear at present whether the best way is through the approach in which a number of models each finely tuned for specific conditions are employed, or the development of a single general model for the Reynolds equations capable of predicting a wide range of flows.

For the present study, the mixing length model (so called the zero equation model) is chosen for the following reasons:

(1) The mixing length theory has been applied to the tube and annulus geometries over four decades and shown good agreement with available experimental data. The finned annulus geometry is considered to be an extension of the basic annulus geometry.

(2) Ivanovic [32] showed that the mixing length model produced local heat transfer coefficients along the finned periphery in good agreement with a low-Reynolds number k-ϵ turbulence model of Jones and Launder [44].

(3) Since the finite element method used in the present study utilizes a fine mesh near the wall boundaries instead of special wall functions, a simple turbulence model is preferred to keep calculation times reasonable.

To close Equations (2-16) and (2-17), μ_t and k_t must be determined first. Ivanovic [32] examined a number of candidate turbulence models for use in the finned geometry. This work concluded that the high Reynolds number k-ϵ model using the wall functions cannot be used in near-wall regions. In addition, it found that a substantial portion of the inter-fin space was in effect a near-wall region (i.e., $y^+<15$). Since the available wall functions account for the influence of only a single wall, they are not suitable for the finned geometries where the influences of both tube wall and fin walls are important. Thus for the present study, a turbulence model based on the mixing length theory was adapted to calculate the turbulent viscosity and thermal diffusivity.

From Equation (2-21), the eddy viscosity for a two-dimensional flow is obtained

$$\mu_t=\rho l^2\left[\left(\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial w}{\partial y}\right)^2\right]^{\frac{1}{2}} \tag{2-31}$$

$$k_t=k_l\,\frac{\mu_t}{\mu_l}\,\frac{Pr}{Pr_t} \tag{2-32}$$

$$Pr_t=\frac{\epsilon_M}{\epsilon_H}=\frac{\overline{w'v'}\dfrac{\partial T}{\partial y}}{\overline{T'v'}\dfrac{\partial w}{\partial y}} \tag{2-33}$$

where Pr is the molecular Prandtl number and Pr_t is the turbulent Prandtl number.

The essence of the heat and momentum analogy method of calculation lies in the assumption of a definite relationship between the thermal and momentum eddy diffusivities. A wide range of the ratio of momentum to thermal eddy diffusivity ($\epsilon_M/\epsilon_H=Pr_t$) was used in the literature for annuli some of which are 0.9 by Patankar et al. [20], 0.83 by Kays and Leung [1], and 1.0 for $Pr\geq 0.1$ by Wilson and Medwell [10]. A review of the turbulent Prandtl number can be found in Kays [52].

The tube wall and the fin surface simultaneously influence the mixing length. The closer the point is to one of these surfaces, the greater should be the effect of that surface on the resultant mixing length. To fulfill this requirement, a superposition method proposed in Reference 20 is used.

Consider a point which is situated at distance y from the tube wall and distance s from the fin surface. The mixing length at the point is taken to be the resultant of two contributions. First, considering a pipe flow without fins, the mixing length at y is $l_p(y)$ where subscript p denotes pipe flow. Next, if an analogy between the inter-fin space and a parallel plate channel is applied, the mixing length at s is $l_c(s)$ where c denotes channel flow. The mixing length, l_p is obtained from the Nikuradse work on turbulent pipe flow (Reference 53).

The van Driest damping function $(D_f)_p$ (Reference 19) is used so as to extend the Prandtl mixing length all the way to the wall instead of truncating it to zero at an assumed outer edge of the sublayer. It bridges the gap between the fully turbulent region and the viscous sublayer. The expression for the mixing length proposed by van Driest was used

$$\frac{1}{l}=\frac{1}{l_p}+\frac{1}{l_c} \tag{2-34}$$

$$(D_f)_p=1-\exp(-\frac{y^+}{A^+}) \tag{2-35}$$

$$(D_f)_c=1-\exp(-\frac{s^+}{A^+}) \tag{2-36}$$

$$l_p=(D_f)_p L_p \tag{2-37}$$

$$l_c=(D_f)_c L_c \tag{2-38}$$

where $A^+=26$

For the analysis of an annulus, a fully developed turbulent flow is considered with inner radius r_i and outer radius r_o. The velocity, which is zero at r_i, increases with increasing r, attaining a maximum at $r=r_m$, and then decreases to zero at r_o. The radius of maximum velocity is numerically determined.

A mixing length expression of Nikuradse type is used for both inside the radius of maximum velocity $r_i \le r \le r_m$, and outside the radius of maximum velocity of an annulus $r_m \le r \le r_o$. As shown in Figure 2.2, the following is defined:

$$\begin{aligned} y&=r-r_i \\ y_m&=r_m-r_i \\ y_o&=r_o-r_i \\ y_{om}&=y_o-y_m \end{aligned} \tag{2-39}$$

The mixing lengths in the region inside r_m and in the region outside r_m of an annulus are given, respectively, by L_i and L_o

$$\frac{L_i}{y_m}=b_1-b_2(1-\frac{y}{y_m})^2-b_3(1-\frac{y}{y_m})^4 \tag{2-40}$$

$$\frac{L_o}{y_{om}}=c_1-c_2\frac{(y-y_m)^2}{y_{om}^2}-c_3\frac{(y-y_m)^4}{y_{om}^4} \tag{2-41}$$

The constants c_1, c_2 and c_3 for a pipe flow are obtained from the Nikuradse work on turbulent pipe flow (Reference 53) as c_1=0.14, c_2=0.08, c_3=0.06. The flow outside r_m is similar to a pipe flow except that the location of maximum velocity is situated at a distance y_{om} from the outer wall instead of at the pipe centerline. With this reason, the constants c_1, c_2 and c_3 were taken equal to those for a pipe flow.

The constants b's in Equation (2-40) are determined to satisfy the following three conditions:

(1) $L_i=0$ at $y=0$,

(2) $(dL_i/dy)=\kappa_i$ at $y=0$ and

(3) $L_i=L_o$ at $y=y_m$.

Applying these three conditions to Equation (2-40) results in

$$\begin{aligned} b_1&=0.14\frac{y_{0m}}{y_m}\\ b_2&=2b_1-0.5\kappa_i\\ b_3&=0.5\kappa_i-b_1 \end{aligned} \tag{2-42}$$

Thus for given κ_i, b_1, b_2 and b_3 can be determined.

The coefficient κ_i is evaluated using the relationship derived by Roberts [3].

The relationship of κ_i with κ_o was obtained using the Reichardt [18] Equation (2-44) for the eddy viscosity in a pipe flow together with a corresponding Equation (2-43) for the region inside r_m.

$$\frac{\epsilon_i}{\nu}=\frac{\kappa_i}{6}(r_m-r_i)\frac{u_i^*}{\nu}\left[1-\left(\frac{r_m-r}{r_m-r_i}\right)^2\right]\left[1+2\left(\frac{r_m-r}{r_m-r_i}\right)\right] \tag{2-43}$$

$$\frac{\epsilon_o}{\nu}=\frac{\kappa_o}{6}(r_o-r_m)\frac{u_o^*}{\nu}\left[1-\left(\frac{r-r_m}{r_o-r_m}\right)^2\right]\left[1+2\left(\frac{r-r_m}{r_o-r_m}\right)\right] \tag{2-44}$$

Equating Equations (2-43) and (2-44) at $r=r_m$ yields

$$\kappa_i=\kappa_o\frac{r_o-r_m}{r_m-r_i}\sqrt{\frac{\tau_o}{\tau_i}} \tag{2-45}$$

The shear stress variation as a function of the radial coordinate is obtained from a force balance on a cylindrical element of fluid in the annular cross section (Knudsen [54]). It is assumed that the positions of maximum velocity and zero shear stress are coincident (Brighton and Jones [5]). For $r_i \le r \le r_m$, carrying out a force balance on an annular element of fluid gives

$$-\frac{dp}{dx}=2\frac{\tau_i r_i-\tau r}{r^2-r_i^2} \tag{2-46}$$

$$\frac{\tau}{\tau_i}=\frac{r_i}{r}\left(\frac{r_m^2-r^2}{r_m^2-r_i^2}\right) \tag{2-47}$$

For $r_m \le r \le r_o$,

$$-\frac{dp}{dx}=2\frac{\tau_o r_o-\tau r}{r_o^2-r^2} \tag{2-48}$$

$$\frac{\tau}{\tau_o}=\frac{r_o}{r}\left(\frac{r^2-r_m^2}{r_o^2-r_m^2}\right) \tag{2-49}$$

Combining Equations (2-47) and (2-49) gives

$$\frac{\tau_i}{\tau_o}=\frac{r_o}{r_i}\left(\frac{r_m^2-r_i^2}{r_o^2-r_m^2}\right) \tag{2-50}$$

where r_i is the inner radius, r_o the outer radius, r_m is the radius of maximum velocity, τ_i and τ_o are the shear stress at the inner wall and at the outer wall, respectively.

Substituting Equation (2-50) into Equation (2-45) gives

$$\frac{\kappa_i}{\kappa_o}=\frac{\frac{r_o}{r_i}-\Omega}{\Omega-1}\left[\frac{\left(\frac{r_o}{r_i}\right)^2-\Omega^2}{\left(\frac{r_o}{r_i}\right)(\Omega^2-1)}\right]^{\frac{1}{2}} \tag{2-51}$$

where $\Omega=r_m/r_i$.

Similarly, the mixing length influenced by the fin tip and the outer wall is obtained by using the same method as described. The only difference is that the distance from the inner wall becomes that from the fin tip.

Now, the mixing length influenced by the fin side is obtain by using the analogy to a parallel plate

$$\frac{L_c}{s_o}=a_1-a_2\left(1-\frac{s}{s_o}\right)^2-a_3\left(1-\frac{s}{s_o}\right)^4 \tag{2-52}$$

where s_o is the half width between fins, $r\theta_o$ and θ_o the half angle between fins. As before, applying the conditions of (1) L_c=0 at s=0 and (2) $(dL_c/ds)=\kappa$ at s=0 results in

$$\begin{aligned} a_2&=2a_1-0.5\kappa \\ a_3&=0.5\kappa-a_1 \end{aligned} \tag{2-53}$$

So the value of a_1 needs to be determined. The value of a_1 cannot exceed 1 since the mixing length can not exceed the physical dimension. The results were found to be not sensitive to the value of a_1. The value of a_1=0.8 was determined to be an optimum value in comparison with finned tube data by Patankar et al. [20], and thus a_1=0.8, a_2=1.4 and a_3=-0.6 were used for given κ=0.4.

The y^+ and s^+ used in the van Driest functions are defined by

$$y^{+}=y\frac{\sqrt{\frac{\tau_w}{\rho}}}{\nu} \tag{2-54}$$

$$s^{+}=s\frac{\sqrt{\frac{\tau_w}{\rho}}}{\nu} \tag{2-55}$$

where y is the distance from the wall, s is circumferential distance $r\theta$ from the horizontal and s_o is half distance between fins $r\theta_o$.

The wall shear stress is determined for both inner and outer walls applying the respective velocity gradient at the wall using

$$\tau_w=\mu_l\sqrt{\left(\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial w}{\partial y}\right)^2} \tag{2-56}$$

So far the mixing length model has been established. However, as discussed in Section 2.3, the use of the mixing length theory has an inherent shortcoming that the turbulent eddy viscosity as defined by Equation (2-31) becomes zero at the location of maximum velocity since the velocity gradient becomes zero at this location, and thus the thermal diffusivity becomes zero as predicted by Equation (2-32). Although it is confined to a relatively small portion near the location of maximum velocity, it is considered unphysical based on the experimental deduction of μ_t by Lee and Park [7] and Reichardt [18]. If it is uncorrected, this will result in an unphysical kink in the temperature profile near the location of maximum velocity because of sudden reduction of the effective thermal viscosity, particularly for small values of Pr (0.7). However, the effects on the overall pressure drop and heat transfer rate were found to be small in terms of C_f and Nu.

To remedy the shortcoming, the Reichardt expression for the eddy viscosity given by Equation (2-44) was utilized. At $r=r_m$, Equation (2-44) reduces to

$$\mu_t=\rho\frac{\kappa_o}{6}(r_o-r_m)u_o^* \tag{2-57}$$

Using the definition for μ_t at $r=r_m$, Equation (2-31) reduces to

$$\mu_t=\rho l_m^2\left(\frac{\partial w}{\partial y}\right)_m \tag{2-58}$$

Equations (2-57) and (2-58) are equated to obtain $\partial w/\partial y$ at $r=r_m$ which produces the limiting low value of the eddy viscosity. The limiting $(\partial w/\partial y)_m$ value is used only for the purpose of obtaining a minimum value of the eddy viscosity near the location of maximum velocity. Further discussion and the results of using this remedy are given in Section 4.1.3.

2.5 Definition

The following definitions are used to reduce numerical results for comparison with available data.

The Reynolds number is defined by

$$Re=\frac{\rho W D_h}{\mu} \tag{2-59}$$

The friction coefficient is defined by

$$C_f=\frac{-\frac{dp}{dz}\frac{D_h}{2}}{\rho W^2} \tag{2-60}$$

The heat transfer coefficient is defined by

$$h=\frac{q}{T_w - T_b} \tag{2-61}$$

The Nusselt number is defined by

$$Nu=\frac{hD_h}{k} \tag{2-62}$$

The cross-sectional average velocity is defined by

$$W=\frac{1}{A_t}\int wdA \tag{2-63}$$

The bulk fluid temperature is defined by

$$T_b=\frac{\int wTdA}{\int wdA} \tag{2-64}$$

The average heat flux to the fluid is

$$q_{ave}=\rho C_p\ W\ \frac{A_f}{P_{ht}}\frac{dT}{dz} \tag{2-65}$$

CHAPTER 3

NUMERICAL PROCEDURE

3.1 Overview

The steady-state conservation equations for mass, momentum and energy, and the equations for turbulence models in two dimensions may be expressed in the following general form (e.g., [55]):

$$\frac{\partial}{\partial x}\left[K_1\frac{\partial\phi}{\partial x}+K_2\frac{\partial\phi}{\partial y}\right]+\frac{\partial}{\partial y}\left[K_3\frac{\partial\phi}{\partial x}+K_4\frac{\partial\phi}{\partial y}\right]+K_5\frac{\partial\phi}{\partial x}+K_6\frac{\partial\phi}{\partial y}+K_7\phi+K_8=0 \tag{3-1}$$

Equation (3-1) may be solved with specified boundary conditions. As shown in Figure 3.1, the boundary conditions on the boundary Γ of the domain may be in one or more of the following forms:

(1) essential (Dirichlet) boundary condition on Γ_1

$\phi = \overline{\Phi}$

(2) natural (Neumann) boundary condition on Γ_2

$q = \overline{q}$

(3) general boundary condition on Γ_3

$$q = \overline{q} + K_9(\phi - \overline{\Phi}) = (\overline{q} - K_9\overline{\Phi}) + K_9\phi = \overline{Q} + K_9\phi \tag{3-2}$$

where ϕ may be any one of the dependent variables such as velocity components, pressure, and fluid and solid temperatures, K_1 to K_7 may be constant material properties or coupled, nonlinear convection and diffusion coefficients, and K_8 is a source term. The constant, $K_{9,}$ is a specified value (e.g., heat transfer coefficient). The overbar variable denotes a specified boundary value. As can be seen in the finite element formulation, each finite element may have different terms and different values of K_i.

The finite element method was chosen for use in the present study instead of the finite difference counterpart. The main reason was to be able to model accurately an irregular geometry that contains curved and square boundaries such as a finned geometry. Although the finite element method has been used widely to study mean flow fields, its application to the turbulent boundary layer flows appears to have been limited to a simple tube (References 56-58). The present study applies the method to boundary layer flows in a complex geometry as in finned passages.

A variational approach is used in obtaining the finite element formulation of the governing equation with specified boundary conditions. The method used is the variational finite element model (Reference 55) in which the test function, w used for the variational formulation is the same as the approximation function used to represent a dependent variable, ϕ. The variational formulation results in an algebraic integral form of Equation (3.1) for each element. The integration of the integral is performed numerically on an element basis by transforming the physical coordinates (the global nodes) of an element to the master coordinates (the element nodes) through the Jacobian matrix. The element-based equations are combined with the adjacent elements by imposing the continuity at their interfaces. When this algebraic equation is applied to all elements, it results in a matrix of algebraic equations for all element nodes in the domain. The matrix for all unknown nodal values of each variable is then solved using a direct matrix solver.

The computer model used in this study is based on the finite element program which was used for solving one linear, two-dimensional, second-order, partial differential equation (Reference 55). In this study, a generalized procedure was implemented into the original program to be able to solve a number of coupled, nonlinear, partial differential equations. The classical variational finite element method used in the study is well reported in many finite element textbooks (References 59-62). The method used is briefly described for completeness, and the computational procedure for solving a number of nonlinear, coupled partial differential equations is given in detail.

3.2 Finite Element Formulation

The basic idea of the finite element method is to divide a given domain into a number of simple geometric shapes called finite elements. In the following the variational form of the governing equations with specified boundary conditions (Equations (3.1) and (3.2)) is derived over a typical element Ω_e. The resulting equation is applied to all elements, maintaining the continuity between the elements [59].

There are three major steps in the derivation of a variational finite element model:

(1) Take all non-zero terms to one side of the equality as done in Equation (3.1), multiply the resulting equation by a test function w, and integrate the resulting

equation over the domain of an element Ω_e.

(2) Reduce the second spatial derivative terms in Equation (3.1) using integration by parts (or Green's theorem) so that ϕ and w are differentiable only once with respect to x and y.

(3) Introduce the approximation and test functions into the variational form from step 2, and express the resulting equation in a matrix form.

Applying step 1 to Equation (3-1) leads to

$$\begin{aligned}&\int_{\Omega_e} w(x,y)[\frac{\partial}{\partial x}(K_1\frac{\partial\phi}{\partial x}+K_2\frac{\partial\phi}{\partial y})+\frac{\partial}{\partial y}(K_3\frac{\partial\phi}{\partial x}+K_4\frac{\partial\phi}{\partial y})\\&+K_5\frac{\partial\phi}{\partial x}+K_6\frac{\partial\phi}{\partial y}+K_7\phi+K_8]dxdy=0\end{aligned} \tag{3-3}$$

Following step 2, Green's theorem of Equation (3-4) (Reference 60) is applied to the terms containing second-order spatial derivatives in Equation (3.1).

$$\begin{aligned}&\int_\Omega \alpha\frac{\partial\beta}{\partial x}dxdy=-\int_\Omega\frac{\partial\alpha}{\partial x}\beta+\int_\Gamma\alpha\beta n_x d\Gamma\\&\int_\Omega \alpha\frac{\partial\beta}{\partial y}dxdy=-\int_\Omega\frac{\partial\alpha}{\partial y}\beta+\int_\Gamma\alpha\beta n_y d\Gamma\end{aligned} \tag{3-4}$$

Then Equation (3-3), including the specified boundary conditions, becomes

$$\begin{aligned}&\int_{\Omega_e}[\frac{\partial w}{\partial x}(K_1\frac{\partial \phi}{\partial x}+K_2\frac{\partial \phi}{\partial y})+\frac{\partial w}{\partial y}(K_3\frac{\partial \phi}{\partial x}+K_4\frac{\partial \phi}{\partial y})\\&+K_5 w\frac{\partial \phi}{\partial x}+K_6 w\frac{\partial \phi}{\partial y}+K_7 w\phi+K_8 w]dxdy\\&+\int_{\Gamma_{2e}} wq_n dxdy+\int_{\Gamma_{3e}} w\bar{Q}_n ds+\int_{\Gamma_{3e}} K_9 w\phi ds=0\end{aligned} \qquad (3\text{-}5)$$

where

$$\begin{aligned}&q_n=q_x n_x+q_y n_y\\&q_x=-(K_1\frac{\partial \phi}{\partial x}+K_2\frac{\partial \phi}{\partial y})\\&q_y=-(K_3\frac{\partial \phi}{\partial x}+K_4\frac{\partial \phi}{\partial y})\end{aligned} \qquad (3\text{-}6)$$

and n_x and n_y are the unit vectors in the x and y direction.

Equation (3-3) holds for any test function w. The use of Green's theorem reduced the order of the second-order terms and resulted in the natural boundary integral on Γ_{2e} (the third last term in Equation (3-5)). The last two terms of Equation (3-5) come from the specified boundary conditions. When the natural boundary integrals are summed over the adjacent elements, the net contribution becomes zero unless the physical boundary of the domain is encountered. Thus it is not necessary to evaluate such natural flux integrals when a portion of the element does not coincide with the physical boundary Γ of the domain Ω. The method of imposing essential boundary conditions is described in a later section.

Suppose that the dependent variable ϕ and the test function are approximated over a typical element Ω_e by

$$\begin{aligned}&\phi(x,y)\approx\phi^e(x,y)=\sum_{j=1}^{n}\phi_j^e\psi_j^e(x,y)\\&w(x,y)=\sum_{i=1}^{n}w_i{}^e\psi_i^e(x,y)\end{aligned}\qquad(3\text{-}7)$$

where $\phi_j^e(x,y)$ represents an approximation of $\phi(x,y)$ over the element Ω_e, ϕ_j^e and w_i^e are the values of functions ϕ^e and w^e at the element nodes i and j in the element Ω_e, and ψ^e are the approximation functions.

Finally following step 3, Equations (3-7) are substituted into Equation (3-5) to obtain the following algebraic equation

$$\begin{aligned}&\int_{\Omega_e}[\frac{\partial\psi_i^e}{\partial x}(K_1\sum_{j=1}^{n}\phi_j^e\frac{\partial\psi_j^e}{\partial x}+K_2\sum_{j=1}^{n}\phi_j^e\frac{\partial\psi_j^e}{\partial y})+\frac{\partial\psi_i^e}{\partial y}(K_3\sum_{j=1}^{n}\phi_j^e\frac{\partial\psi_j^e}{\partial x}\\&+K_4\sum_{j=1}^{n}\phi_j^e\frac{\partial\psi_j^e}{\partial y})+K_5\psi_i^e\sum_{j=1}^{n}\phi_j^e\frac{\partial\psi_j^e}{\partial x}+K_6\psi_i^e\sum_{j=1}^{n}\phi_j^e\frac{\partial\psi_j^e}{\partial y}\\&+K_7\psi_i^e\sum_{j=1}^{n}\phi_j^e\psi_j^e+K_8\psi_i^e]dxdy+\int_{\Gamma_{2e}}\psi_i^e q_n dxdy+\int_{\Gamma_{3e}}\psi_i^e\bar{Q}_n ds\\&+\int_{\Gamma_{3e}}K_9\psi_i^e\sum_{j=1}^{n}\phi_j^e\psi_j^e ds=0\end{aligned}\qquad(3\text{-}8)$$

Equation (3-8) may be written in a matrix form:

$$\sum_{j=1}^{n}K_{ij}^e\phi_j^e=F_i^e\qquad(3\text{-}9)$$

where

$$K_{ij}^{e}=\int_{\Omega_e}[\frac{\partial\psi_i^e}{\partial x}(K_1\frac{\partial\psi_j^e}{\partial x}+K_2\frac{\partial\psi_j^e}{\partial y})+\frac{\partial\psi_i^e}{\partial y}(K_3\frac{\partial\psi_j^e}{\partial x}+K_4\frac{\partial\psi_j^e}{\partial y}) \\ +K_5\psi_i^e\frac{\partial\psi_j^e}{\partial x}+K_6\psi_i^e\frac{\partial\psi_j^e}{\partial y}+K_7\psi_i^e\psi_j^e]dxdy+\int_{\Gamma_{3e}}K_9\psi_i^e\psi_j^e ds \tag{3-10}$$

$$F_i^e=\int_{\Omega_e}K_8\psi_i^e dxdy-\int_{\Gamma_{2e}}q_n\psi_i^e dxdy-\int_{\Gamma_{3e}}\bar{Q}_n\psi_i^e ds \tag{3-11}$$

3.3 Coordinate Transformation

The integrand in the square bracket in the coefficient integral of Equation (3-10) is given as a function of the global coordinates x and y. As shown in Figure 3.2, the global x-y coordinates are transformed to the master ξ-η coordinates only to facilitate numerical evaluation of the integrals. The integrand contains not only functions but also derivatives with respect to the global coordinates (x,y). Thus, the relationships between $(\partial\psi/\partial x$, $\partial\psi/\partial y)$ and $(\partial\psi/\partial\xi$, $\partial\psi/\partial\eta)$ are also needed. A quadrilateral element $\Omega_e(x,y)$ used in the present study is transformed to a master square element $-1\leq(\xi,\eta)\leq1$ in $\Omega_m(\xi,\eta)$.

The transformation between the global element $\Omega_e(x,y)$ and the master element $\Omega_m(\xi,\eta)$ is accomplished by a coordinate transformation of the form

$$x=\sum_{i=1}^{m} x_i^e \psi_i^e(\xi,\eta)$$
$$y=\sum_{i=1}^{m} y_i^e \psi_i^e(\xi,\eta) \tag{3-12}$$

The dependent variables of the problem are approximated by expressions of the form

$$\phi=\sum_{i=1}^{n} \phi_i^e \psi_i^e(\xi,\eta) \tag{3-13}$$

where ψ_i denotes the finite element approximation functions of the master element Ω_m. The functions ψ_i used for the approximation of the dependent variable (Equation (3-13)) may be different from that used in that of the geometry (Equation (3-12)). The present study uses the isoparametric formulation (m=n) where equal degree of approximation is used for both geometry and dependent variables. All quadrilateral elements (i.e., a four-sided element whose sides are not parallel), Ω_m in the *x-y* plane can be transformed to the same four-noded square master element, Ω_m in the ξ-η plane.

The approximation (also called interpolation) functions depend on the number of nodes in the element and on the shape of the element. The shape of element is such that its geometry is uniquely defined by a set of points, which serve as the element nodes in the development of the interpolation functions.

The approximation functions for a 4-noded quadrilateral element are

$$\begin{aligned}\psi_1&=\frac{1}{4}(1-\xi)(1-\eta)\\ \psi_2&=\frac{1}{4}(1+\xi)(1-\eta)\\ \psi_3&=\frac{1}{4}(1+\xi)(1+\eta)\\ \psi_4&=\frac{1}{4}(1-\xi)(1+\eta)\end{aligned} \tag{3-14}$$

The functions $\psi_i(x,y)$ can be expressed in terms of the local coordinates (ξ,η) by the chain rule of partial differentiation

$$\begin{aligned}\frac{\partial\psi_i^e}{\partial\xi}&=\frac{\partial\psi_i^e}{\partial x}\frac{\partial x}{\partial\xi}+\frac{\partial\psi_i^e}{\partial y}\frac{\partial y}{\partial\xi}\\ \frac{\partial\psi_i^e}{\partial\eta}&=\frac{\partial\psi_i^e}{\partial x}\frac{\partial x}{\partial\eta}+\frac{\partial\psi_i^e}{\partial y}\frac{\partial y}{\partial\eta}\end{aligned} \tag{3-15}$$

Equation (3-15) may be rewritten in a matrix form

$$\begin{Bmatrix}\dfrac{\partial\psi_i^e}{\partial\xi}\\ \dfrac{\partial\psi_i^e}{\partial\eta}\end{Bmatrix}=\begin{bmatrix}\dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi}\\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta}\end{bmatrix}\begin{Bmatrix}\dfrac{\partial\psi_i^e}{\partial x}\\ \dfrac{\partial\psi_i^e}{\partial y}\end{Bmatrix} \tag{3-16}$$

Equation (3-16) gives the relationships between the derivatives of ψ_i^e with respect to the global and local coordinates. The matrix in Equation (3-16) is called the Jacobian matrix

$$[J]=\begin{bmatrix}\dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi}\\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta}\end{bmatrix} \tag{3-17}$$

Thus, $(\partial\psi/\partial x, \partial\psi/\partial y)$ can be related to $(\partial\psi/\partial\xi, \partial\psi/\partial\eta)$ using Equation (3-16) by inverting the Jacobian matrix

$$\begin{Bmatrix} \frac{\partial\psi_i^e}{\partial x} \\ \frac{\partial\psi_i^e}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial\psi_i^e}{\partial\xi} \\ \frac{\partial\psi_i^e}{\partial\eta} \end{Bmatrix} \tag{3-18}$$

Equation (3-18) requires that the Jacobian matrix be non-singular i.e., its determinant being non-zero.

Using the transformation in Equation (3-12), it can be written

$$\frac{\partial x}{\partial\xi} = \sum_{i=1}^{m} x_i \frac{\partial\psi_i^e}{\partial\xi}, \quad \frac{\partial y}{\partial\xi} = \sum_{i=1}^{m} y_i \frac{\partial\psi_i^e}{\partial\xi} \tag{3-19}$$

$$\frac{\partial x}{\partial\eta} = \sum_{i=1}^{m} x_i \frac{\partial\psi_i^e}{\partial\eta}, \quad \frac{\partial y}{\partial\eta} = \sum_{i=1}^{m} y_i \frac{\partial\psi_i^e}{\partial\eta} \tag{3-20}$$

Given the global coordinates (x_i, y_i) of element nodes and the interpolation functions ψ_i^e, the Jacobian matrix can be evaluated using Equations (3-19) and (3-20).

The element area $dA=dxdy$ in element Ω_e is transformed to

$$dA = |J| d\xi d\eta \tag{3-21}$$

3.4 Numerical Integration over a Master Element

The transformation of the geometry and the variable coefficients of the differential equation from the global coordinates (x,y) to the local coordinates (ξ,η) resulted in algebraically complex integrals. They preclude analytical (exact) evaluation of the integrals. Thus numerical integration is used to evaluate such complicated integrals. The transformation enables numerical evaluation of integrals for a master element, $\Omega_m(\xi,\eta)$ over $-1 \le (\xi,\eta) \le 1$ using the Gauss-Legendre quadrature formula

$$\int_{\Omega_m} F(\xi,\eta)d\xi d\eta = \int_{-1}^{1}\int_{-1}^{1} F(\xi,\eta)d\xi d\eta \approx \sum_{I=1}^{M}\sum_{J=1}^{N} F(\xi_I,\eta_J)W_I W_J \tag{3-22}$$

where M and N denote the number of Gauss quadrature points, (ξ_I,η_J) denote the Gauss point coordinates (see Figure 3.3), and W_I and W_J denote the corresponding Gauss weights (Reference 59). In the present model, the interpolation functions are of the same degree in both ξ and η (i.e., M=N).

3.5 Assembly of Elements

The assembly of finite elements to obtain the equations of the entire domain is based on the following two rules (refer to Reference 62 for further details):

(1) continuity of the primary variable (e.g., temperature)

(2) balance of secondary variables (e.g., heat flux)

The correspondence between the local (the nodes of each element) and global (the nodes of the finite element mesh) nodal values imposes the continuity of the primary variables at the nodes along the interface between the two elements.

The flux from the two elements should be equal in magnitude and opposite in sign at the interface between the two elements. The balance of secondary variables are imposed by adding the two equations from the two elements at the common node.

When a number of elements are connected, the assembly of the elements is carried out by putting element coefficients into proper locations of the global coefficient matrix. This procedure is implemented in the computer program with the help of the connectivity relations, i.e., the correspondence of the local node number to the global node number.

3.6 Imposition of Essential Boundary Conditions

As discussed earlier, natural boundary conditions are imposed through the finite element formulation. Essential boundary conditions are imposed using the row-column elimination method (Reference 55). The method is implemented without altering the size of the global coefficient matrix and without rearranging the rows and columns.

The finite element formulation results in a system of linear equations in the form

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots & K_{1n} \\ K_{21} & K_{22} & K_{23} & \cdots & K_{2n} \\ K_{31} & K_{32} & K_{33} & \cdots & K_{3n} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ K_{nl} & K_{n2} & K_{n3} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ \cdot \\ \cdot \\ u_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ \cdot \\ \cdot \\ f_n \end{Bmatrix} \tag{3-23}$$

Suppose that the value of

$$u_2=\alpha \tag{3-24}$$

is a boundary condition to be imposed. The second equation in Equation (3-23) is replaced with Equation (3-24) to enforce the boundary condition for asymmetric coefficient matrices. For symmetric coefficient matrices, Equation (3-25) is further modified to Equation (3-26). This procedure enables us to retain the original order and symmetry of the coefficient matrix.

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots & K_{1n} \\ 0 & 1 & 0 & \cdots & 0 \\ K_{31} & K_{32} & K_{33} & \cdots & K_{3n} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ K_{n1} & K_{n2} & K_{n3} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ \cdot \\ \cdot \\ u_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ \alpha \\ f_3 \\ \cdot \\ \cdot \\ f_n \end{Bmatrix} \tag{3-25}$$

$$\begin{bmatrix} K_{11} & 0 & K_{13} & \cdots & K_{1n} \\ 0 & 1 & 0 & \cdots & 0 \\ K_{31} & 0 & K_{33} & \cdots & K_{3n} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ K_{n1} & 0 & K_{n3} & \cdots & K_{nn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \cdot \\ \cdot \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ \alpha \\ f_3 \\ \cdot \\ \cdot \\ f_n \end{bmatrix} - \begin{bmatrix} K_{12}\alpha \\ 0 \\ K_{32}\alpha \\ \cdot \\ \cdot \\ K_{n2}\alpha \end{bmatrix} \tag{3-26}$$

3.7 Grid Generation

In both finite-element and finite-difference methods, the accuracy of a solution depends on the fineness of the mesh and on the mesh distribution. The regions of large gradients need to be represented by small elements or a fine grid.

The present study followed the following general guidelines (Reference 62) for generation of a finite element mesh:

1. The mesh matches the flow and solid geometries of the computational domain accurately.
2. The mesh is such that large gradients in the solution (temperatures and velocities) are accurately represented.
3. The mesh does not contain elements with very large aspect ratios (i.e., ratio of the largest side to the smallest side of an element) or large angular distortions, especially in regions of large gradients.

A mesh generation program was developed and used to generate a mesh for a finned geometry considering the above guidelines. A structured mesh generation scheme is used for ease of numbering and positioning elements and element nodes. A gradient option is used to allow the gradual stretching or clustering of the nodes, particularly near the wall boundaries.

The essence of the present study is to be able to predict flow and temperature gradients near the wall. No wall function which is frequently used is employed. Instead, very fine nodes are introduced in a very short distance from the boundary.

Reynolds [63] notes that the large velocity gradient near the wall causes the large eddies to be broken up into small ones which dissipate the kinetic energy by the action of fluid viscosity. In order to achieve the observed steady rise in the dissipation rate, the small eddies must become even smaller as the Reynolds number increases. Thus it is expected that the thickness of the viscous sublayer will decrease as the Reynolds number rises, and that the turbulent activity will extend nearer to the wall.

The temperature profile becomes flatter with high Pr numbers. At a high Pr, the momentum boundary layer is thicker than the thermal boundary layer; at a low Pr number the thermal boundary layer is thicker. Thus, the region immediately adjacent to the heated wall should be shorter and finer. For an annulus or finned annulus geometry, the nodes in the vicinity of the point of maximum velocity should also be finer to capture the point.

Baker [64] noted that the accuracy of parabolic element solutions can be up to a factor of 50 improvement over the corresponding linear element results. Logarithmic wall elements (Taylor [65]) handles large velocity gradients near the wall.

3.8 Computational Procedure

The finite element formulation discussed in the preceding sections applies only to one equation with given coefficients K_i. A computational procedure is developed to perform a wide range of problems only through the changes of an input file and/or the designated subroutines defining nonlinear coefficients and source terms in Equation (3-1). Equation (3-1) representing all governing equations is solved sequentially one after another until all governing equations converge to a unique solution.

General Procedure

The following iterative procedure was chosen to obtain a solution for a set of nonlinear, coupled equations:

(1) A set of governing, coupled, nonlinear, partial differential equations to be solved is determined. Additional closure equations for a turbulence model can also be included.

(2) The "*K*" terms in Equation (3-1) are specified for all elements in every governing equation. The form or value of each "*K*" may be different in different regions of the solid or the fluid in each governing equation. They may be given as any combination of a constant, a functional form and a nonlinear function. When the form is given as a nonlinear function, the function for each term in each equation should be defined in the designated subroutines.

(3) The calculation domain is discretized into a number of finite elements using a quadrilateral element and the nodal coordinates and interconnections are input to the input file. The grid generation program discussed in the preceding section is used for this purpose.

(4) Each equation is solved at a time. When the next equation contains any term given as a function of the variables from the previous equations, the most recently calculated values are used. This method is used with a relaxation scheme to help

converge solutions of a set of coupled, nonlinear equations.

$$\phi_i=\lambda\phi_i^n+(1-\lambda)\phi_i^{n-1} \tag{3-27}$$

where ϕ_i is a dependent variable, n and *n-1* denote the value at current step *n* and previous step *n-1*, respectively, and λ is a relaxation factor which is specified in the input file and may be assigned a value between 0 and 1 (0.5 used throughout the simulation).

This procedure continues until all nonlinear coefficients are converged to the prescribed convergence criteria for all dependent variables.

$$\frac{\sum_{j=1}^{m}|\phi_i^n-\phi_i^{n-1}|}{m}\le R_i \tag{3-28}$$

where R_i is a prescribed tolerance value for each variable *i*, and *m* is the number of nodes.

Convergence

The variables used to solve turbulent diffusivities in Equations (2-16) and (2-17) are coupled and nonlinear. All nonlinear coefficients should converge to the prescribed convergence criteria.

The first iteration starts with laminar viscosity and obtain a velocity field. The velocity field is used to update all coupled variables $\mu_t(l, \partial w/\partial r)$, $k_t(\mu_t)$, $r_m(w)$, $\kappa_i(r_m)$, $l(\tau_w)$, $\tau_w(l, \partial w/\partial r)$. Subsequent iterations use the most current velocity profile to update all the variables. Iteration stops when the convergence criteria are met.

The values of R_i in Equation (3-28) used for both velocity (w) and temperature (T) are 5 (10^{-3}). The use of this tolerance value resulted in maximum relative residuals (i.e., $|\phi^n-\phi^{n-1}|/\phi^n$) of all nodes for all cases less than 4 (10^{-6}) and 3 (10^{-4}) for w and T, respectively. These residuals always decreased monotonically with iteration.

When a converged solution is obtained, all dependent variables at nodes and elements are available for a post analysis to obtain friction factor, wall shear stresses and local and average Nusselt numbers.

3.9 Matrix Solver

The finite element formulation of the governing differential equation leads to a system of linear equations with a coefficient matrix which is banded. If the coefficients for the first-order space derivatives (convective terms) in Equation (3-1) are zero (i.e., $K_5=K_6=0$), the banded matrix is symmetric, otherwise it is asymmetric. Both symmetric and asymmetric equation solvers are employed in the code for the solution of the system of linear equations. The choice of the solver is made in the code depending on the values of K_5 and $K_{6.}$ The solvers are based on a Gaussian elimination scheme. They use the banded properties of the coefficient matrix both in storage and computation.

Care has been exercised in numbering the nodal points so that a minimum band width can be obtained to reduce the computational time.

3.10 Verifications

Simulated results are always verified for their consistencies on the wall shear stress and pressure drop relationship and also for the heat balance. It has been noticed that when the near-wall grid was not adequate, the heat leaving the surface did not agree with the heat generated and the heat received by the fluid although the latter two agreed.

Wall Shear Stress

As shown in Equation (3-29), the average wall shear stress was obtained by integrating the calculated velocity gradients over all the surfaces. This value should satisfy the input pressure drop as in Equation (3-30). The difference was allowed to be less than 0.1%.

$$\tau_{w,avg}=\frac{1}{P_{wet}}\int_{walls}\mu_l\frac{\partial w}{\partial n}\delta s \qquad (3\text{-}29)$$

where n is the normal distance from the wall, and δs is the wall distance increment over which the velocity gradient takes place.

$$\tau_{w,avg}=-\frac{dp}{dz}\frac{D_h}{4} \qquad (3\text{-}30)$$

Heat Balance

The internal heat generation rate per unit volume, q_{gen} in the heater tube or the fuel is specified through the input.

$$Q=q_{gen}Vol \tag{3-31}$$

The heat transferred into the fluid is calculated using the simulated cross-sectional

$$Q=\rho C_p W\frac{dT}{dz}A_{flow}\delta z \tag{3-32}$$

average velocity W and the other input variables from Equation (3-32).

where δz is the axial heated length.

The heat leaving the heated surface is numerically integrated as in Equation (3-33):

$$Q=\int_{heated\ surface}-k_l\frac{\partial T}{\partial n}\delta s\delta z \tag{3-33}$$

To satisfy the heat balance, the heat generated must equal the heat leaving the heated wall and also the heat received by the fluid, i.e., all Q's from Equations (3-31) to (3-33) must equal. The difference was allowed to be less than 0.1%.

CHAPTER 4

RESULTS AND DISCUSSION OF SINGLE-PHASE ANALYSIS

A number of analytical and numerical solutions, and experimental data available in the literature for annuli and finned annuli were simulated to establish primarily:

(1) the accuracy of the finite element numerical procedure, and

(2) the validity of the turbulence model to the finned annulus geometry.

The objectives were met in steps. First, to establish the accuracy of the numerical model, the model simulated the large number of experimental and analytical data for smooth annuli in the literature. The model also simulated the analytical work of Patankar et al. [20] for finned annuli. Secondly, to support the validity of the turbulence model, AECL data [27] for a finned annulus geometry were used.

4.1 Analysis of Smooth Annuli

Before proceeding with predicting turbulent flow in the more complex geometry of a finned annulus, the present model was first applied to turbulent flow in a concentric annulus. The turbulent flow is more complex than its laminar flow counterpart since both Reynolds number and Prandtl number beccme parameters. Fully developed flow and temperature profiles with constant heat rate per unit length were predicted for a wide range of radius ratio r_o/r_i (1.6 to 80.7), Reynolds number (10^4 to 10^6) and Prandtl number (0.7 to 10). The validity of the solutions in annular passages is demonstrated by comparing with available experimental and analytical data for:

(1) eddy viscosity,

(2) location of maximum velocity,

(3) velocity profile,

(4) friction coefficient,

(5) temperature profile, and

(6) heat transfer rate.

4.1.1 Solution Procedure

The equations governing incompressible, fully developed, turbulent fluid flow and heat transfer using the eddy viscosity concept are given in Equations (2-16) and (2-17). These equations were solved to calculate detailed flow and temperature distributions and thus to determine the friction factors and Nusselt numbers for annuli.

The present turbulence model based on the mixing length approach is described in Chapter 2. The numerical procedure for solving Equations (2-16) and (2-17) is given in Chapter 3. In the present study, r_m and the sublayer thickness are calculated. Instead of using the wall function, a fine grid near both the inner and outer walls was used to obtain a solution.

4.1.2 Modelling

The annular geometry simulated here is schematically shown in Figure 4.1. The major assumptions and simplifications used are:

- the annulus is concentric. Both walls are smooth with the inner wall heated and the outer wall adiabatic,
- velocity and temperature profiles are fully developed,
- the mixing length model is used to obtain eddy diffusivities,
- the turbulent Prandtl number is given by a constant value Pr_t=0.9,
- the radial pressure gradient is negligible, and
- axial thermal conduction and eddy diffusion are negligible.

The von Kármán constant for the region outside the location of maximum velocity is κ_o=0.4 while that for the region inside the location of maximum velocity κ_i is expressed as a function of r_m which are determined numerically (see Section 2.4). The constants used in the van Driest model are A_i^+=A_o^+=26 everywhere.

The grid convergence test was performed by varying:

- the total number of elements,
- the number of near-wall elements,
- the aspect ratio (i.e., the ratio of the largest side to the smallest side of an

element), and

- the gradient of the wall nodes (defined as the ratio of grid size between the first and the last node in the near-wall region. The grid distances of the remaining elements are linearly incremented between the first and the last node).

The case chosen was the experimental case of Lee [6] which has r_o/r_i=1.632, Re=4.0(10^4), Pr=0.7 and T_i=48.44°C (The results are compared in Section 4.1.7). The reference grid is one stripe of 0.27° of the annulus. It has 69 quadrilateral elements and 140 nodes, consisting of 10 elements close to the inner wall, 10 elements close to the outer wall and 49 elements in the central flow region joining the two inner and outer regions (see Figure 4.2). Typical near-wall region of 10 elements used for Pr=7 is in the order of 0.1 mm whereas that used for Pr=0.7 is in the order of 1.0 mm. This grid was determined through a test by varying the total number of elements. Figures 4.3 and 4.4 show that the reference grid is adequate in view of negligible changes in the fluid temperature and velocity profiles by doubling and quadrupling uniformly the total number of elements of the reference grid.

A further grid test was performed using the reference grid. In Figures 4.5 to 4.8, "Fractional change" in the y axis is defined as test values divided by the reference value (i.e., $C_{f,test}/C_{f,ref}$ and Nu_{test}/Nu_{ref}). It is clear from Figure 4.5 that the number of wall elements is quite important to be able to capture sharp temperature and velocity gradients in the near-wall region. In addition, as shown in Figure 4.6, the degree of angular section which determines the size of element aspect ratio should be chosen in accordance with the size of radial grid. The gradient of wall nodes helped to capture sharp temperature and velocity gradients and reduced the number of wall nodes required to obtain an accurate solution (see Figure 4.7).

4.1.3 Eddy Diffusivity

The success of the mixing length model hinges on accurate prediction of the turbulent viscosities and thus the turbulent thermal diffusivities through the turbulent Prandtl number. The present analysis was compared with the turbulent viscosity profiles obtained experimentally by Lee and Park [7] and also with the analyses using other turbulent viscosity models.

Figure 4.9 presents comparison of the present analysis with the turbulent viscosity profiles obtained experimentally by Lee and Park [7]. The flat region predicted by the present theory is near the location of maximum velocity.

The experimental eddy viscosity distributions in Figure 4.9 were determined via velocity gradients (Equation (4-1)) from the velocity measurements (Reference 7) using:

$$\mu_{eff}=\frac{\frac{\tau}{\rho}}{\frac{dw}{dr}} \quad , \qquad (4\text{-}1)$$

where $\mu_{eff}=\mu_l+\mu_t$. The experimental results indicated that the eddy viscosity increases to a peak on both inside and outside the radius of maximum velocity and then decreases slightly near the radius of maximum velocity, but does not go to zero. The classical mixing length model based on Equation (2-31) will predict the turbulent eddy viscosity to approach zero at the location of maximum velocity since the velocity gradient becomes zero at the maximum velocity point. The present model avoids this situation by assuming a minimum turbulent viscosity near the location of maximum velocity (See Section 2.4). A nearly flat profile near the maximum velocity point is due to this modelling and a slight increase of μ_t towards this point is due to a slight increase of the mixing length. The present and measured profiles of the eddy viscosities agree quite well for all Reynolds numbers.

Figure 4.10 shows the comparison between the data of Lee and Park [7], and the analysis using the turbulent viscosities of Deissler [17] for the sublayer and of Reichardt [18] for the fully turbulent layer (Equations (2-43) and (2-44)). The constant, κ_i, was evaluated from Roberts' expression [3] (Equation (2-45)). For this analysis, the sublayer thickness was assumed to be at $y^+=26$ for both inner and outer regions. The numerically determined value of the location of maximum velocity was used. As seen in the figure, the turbulent viscosities using this model overestimated the data by up to about 40%.

Figure 4.11 presents the results of using r_m of Kays and Leung [1] rather than the numerically determined r_m of the present model. Although the r_m correlation given by Kays and Leung has been endorsed for a wide range of radius ratios (r_o/r_i up to 81) by Roberts [3], comparison between Figures 4.9 and 4.11 suggests that the present model using the numerically determined r_m yields more accurate prediction of the turbulent viscosities than using the Kays and Leung r_m correlation.

Figure 4.12 compares the performance of all four eddy viscosity models for $Re=1.1(10^5)$ with the data of Lee and Park [7]. The present model produced the best agreement with the data. It is noted that the present model using r_m of Kays and Leung produced a nearly identical profile for the region outside r_m to that for the present model using the numerically determined r_m since both models used von Kármán's constant κ_o=0.4 for this region.

4.1.4 Location of Maximum Velocity

A further difficulty with the case of turbulent annulus flow is that the position of zero shear and thus the wall stresses are not known a priori. Brighton and Jones [5] reported that the zero Reynolds stress (zero velocity gradient) and maximum velocity coincide within the accuracy of the experimental results.

The radius of maximum velocity for laminar flow, r_{ml} in an annulus is given by Lamb [66]:

$$\frac{r_{ml}}{r_i}=\left[\frac{\left(\frac{r_o}{r_i}\right)^2-1}{2\ln\left(\frac{r_o}{r_i}\right)}\right]^{\frac{1}{2}} \tag{4-2}$$

Kays and Leung [1] presented a correlation (Equation (4-3)) for the radius of maximum velocity for r_o/r_i less than 10.

$$\frac{r_{mt}}{r_i}=\frac{1+\left(\frac{r_o}{r_i}\right)^{0.657}}{1+\left(\frac{r_i}{r_o}\right)^{0.343}}=\Omega \qquad (4\text{-}3)$$

where r_{mt} is the radius of maximum velocity for turbulent flow in an annulus. Equation (4-3) is reported to be adequate for a wide range of r_o/r_i up to 80.7 (as confirmed by the experiments by Roberts [3] and Lee and Park [7]). Barrow et al. [4] also presented a correlation for the radius of maximum velocity which produced results similar to those from Equation (4-3) for r_o/r_i less than 10.

Figure 4.13 shows a comparison of the present simulations with experiments of Brighton and Jones [5] and Ivey [67] on r_m, and also the predicted values of r_m of Kays and Leung [1] and r_m for laminar flow. The agreement is quite good given the uncertainty of such measurement. For high r_i/r_o, the location of maximum velocity of turbulent flow converges to that of laminar flow. As shown in the figure, the location of maximum velocity for turbulent flow moves closer to the inner wall for a given annulus geometry in comparison with that for the laminar flow. Consequently, the ratio of the shear stress between the inner wall and the outer wall is less than that for laminar flow (see Equation (2-50)), indicating more contribution of the outer wall to the pressure loss for turbulent flow. The trend is similar to that of the Kays and Leung correlation given by Equation (4-3).

It was found from the present analysis that r_m did not change with Re for high Reynolds numbers ($>10^4$). This trend is consistent with Quarmby's measurements [11] that the change of maximum velocity radii ratio (r_{mo}/r_{mi}) was negligible at high Reynolds numbers ($>10^4$ for $r_o/r_i < 10$).

4.1.5 Velocity Profile

Fully developed annular flow involves the combination of two boundary layers, each extending from a wall to the point of maximum velocity. Unlike those that meet at the center of a pipe or midway between parallel planes, annular flow is quite different in velocity distribution, shear stress and turbulence quantities (Barrow et al. [4]).

The mechanism of the flow outside the radius of maximum velocity is similar to that occurring in circular pipe flow. This is not true for the flow inside the radius of maximum velocity. The standard universal velocity profile is not adequate for the turbulent velocity distribution inside the radius of maximum velocity.

Brighton and Jones [5] showed that velocity distributions near the outer wall fit the law of the wall for all radius ratios. The inner velocity profiles are in agreement with the law of the wall for high r_i/r_o ratios (0.562 and 0.375), but for lower r_i/r_o, u^+ is less than predicted by the law of the wall for y^+ greater than about 40, with the deviation increasing with decreasing r_i/r_o.

Figure 4.14 compares the velocity profiles outside the radius of maximum velocity of the present analysis with those in Reference 4. Agreement with the experimental data is quite good. The velocity profiles are essentially the same for a wide range of radius ratios (r_o/r_i from 1.632 to 80.72).

Figure 4.15 compares the velocity profiles inside the radius of maximum velocity of the present analysis with those in Reference 4. Agreement with the experimental data is quite reasonable. As indicated by Barrow et al. [4], agreement of the velocity distributions between the outer wall of the annulus and the pipe might be expected because the ratio of the boundary layer thickness to outer wall radius is often much less than that for the pipe and consequently the lateral curvature effects are reduced. However, inside r_m the velocity profile is affected by the inner wall curvature and depends on the annulus radius ratio.

Figure 4.16 compares the velocity profiles with the Brighton and Jones measurements [5]. The overall agreement of the profiles is quite good for both inside and outside the radius of maximum velocity for all Reynolds numbers and radius ratios. The radii of maximum velocity were also well predicted for all cases. As the ratio r_o/r_i was increased, the location of maximum velocity moved to the inner wall and made the velocity gradient steeper from the inner wall. Figure 4.17 compares the predicted velocity profiles outside r_m with the standard universal velocity profile of

$$u^+=\frac{1}{\kappa}\ln y^+ + B \qquad (4\text{-}4)$$

with κ=0.36 and B=3.8. As discussed before, the log-law profile is quite adequate to represent the velocity profile outside r_m for the wide range of Reynolds number and radius ratio.

4.1.6 Friction Coefficient

Figure 4.18 compares the friction coefficients with those of experimental data of Brighton and Jones [5] and those for smooth pipe flow. The smooth tube data were based on an empirical equation that fits the Kármán-Nikuradse equation [38]:

$$C_f=0.046Re^{-0.2} \tag{4-5}$$

As shown in the figure, the friction coefficients for flow in annuli with smooth walls are slightly higher (up to 10%) than those for pipe flow.

4.1.7 Temperature Profile

The temperature profiles were measured by Lee [6] for $r_o/r_i=1.632$ and $Re=4(10^4)$, and also for $r_o/r_i=2.584$ and $Re=2(10^4)$.

Predictions were made from the present model specifying the following conditions:

(1) -dp/dz which matches the measured mass flow rate,

(2) dT/dz which matches the measured constant heating rate,

(3) inner wall temperature, T_{wi},

(4) w=0 at both walls,

(5) dT/dr=0 at the outer wall, and

(6) $\partial w/\partial\theta=0$ and $\partial T/\partial\theta=0$ on the symmetry lines.

Fluid properties were evaluated at bulk temperature.

Figures 4.19 and 4.20 compare the predicted temperature profiles with the experimental data of air [6]. In Figure 4.19, the maximum difference between the measured and calculated temperatures was about 3°C when $(T_i-T_o)=29.2$°C. The predicted Nusselt numbers corresponding to the conditions in Figures 4.19 and 4.20 were calculated and found to be lower than the experimental Nusselt numbers by 5 and 8%, respectively.

4.1.8 Heat Transfer Rate

Figure 4.21 compares the predicted Nusselt numbers for $r_o/r_i=2$ and $Pr=0.7$ with the experiment and analysis of Kays and Leung [1]. Figure 4.22 compares the predicted Nusselt numbers for two Prandtl numbers of 0.7 and 10 with those of Kays and Leung [1]. All fluid properties were evaluated at bulk temperature. As shown in the figures, agreement is good.

4.1.9 Summary

In the present analysis of annuli, Nikuradse's mixing length relation for turbulent pipe flow [53] was applied to the region outside the location of maximum velocity. For the region inside the location of maximum velocity, the modified von Kármán constant of Roberts [3] was used. The van Driest damping function [19] was used to bridge the mixing length between the fully turbulent region and the viscous sublayer. The mixing length parameters that affect the velocity and temperature profiles are A^+, κ_i, κ_o and Pr_t. The values of the parameters used in the present study are A^+=26, κ_i expression by Roberts [3], κ_o=0.4, and Pr_t=0.9. A similar procedure was used earlier by Patankar et al. [20]. The main difference is that the present model chose to use the calculated radius of maximum velocity. The reason is that the use of the Kays and Leung r_m [1] was found to overestimate by up to 30% the eddy viscosities of the experiments [7].

Fully developed flow and temperature profiles were predicted for the wide range of radius ratio r_o/r_i (1.6 to 80.7), Reynolds number (10^4 to 10^6) and Prandtl number (0.7 to 10). The overall agreement between the present numerical results and data available in the literature for the annulus geometry is quite reasonable not only in terms of velocity and temperature profiles, but also friction coefficients and Nusselt numbers.

4.2 Comparison with Previous Finned Annuli Analysis

The present turbulence model based on the modified mixing length model is similar to that used in the Patankar et al. analysis [20]. Thus, the present simulation was compared with a finned annulus case previously analyzed by Patankar et al. However, as detailed in Section 2.4, it is noted that there are differences in the modelling approach as the present model used:

(1) The numerically determined r_m values (one r_m value along each radial grid line) rather than the single value used in Patankar et al. based on that of Kays and Leung. Thus, this difference influenced the variables such as wall shear stresses,

the von Kármán constant on the inner wall and the coefficients in the mixing length equations that depend on r_m,

(2) The limiting values of the turbulent viscosities near zero velocity gradient (i.e., near r_m) derived from Reichardt's expression rather than the value of zero at r_m, and

(3) The values of coefficients b_1, b_2 and b_3 in the mixing length equation for inside the location of maximum velocity (Equation (2-42)) given as a function of r_m/r_i and r_o/r_i rather than the constant values used in Patankar et al.

The analysis was performed for a case with the same conditions of Patankar et al.:

- $r_o/r_i=2$,
- 12 thin (zero thickness) fins attached to the inner wall of the annulus,
- $H/(r_o-r_i)= 0.4$ where H is the fin height,
- $Pr = 0.7$,
- uniform axial heating at the inner tube and fin walls and adiabatic at the outer tube wall,
- fluid properties evaluated at 20°C, and
- fully developed velocity and temperature profiles.

The governing equations are already given by Equations (2-16) and (2-17) in Section 2.2. The thermal boundary conditions used are $T=T_w$ along the fin height and around the inner tube circumference, and $\partial T/\partial\theta=0$ on the symmetry lines. The velocity boundary conditions are $w=0$ on the walls and $\partial w/\partial\theta=0$ on the symmetry lines.

The geometry and the grid used for the present simulation are given in Figures

4.23 and 4.24, respectively. The number of nodes and finite elements used are 1071 and 1000, respectively.

The results of the present computations are compared with those of Patankar et al. in Figures 4.25 and 4.26. The Nusselt number Nu and friction coefficient C_f are plotted against Reynolds number (based on the hydraulic diameter). The simulations were made in two ways using: (1) the numerically determined r_m and (2) Kays and Leung's r_m. The present analysis using the calculated r_m values simulated lower C_f and Nu than the analysis of Patankar et al. while the present model using the r_m value of Kays and Leung simulated higher C_f and Nu than the Patankar et al. analysis. The differences among the three predictions are reasonable in view of the differences in modelling approach mentioned earlier.

Figures 4.27 and 4.28 compare the local heat transfer coefficient distribution around the heated tube circumference and along the fin height between the present and Patankar et al. analyses. The local heat transfer coefficients (HTC) in Figures 4.27 and 4.28 were normalized by the area-averaged HTC over the tube and over the fin height, respectively. There are slight differences in the distributions between the two analyses. The difference near the fin tip is believed to be caused by a much finer grid before and after the fin tip used in the present model compared with that of Patankar et al. The present model using the Kays correlation shows a closer agreement since the analysis of Patankar et al. used the same correlation. It is noted that the heat transfer coefficient is highest at the tube center and reduces to zero at the corner between the tube and the fin base, and then increases towards the fin tip.

There are no experimental data for this geometry of zero-thickness fins. The present model using the calculated r_m was demonstrated to predict eddy viscosities closer to the experiments than that using the Kays and Leung r_m for an annulus (see Section 4.1.3).

4.3 Comparison with AECL Finned Annulus Data

4.3.1 AECL Single-Phase Experiments

Facility and Measurements

Figure 4.29 shows a schematic diagram of the test facility located in AECL-WL [29]. The vertical test section contains a finned pin placed inside a glass tube. The finned heater is constructed from a heater tube spray-coated with an electrical insulation of uniform aluminium oxide (Al_2O_3) layer of 0.1 mm, and clad with an aluminium sheath with 8 or 10 rectangular, longitudinal fins. The dimensions of the 8-fin pin geometry are given in Figure 4.30. Power is supplied at a uniform rate by passing current through the tube wall. Two glass tube diameters (17 and 24 mm ID) were used to study the effects of hydraulic diameters on the heat transfer characteristics.

The sheath and fin tip temperatures were measured at three locations along the heater using K-type thermocouples (see Figure 4.31): one measured on the sheath at the midpoint between two fins and another at the mid point at the fin tip. The fluid temperatures at the test section inlet and outlet were measured by resistance temperature detectors. The pressure drops in the test section were measured using differential pressure transmitters. The test section inlet and outlet pressures were measured using absolute pressure transmitters. A turbine flowmeter was used to measure the volumetric flow rate at the test section inlet. The power input was calculated from the voltage and current measured across the heater.

Figure 4.31 also shows the measurement locations of flow, fluid temperatures, wall temperatures, pressure and differential pressure. Appendix A, which reproduces the AECL data, gives velocity, bulk temperature and pressure at the measurement location in the downstream end (Section 3 in Figure 4.31), which is less than 50 mm from the heater outlet end. At this location, the value of pressure was actually measured locally using dp cells and the value of bulk temperature was calculated from an energy balance using inlet bulk temperature and power input. The value of velocity was calculated from the measured volumetric flow rate at the inlet. Appendix A also includes two wall temperature measurements for single phase flow. The inlet conditions and the power supplied to the heater were varied in the range of:

Power:	0 - 200 kW
Pressure:	100 - 300 kPa (abs)
Velocity:	0 - 6 m/s
Inlet Fluid Temperature:	15 - 100°C

AECL performed additional pressure drop measurements for single-phase water flow [30]: 0-7.5 m/s, a 440-mm length starting 80 mm from the test section inlet, isothermal conditions for an 8-fin pin in a 17-mm ID glass tube (D_h=7.3 mm). If a ratio of L/D of about 20 were required for the flow to be fully developed, about a 150-mm length or longer would be required for D_h=7.3 mm. Thus, the AECL Δp measurements are expected to be slightly higher than the Δp of a fully developed flow over the same length.

Detailed information on the instrument calibration, test procedure and an estimation of the measurement errors can be found in References 27 and 29.

Test Procedure

The test facility had four parameters which could be adjusted individually or in combinations: power, pressure, volumetric flow and inlet temperature. Experiments were performed by varying one of the four parameters and keeping the remaining three parameters constant.

Dissolved and trapped noncondensible gas was removed from the loop in the following procedure. The pump speed was cycled from 0 to maximum flow until there was no visible gas passing through the glass test section. The dissolved noncondensible gas was removed from the loop fluid by boiling the fluid in the surge tank which was vented to the atmosphere. This was accomplished by establishing a volumetric flow rate of 0.6 L/s and applying 20 kW of power to the test section. These conditions were maintained for one hour. Degassing was done each testing day prior to any experiment.

Both the AECL single-phase and ONB data used in this study were produced by increasing power or by increasing inlet temperature. When experiments were performed by increasing power, the volumetric flow, test section outlet pressure and test section inlet temperature were held constant. The power applied to the heater was increased gradually. A sufficient time lapse was allowed for steady-state conditions to be achieved. This procedure was repeated until sufficient single-phase data points were collected.

When experiments were performed by increasing inlet temperature, steady-state initial conditions of outlet pressure, flow and power were achieved. Then the heat exchanger secondary side cooling water flow was reduced such that the test section inlet water temperature was increased at approximately 2°C per minute.

Experimental Uncertainties

Reference 27 provides the accuracy of all measurements of fluid and surface temperatures, pressures, differential pressures, flow, current and voltage. The overall uncertainty with surface temperature measurements was estimated including the thermocouple error (1.1°C), fin effect (0.8°C), calibration (0.8°C) and mounting (0.8°C) and is reported to be within ±3.5°C in Reference 27. Two local surface temperatures were the main measured values used for comparison with the present analysis.

Additional uncertainties associated with the surface measurements may be caused by:

- a flow disturbance around the thermocouple junction. The junction is of a disk shape (with diameter of 0.01" and thickness of 0.005") and is embedded into the sheath. It was noted during the pre-tests that the direction in which the thermocouple junction faces influenced the reading and thus it was mounted to face against the flow,

- manufacturing tolerances on the dimensions of the fin and the sheath,
- eccentricity of an inner finned pin in the tube. This would cause subchannel flows to be distributed unevenly around the interfin regions, and
- nonuniformity of the heater thickness around the circumference. The circumferential variation of a heater thickness by manufacturer's tolerance would cause redistribution of the heat supplied, particularly for high power tests (see Section 4.3.2 for its sensitivity).

4.3.2 Analysis of AECL Single-Phase Finned Annulus Data

Validation is required to demonstrate the adequacy of the present models applied to the finned geometry, particularly:

1. The modelling choices such as the use of numerically determined r_m values (a number of r_m values evaluated along each radial grid line) and the use of limiting turbulent viscosities near zero velocity gradient based on Reichardt's expression (see Sections 2.4 and 4.1.3), and
2. The modified mixing length theory which takes into account the influences of both the tube and the fin walls. The value of coefficient a_1 in the mixing length equation (Equation (2-52)) for the fin side is adjusted.

The adequacy of item 1 was demonstrated extensively for the annulus geometry in Section 4.1. AECL experimental data were used for validation, particularly item 2.

AECL experiments were selected for simulation to demonstrate the effects of flow velocity, subcooling and heater power. The simulated wall temperatures are compared with the measured.

Modelling

The problem is already described in Section 2.1 along with the boundary conditions. As shown in Figure 4.32, there are three different regions in the domain: region 1 is the heater, region 2 is the sheath and the fin, and region 3 is the flow region. The governing energy equations and physical properties of each region are quite different. The governing momentum and energy equations for the flow region of fully developed velocity and temperature are given by Equations (2-16) and (2-17). The energy equation for the heater tube region is given by Equation (2-18). The energy equation for the sheath region including the fin is also given by Equation (2-18) but with no heat generation term q_{gen}. The continuity of temperature and heat flux was imposed at the interfaces between the regions.

The von Kármán constant for the outer wall is κ_o=0.4 while that for the inner wall κ_i is expressed as a function of r_m which are numerically determined (see Section 2.4). The constants used in the van Driest model are A^+=26 for both the inner and outer walls. The turbulent Prandtl number is given by a constant value of Pr_t=0.9.

The analysis was performed for an 8-fin heater in a 17-mm glass tube shown in Figure 4.32. Considering the symmetry of the geometry, a one-sixteenth part of the cross section of an 8-fin pin shown in Figure 4.32 is discretized into a grid of 1440 finite elements and 1519 nodes as shown in Figure 4.33. Instead of using the wall function, a number of fine nodes (10 nodes) were used for the near wall regions.

The input data for the program are as follows:

(1) -dp/dz which matches the measured mass flow rate,

(2) dT/dz which matches the measured constant heating rate,

(3) internal heat generation rate to the heater which matches the measured power to the test section,

(4) temperature at a selected node which gives the measured bulk fluid temperature,

(5) w=0 at both walls,

(6) dT/dr=0 at the outer wall, and

(7) $\partial w/\partial\theta=0$ and $\partial T/\partial\theta=0$ on the symmetry lines.

The following calculation procedure was used to simulate each single-phase experiment:

1. Make initial guesses of -dp/dz and μ_{eff} (using μ_l). Specify a temperature at a selected node.

2. Solve Equations (2-16) to (2-18) for the w and T fields, respectively.

3. Calculate $\dot{m}$ from average velocity W. Calculate bulk fluid temperature T_b from the T field.

4. Compare $\dot{m}$ and T_b in Step 3 with the experimental values. If deviation exceeds

1%, modify -dp/dz and the specified temperature. Repeat Steps 2 to 4 until deviation is within the tolerance.

Now velocity and temperature distributions were obtained as well as -dp/dz, T_{sh} and T_{ft} for the given $\dot{m}$, T_b and power.

Comparison with AECL Single-Phase Data

Figure 4.34 compares the predicted pressure gradient with the experimental value based on the measured pressure drop across the 440 mm test section at various flow rates. As discussed in Section 4.3.1, the AECL Δp measurements would exceed the Δp of a fully developed flow since some length of the 440-mm length would be in developing flow. Even though the present predictions for fully developed conditions exceed the measured values (which include a developing part), the deviations are acceptably small.

Figures 4.35, 4.36 and 4.37 compare the predicted surface temperatures with the two measured wall temperatures for flow velocities of 1.2, 2.0 and 4.0 m/s, respectively, at the sheath between fins and at the fin tip. It is noted that the effect of pressure on the heat transfer rates is shown to be negligible in the figures (except for few outliers in Figure 4.37). For the case of velocity of 1.2 m/s, the predicted temperatures are within the error bounds for a wide range of heat generation rate. For the cases of higher velocities of 2.0 and 4.0 m/s, the predicted wall temperatures both at the sheath and at the fin tip are higher than the measured temperatures. The overprediction of the wall temperatures at high flows and high powers corresponds to an underprediction of the heat transfer coefficients up to 15% by the present analysis (underprediction of the Nusselt number by the same amount) compared to the experiments.

Figure 4.38 includes the wall temperatures calculated using the heat transfer correlation of Stein and Begell [13] and assuming that the same amount of heat was transferred through the surface area of the inner wall of the annulus without the fins:

$$h=\frac{Nu\ k}{D_h}=\frac{0.02k}{D_h}Re^{0.8}Pr^{0.33}\left(\frac{r_o}{r_i}\right)^{0.5} \tag{4-6}$$

$$(T_w-T_b)=\frac{Q}{hA_{ht}} \tag{4-7}$$

where Q is the element power and A_{ht} is the element surface area. Equation (4-6) was developed for the range of r_o/r_i of 1.235-1.695, Re of $2.2(10^4)$-$3.0(10^5)$ and for water. For water flow inside a centrally heated annulus, Nixon [14] further supported this equation provided that the fluid properties are evaluated at $\frac{1}{2}(T_b+T_w)$ based on data of Pr of 2.0-8.5, Re of $6.0(10^4)$-$6.0(10^5)$, r_o/r_i of 1.33-2.45 and $D_h=5.3(10^{-3})$-$4.53(10^{-2})$ m. As shown in Figure 4.38, the presence of fins reduced the surface temperatures significantly.

Typical simulated results are shown in Figures 4.39 (temperature distribution) and 4.40 (heat transfer coefficient and heat flux distributions along the finned surface). Figure 4.39 shows that the surface temperature is the highest at the sheath center and decreases towards the fin tip. Figure 4.40 shows that the heat transfer rate decreases towards the fin corner from the sheath center and increases towards the fin tip reaching the peak at the fin tip edge. The distance on the x-axis is normalized by the total periphery between the sheath center and the fin tip center. The case is a simulation of test number 277 in Appendix A. The definitions of h_{ave} and q_{ave} are

$$q_{ave}=h_{ave}(T_{w,ave}-T_b) \tag{4-8}$$

$T_{w,ave}$, h_{ave} and q_{ave} are the area-averaged values over the entire inner surface including the sheath, fin side and fin tip.

As shown in Figure 4.40, the heat flux distribution is quite nonuniform along the inner periphery of the sheath, the fin side and the fin tip. The heat flux over the sheath decreases towards the corner of the fin base reaching a minimum at the corner, then increases along the fin side reaching its peak at the edge of fin tip, and then stays nearly uniform along the fin tip. Not only does the heat transfer coefficient increase along the fin height, but heat transfer itself increases. The heat transfer coefficient is shown to be higher over the fin tip than over the other inner periphery. This finding differs from the conventional assumption of negligible heat transfer through the fin tip. The temperature distribution shows its peak at the sheath center between fins, and decreases along the fin side reaching its minimum at the edge of the fin tip, and increases slightly along the fin tip.

A sample of the detailed flow and temperature distributions are shown in Figures 4.41 (iso-vels) and 4.42 (iso-therms), respectively. As shown in the figures, a sharp velocity gradient is concentrated near the first thin layer from the walls. The presence of fins pushes the velocity gradients towards the location of maximum velocity. However, the velocity profile outside the location of maximum velocity appears unperturbed by the fins. This led to a steeper velocity gradient at the fin tip than over the sheath, contributing to the increased heat transfer coefficient at the tip shown earlier. Similarly, as shown in Figure 4.42, most fluid temperature gradients take place within a very short distance from the heated wall. The presence of fins also affected the temperature distribution inside the sheath, resulting in lower temperatures at the root of the fin than over the sheath. This would cause an error in the conventional analysis of the fin effectiveness by assuming the uniform fin base temperature when heat is generated internally in the solid.

Sensitivity Analysis

A number of sensitivity cases were run using test number 277 in Appendix A.

Grid Convergence Test

The governing equations were approximated through the numerical integration using the finite element method. Thus, the grid was selected to obtain an accurate solution.

The rules for generating the grid discussed in Section 3.7 were used for determining the grid size. A systematic grid convergence test made with an annulus geometry in Section 4.1.2 was considered for node distribution and element aspect ratio. For the present study, since no pre-determined wall function is used, the near-wall region within the first 0.1 to 1-mm layer from the wall was represented by a fine grid of 10 nodes. This fine noding was applied to all surfaces of the inner sheath, the fin side, the fin tip and the outer tube. A fine grid was also applied in the circumferential direction not to contain elements of very large aspect ratios (i.e., the ratio of the largest side to the smallest side of an element).

As shown in the grid convergence test for an annulus in Section 4.1.2, the use of gradients for wall nodes in the range of 10 to 1000 had negligible changes. Modelling the near-wall region in the range of 0.1 to 1 mm also made negligible differences.

Effect of Mixing Length

Increasing the mixing length through increasing the value of a_t in Equation (2-52) (the reference value of 0.8 used throughout the study) to 1.0 increased the Nusselt numbers by 1% and reduced the sheath and fin tip temperatures by 1°C.

Effect of Turbulent Prandtl Number

As described in Section 2.4, the turbulent viscosity is defined as

$$Pr_t = \frac{\epsilon_M}{\epsilon_H} = \frac{\overline{w'v'}\dfrac{\partial T}{\partial y}}{\overline{T'v'}\dfrac{\partial w}{\partial y}} \tag{4-9}$$

This definition indicates that four quantities of turbulent shear stress, turbulent heat flux, velocity gradient and temperature gradient are needed to evaluate Pr_t. This is the reason why the scatter of experimental data tends to be large [52]. A survey of different models of the turbulent Prandtl number Pr_t by Kays [52] suggested the following based on experimental data for air:

$$Pr_t = \frac{1}{0.588 + 0.228(\epsilon_M/\nu) - 0.044(\epsilon_M/\nu)^2\left[1 - \exp(\frac{-5.165}{\epsilon_M/\nu})\right]} \qquad (4\text{-}10)$$

It provides a relatively high value of Pr_t near the wall but approaches 0.85 as ϵ_M/ν (thus y^+) increases. For Pr_t for water, Hollingsworth et al. [68] proposed

$$Pr_t = 1 + 0.855 - \tanh[0.2(y^+ - 7.5)] \qquad (4\text{-}11)$$

Pr_t in this equation also approaches 0.855 as y^+ is increased. The model did not change the wall temperatures compared to the reference value of 0.9. However, lowering the Pr_t value to 0.8 increased the Nusselt number by about 6% and reduced the sheath and fin tip temperatures by 5°C and 3°C, respectively.

Effect of r_m

As in the case of annulus, the use of r_m from Kays and Leung [1] increased the heat transfer rate and pressure drop and thus increased Nu and C_f. This was expected since, as shown in Section 4.1.3, the use of r_m of Kays and Leung overpredicted turbulent viscosities.

Although only one location of maximum velocity in the radial line was considered in the annulus geometry, it is conceivable that the presence of fins would move the location of r_m in a given radial line. Furthermore, the radii of maximum velocity is expected to vary in the circumferential direction for the finned geometry. These maximum velocity points are important as they are used as the integration end point approached from both inner and outer walls. The single location of maximum velocity may be considered reasonable for small fin heights since the maximum velocities in the circumferential direction would deviate little from that of the annulus geometry. However, the effect is expected to be significant for tall fins. Therefore, r_m's are obtained on every radial line and used as the integration points from both walls.

Effect of Physical Properties

The physical properties such as molecular viscosity, thermal conductivity, density and specific heat were evaluated based on local element temperature for all cases. Test number 277 for D_h=7.3 mm in Appendix A was simulated with the physical properties evaluated based on the bulk fluid temperature. The case based on the bulk fluid temperature reduced the Nusselt number by 20% and increased the sheath and fin tip temperatures by 21 and 17°C, respectively, compared to the reference case. It showed a significant effect especially when the wall and fluid temperatures are significantly different as in the present application.

Effect of Sheath Conductivity

The reference sheath conductivity of k_{sh}=220 W/(m·K) was varied from 10 W/(m·K) to ∞. The value of k_{sh}=10 W/(m·K) was considered to be a minimum value for commercial metals. The Nusselt number increased as the surface material conductivity increased. As expected, the case with $k_{sh}=\infty$ resulted in the sheath and fin temperatures being uniform at 132°C compared to the reference case at 157 and 119°C, respectively, and increased the Nusselt number by 7%. The case with k_{sh}=10 W/(m·K) increased the sheath and fin temperatures to 197 and 74°C, and reduced the Nusselt number by 21%.

Effect of Heat Generation Rate and Heat Split

Adding more power to the heater had very little effect on the Nusselt numbers for the same flow conditions. Figure 4.43 shows that the temperature distribution along the finned surface becomes more nonuniform for higher flows. However, the increase of heat generation rate changed the temperature distribution unnoticeably.

Next examined was nonuniform heat splitting for the same power. The reference case of uniform heat generation rate was compared with cases of nonuniform heat generation rates up to 14% and 50% tilt. The heat is split into: (1) Q_1 for the first 12° from the horizontal and (2) Q_2 for the next 10.5° angular heater segment of the modelled 22.5° segment. The second segment received less heat than the first segment by 14% and 50%. For example, the 14% tilt is defined as

$Q = Q_1 + Q_2$, before split (4-25)

$Q = (1.14Q_1) + (Q_2\text{-}0.14Q_1) = Q_1' + Q_2'$, after split.

Since the sheath thermal conductivity k_{sh}=220 W/(m·K) is high, the effects on the Nusselt number and the wall temperatures were negligible. Therefore, the uncertainty caused by the nonuniform heater thickness discussed in Section 4.3.1 is negligible.

4.4 Study of Geometric Effects of Fins

There are ways of evaluating enhancement in heat transfer of internal finning. Reference 70 provides practical consideration of performance evaluation criteria for enhanced heat transfer surfaces. Patankar et al. [20] used comparison of the ratio of the fin heat load Q_{fin} to the total heat load Q_t both per unit length with the ratio of the fin area A_{fin} to the total heat transfer area A_t. The criterion $Q_{fin}/Q_t > A_{fin}/A_t$ was used to indicate that on a unit area basis the fins are a more effective heat transfer surface than the tube wall.

4.4.1 Calculation Procedure and Input

In the present study, the performance of internal finning is evaluated for the following conditions:

(1) constant average flow velocity (W), and

(2) constant mass flow rate ($\dot{m}$).

These conditions were chosen to facilitate the comparison of heat transfer rate and pressure drop for internally finned annuli with respect to the annulus geometry of the same r_i and r_o. The use of constant average flow velocity is equivalent to the use of constant Re, when Re is evaluated using D_h of an unfinned annulus. Velocity and mass flow rate were chosen to give Re=10^4 and 10^5 of an unfinned annulus.

Table 4.1: Cases simulated for parametric study

Cases simulated		Number of fins	Relative fin height, $H/(r_o-r_i)$
Constant W	6 m/s 0.6 m/s	8	0 to 0.5
		12	0 to 0.5
		16	0 to 0.5
Constant $\dot{m}$	1.1 kg/s 0.11 kg/s	8	0 to 0.5
		16	0 to 0.5

Figure 4.44 shows three simulated geometries of N=8, 12 and 16 with $H/(r_o-r_i)=0.22$. As shown in Table 4.1, the following procedure was used in determining the geometric effects:

(1) Choose the basis for comparison as either constant W or constant $\dot{m}$,

(2) Vary only one condition at a time: (a) fin height for a given number of fins or (b) the number of fins for a given fin height,

(3) Specify the heat generation rate defined in the heater tube. A fully conjugated

problem is solved in which nonuniform heat flux and temperature distributions are taken into account in the overall performance of fins,

(4) Specify the fluid properties based on T_b, and

(5) Obtain a solution that gives the flow conditions - constant W (or constant $\dot{m}$) and T_b.

The reference geometry used is the AECL 8-fin geometry (Figure 4.30) having:

r_i=3.935(10^{-3}) m,

r_o=8.5(10^{-3}) m and

fin width=0.76(10^{-3}) m.

Constant water properties at 50°C were used, namely:

ρ=988 kg/m^3,

μ_f=5.471(10^{4}) Pa·s,

k_f=0.64 W/(m·K), and

C_p=4181 J/(kg·K).

The heat generation rates used are:

q_{gen}=2.979(10^9) W/m^3 for high flow simulations (Constant velocity of 6 m/s and constant mass flow of 1.1 kg/s), and

q_{gen}=5.958(10^8) W/m^3 for low flow simulations (Constant velocity of 0.6 m/s and constant mass flow of 0.11 kg/s).

4.4.2 Effects of Fin Geometry

Figures 4.45 to 4.46 show predicted pressure drops based on constant velocity and constant mass flow rate. Pressure drop increased with the increase of fin height or number of fins for a given mass flow rate (or velocity). The higher fin height required the larger driving force, -dp/dz due to additional resistance by increased fin height. Similarly, for a given mass flow rate (or velocity), the more number of fins required the larger driving force due to increased flow resistance by more fins. Figures 4.45 to 4.46 also show that, for a given fin geometry, pressure drop for the constant mass flow rate is higher than that for the constant velocity. More fins reduce the flow area and thus increases the velocity for the constant mass flow rate cases.

The effect of the presence of fins on the flow distribution is shown in Figures 4.47 and 4.48. In these figures, $\dot{m}_l$ denotes the mass flow passing through the annulus bounded by y=0 and y=H and $\dot{m}_t$ denotes the total mass flow. The fact $(\dot{m}_l/\dot{m}_t)<(A_{fin}/A_t)$ indicates that more of the flow passes through the unfinned area in order to avoid the higher resistance in the interfin spaces. The comparison between Figures 4.47 and 4.48 also show that more of the total flow passes through the interfin regions for higher flow velocity (or higher mass flow rate). The decrease of $\dot{m}_l/\dot{m}_t$ with increasing number of fins indicates that more of the flow passes through the unfinned area to avoid higher resistance in the interfin spaces. The change of $\dot{m}_l/\dot{m}_t$ was negligible whether the constant $\dot{m}$ or constant W is used.

Figures 4.49 and 4.50 show that the fins are more effective than the annulus as

$Q_{fin}/Q_t > A_{fin}/A_t$ for nearly all cases. However, as shown in Figure 4.50, an exception can be seen in tall fins ($H/(r_o-r_i)=0.5$) for the 8-fin geometry such that fins became slightly less effective for the high flow case. As discussed before, this is because more of the total flow passes through the interfin regions for high flows, resulting in more of the total heat leaving through the sheath. Figures 4.49 and 4.50 also show that, for a given fin height, increasing number of fins increased the heat transfer effectiveness in terms of $(Q_{fin}/Q_t)/(A_{fin}/A_t)$, particularly for low flows. The change in Q_{fin}/Q_t was negligible whether the constant $\dot{m}$ or constant W is used.

Figures 4.51 and 4.52 show that, for a given heat generation rate in the heater, the average heated wall temperature $T_{w,ave}$ decreased with increasing fin height or with increasing number of fins. These figures show that $(Q/L)/(T_{w,ave}-T_b)$ was higher for the constant mass flow case than that for the constant velocity case. The reduction in flow area due to fins increased flow velocity for the constant mass flow rate case, and thus reduced $T_{w,ave}$ for a given heat generation rate.

Figures 4.53 and 4.54 show the effect of fin geometry on wall temperature (T_{sh}) for the high and low flows, respectively. As discussed before, both $T_{w,ave}$ and T_{sh} decreased with increasing fin height or number of fins. The comparison of Figures 4.53 and 4.54 show that the ratio $(T_{sh}-T_b)/(T_{w,ave}-T_b)$ is much higher for the high flow case than the low flow case. It indicates that, for the low flow case, the difference between T_{sh} and $T_{w,ave}$ became smaller (T_{sh} is slightly higher than $T_{w,ave}$) as Q_{fin}/Q_t becomes higher for the low flow case than the high flow case. For the same reason, the ratio $(T_{sh}-T_b)/(T_{w,ave}-T_b)$ is higher with the constant mass flow case as shown in Figure 4.54. From the comparison of

Figures 4.53 and 4.54, the difference in $(T_{sh}-T_b)/(T_{w,ave}-T_b)$ between the 8-fin and 16-fin geometry is more noticeable for the high flow case with negligible difference for the low flow case.

CHAPTER 5

RESULTS AND DISCUSSION OF ONSET OF NUCLEATE BOILING ANALYSIS

5.1 Analysis of AECL Finned Annulus ONB Data

The analysis is now extended to predict the onset of nucleate boiling in a finned annulus and to study the geometric effects of fin height and number of fins.

5.1.1 AECL ONB Experiments in Finned Annuli

The ONB data reported in an AECL report [27] are reproduced in Appendix A. The data were collected for three different geometries:

(1) D_h = 7.3 mm - 8-fin element in a 17 mm ID glass tube,

(2) D_h =13.7 mm - 8-fin element in a 24 mm ID glass tube, and

(3) D_h = 5.4 mm - 10-fin element in a 17 mm ID glass tube.

The data in Appendix A give the conditions at the point of ONB occurrence: the power, fluid velocity, pressure, bulk fluid temperature, and the sheath and fin tip temperatures. The data were already processed from the raw data. A detailed procedure used for reducing the data can be found in Reference 27.

Measurements

The AECL facility in which the ONB tests were conducted is the same as that used for the single-phase tests, and is described in Section 4.3.1. A schematic diagram of the test facility is given in Figure 4.29 and a diagram showing the instrumentation locations is given in Figure 4.31.

The ONB is defined as the point where vapour bubbles first appear and become visually observable on the heated surface. The finned surface is illuminated by a stroboscopic light source to enhance the detection of the small vapour bubbles. An eight-power telescope was used for visual observations of the heated surface. To confirm visually observed ONB points, the ONB was determined for some selected tests from the change in the slope of the surface temperature with respect to power. At high velocities, the change in the temperature slope was not as well defined. Thus, for consistency the visually observed ONB was used in determining all the ONB data points.

Dissolved and trapped noncondensible gas was removed from the loop in each test. The procedure is described in Section 4.3.1.

The ONB data were obtained by increasing power or increasing inlet fluid temperature. The power was incremented to a new value and the loop was allowed to achieve a steady state, while maintaining the test section outlet pressure, volumetric flow and inlet temperature at their initial values. This procedure was repeated until the first bubbles appear on the finned surface. The power level and surface temperatures were then recorded. When experiments were performed by increasing inlet temperature, steady-state initial conditions of pressure, flow and power were achieved. Then the heat exchanger secondary side cooling water flow was reduced such that the test section inlet water temperature was increased at approximately 2°C per minute.

The ONB point was controlled to take place at the downstream end of the test section as shown in Figure 4.31 where the surface temperatures were measured. This point is 50 mm below the test section end. This location was chosen as flow is expected to be fully developed and heat losses by the axial conduction were found negligible. At some power level, the conditions permitted vapour bubbles to form on the heated surface at some of the nucleation sites. As noted in Reference 27, the first bubbles always appeared on the sheath between two fins. At the ONB the vapour bubbles did not detach from the surface. The surface temperature at the ONB was several degrees above the fluid saturation temperature.

AECL also performed a photographic study [30] to measure the cavity sizes on the sheath surface using magnification factors up to 7500. Although the range of cavity sizes was not obtained, an elliptic shape of a cavity of about 2 µm by 6 µm was shown.

Experimental Uncertainties

The experimental uncertainties described in Section 4.3.1 are also applicable here.

There are additional uncertainties associated with the visual determination of the ONB point or the graphical determination of the ONB point (by change in slope of wall temperature).

5.1.2 Comparison with AECL Finned Annulus ONB Data

Modelling

The grid, input parameters and assumptions used for the ONB predictions are identical to those used for the single-phase predictions. It is described in Section 4.3.2. To predict the superheat and heat flux required at the point of ONB, a number of single-phase predictions are made at various heat generation rates for given mass flow, bulk fluid temperature and pressure at the point of ONB. Fully developed flow and temperature conditions are assumed. Power is supplied to the heater tube inside the sheath and the outer tube wall is adiabatic. The simulations provide a curve of superheat versus heat flux. When the Davis and Anderson criterion [36] is applied, this curve is used to find the intersection point with the ONB criterion that defines the ONB point. When the Hsu criterion [33] is applied, the simulation provides the thermal layer thickness which is obtained from the temperature profile at the point of ONB.

The von Kármán constant for the outer wall is κ_o=0.4 while that for the inner wall κ_i is expressed as a function of r_m, which is numerically determined (see Section 2.4). The constants used in the van Driest model are A^+=26 for both the inner and outer walls. A constant value of the turbulent Prandtl number Pr_t=0.9 was used.

The analysis was performed for the finned annulus that consists of an 8-fin heated pin in a 17-mm glass tube shown in Figure 4.32. The grid used is the same as in Section 4.3.2. Fluid properties such as viscosity, thermal conductivity and density were evaluated at the local fluid temperature. All ONB data obtained with the finned annulus geometry of D_h=7.3 mm were analyzed.

ONB Analysis with Hsu's Model

Detailed flow and temperature profiles are predicted from the present model for the finned annulus geometry. The predicted temperature and heat flux distributions are used in conjunction with the ONB criteria of Hsu [33] and Davis and Anderson [36] to determine the ONB. The parametric trend and the magnitudes of heat flux and superheat required for the ONB were predicted and are compared in the following section.

The Hsu theory [33] is based on a bubble nucleus at a site surrounded by a warm liquid. As shown in Figure 5.1, the nucleus begins to grow into a bubble only when the surrounding liquid is sufficiently superheated. The time required for the liquid to attain this superheat is called the waiting period. The transfer of heat from the superheated liquid into the bubble is considered to be a transient conduction process. A cavity is considered effective only if the waiting period is finite. He derived the effective cavity sizes by equating the bubble temperature obtained from the Clausius-Clapeyron and the surface tension equations with the liquid temperature profile obtained from the transient conduction equation as

$$r_{c,\max}=\frac{\delta}{2C}\left[1-\frac{\theta_{sat}}{\theta_w}+\sqrt{\left(1-\frac{\theta_{sat}}{\theta_w}\right)^2-\frac{4AC}{\delta\theta_w}}\right] \tag{5-1}$$

$$r_{c,\min}=\frac{\delta}{2C}\left[1-\frac{\theta_{sat}}{\theta_w}-\sqrt{\left(1-\frac{\theta_{sat}}{\theta_w}\right)^2-\frac{4AC}{\delta\theta_w}}\right] \tag{5-2}$$

where $r_{c,max}$ and $r_{c,min}$ are the maximum and minimum sizes of effective cavities, respectively, $\theta_{sat}=T_{sat}-T_\infty$, $\theta_{sat}=T_{sat}-T_\infty$, $\theta_w=T_w-T_\infty$, $A=2\sigma T_{sat}/(\lambda\rho_v)$, $C=1+\cos\phi$, ϕ is the angle of bubble surface with respect to the horizontal, δ is the limiting thermal layer thickness, σ is surface tension of liquid with respect to its vapour, and λ is the latent heat of vaporization. These equations give the maximum and minimum sizes of effective cavities as a function of subcooling, pressure, physical properties and the thickness of the superheated layer. The superheat required for the ONB, θ_{wo} ($=T_{wo}-T_\infty$), was derived from Equations (5-1) and (5-2) as no cavity will be effective if the discriminant of these equations is negative as

$$\theta_{wo}=\theta_{sat}+\frac{2AC}{\delta}+\sqrt{\left(2\theta_{sat}+\frac{2AC}{\delta}\right)\left(\frac{2AC}{\delta}\right)} \tag{5-3}$$

This equation indicates that there is no sustained boiling existing if $\theta_w < \theta_{wo}$.

The most important parameters in the Hsu criterion are the thermal layer thickness δ and the bulk temperature T_∞. He assumed that there exists a limiting thermal layer δ that for $y<\delta$ molecular transport prevails, while for $y\geq\delta$ the temperature remains at bulk temperature T_∞. His definition of δ is similar to the laminar sublayer thickness. The thermal layer thickness δ depends on the geometry, Re and Pr, and bubble disturbance (bubble size, bubble growth rate). At and up to the ONB, turbulence would be of primary influence on δ for a given Pr. However, these definitions are rather ambiguous and are difficult to determine. To achieve T_∞, it would take much farther distance from the wall than thermal layer thickness δ.

Although some difficulties were encountered in applying the Hsu criterion, the present study assumed that the limiting thermal layer δ is the first layer of a constant temperature slope from the wall and was determined from the temperature profile at the ONB. This is consistent with the validity of a conduction equation in his derivation to obtain the liquid temperature profiles. The bulk fluid temperature was used in place of T_∞ in the model.

For turbulent flows in a finned annulus (with the inner surface heated and the outer surface insulated), the fluid temperature profile has a very sharp gradient immediately near the inner wall and the slope is quite flat and changes very little towards the outer wall. Figure 5.2 shows temperatures in the near-wall region of the temperatures along the mid-sheath radial line (the 22.5° line). There is a layer of a constant-slope temperature profile in the first layer from the wall and that its thickness decreased as Re number increased for a given Prandtl number. The calculated δ values tabulated in Table 5.1 were used to obtain $T_{w,onb}$ with the Hsu criterion (Equation (5-3)) for the analysis of AECL data.

In the evaluation of Equation (5-3), the following values were used:

- the bubble contact angle with respect to the horizontal ϕ=90° corresponding to a hemispherical sphere,
- the latent heat of vaporization λ, vapour density ρ_v and saturation temperature T_{sat} evaluated at saturation for a given pressure,
- the surface tension of liquid with respect to vapour σ evaluated at T_{sat}.

Table 5.1 Comparison of calculated and measured $T_{w,onb}$ for the finned annulus geometry of D_h=7.3 mm

Case	Re	δ, m	Calculated $T_{w,onb}$, °C using δ from the temperature profile	Calculated $T_{w,onb}$, °C using constant δ=2(10^{-4}) m	Measured $T_{w,onb}$, °C
6	9060	4.0(10^{-5})	133	126	128
10	16743	2.4(10^{-5})	137	127	127
11	20755	2.2(10^{-5})	138	127	128
15	28668	2.0(10^{-5})	139	127	129
17	43323	1.3(10^{-5})	144	128	131
20	56399	1.2(10^{-5})	146	129	131
23	68900	5.0(10^{-6})	161	130	130

Although all finned annulus ONB data for D_h=7.3 mm were analyzed, Table 5.1 presents the results of varying flow velocity for fixed bulk fluid temperature of 56°C and pressure of 0.2 MPa. As shown in Table 5.1, agreement is poor between the measured and calculated $T_{w,onb}$ when the thermal layer thickness was obtained from the temperature profile. The calculated ONB temperatures are higher and the disagreement is more noticeable at higher Reynolds numbers. As indicated in the table, the use of a constant value $\delta=2(10^{-4})$ m improved the agreement and brought the overall agreement of all 25 ONB data for the finned annulus geometry of $D_h=7.3$ mm within ±3°C. The calculated thicknesses based on the constant temperature slope near the wall are an order of magnitude smaller than the value of $\delta=2(10^{-4})$ m. Although the use of the constant value δ improved the agreement, it is difficult to justify it since the transient one-dimensional conduction equation applied in Hsu's derivation is valid only within the constant slope part of the temperature profile.

The effective cavity sizes for the ONB data calculated from Equations (5-1) and (5-2) ranged from 1.2 to 28 μm. The cavity size of 2 by 6 μm measured by AECL is within the calculated range.

A sensitivity study of input parameters on $T_{w,onb}$ was made and shows that:

(1) $T_{w,onb}$ decreased by 5-17°C by increasing the contact angle 30 to 90° (the range of typical contact angles for commercial metal surfaces [69]), and the changes are more for higher Re numbers,

(2) the evaluation of surface tension at T_{sh} rather than at T_{sat} reduced $T_{w,onb}$ very little, viz., of the order of 0.1°C (surface tension increases with reducing temperature).

It would be of interest to extend the Hsu model by using the transient fluid energy equation rather than the conduction equation to obtain the waiting period. Although the present model can be used for this purpose, it was not tried because of anticipated long CPU time involved.

ONB Analysis with Davis and Anderson's Model

As in Hsu's model, the basic assumptions of Davis and Anderson [36] are:

1. The bubble nucleus grows at a surface cavity and has the shape of a truncated sphere as shown in Figure 5.3,
2. The equilibrium theory (the Clausius-Clapeyron equation) can be used to predict the superheat required to satisfy a force balance on the bubble (the Gibbs equation for surface tension),
3. A bubble nucleus will grow if the liquid temperature at a distance from the wall equal to the bubble height is greater than the superheat required for bubble equilibrium,
4. The bubble nucleus does not alter the temperature profile in the fluid surrounding it.

Davis and Anderson [36] derived the superheat equation required for a stable bubble using the Gibbs equation for the pressure difference across a bubble, the Clausius-Clapeyron equation and the ideal gas law. Then they equated the slope of the superheat equation with the temperature profile at the wall, and solved for the critical distance from the wall required to initiate nucleate boiling provided that cavities of the size corresponding to this critical distance exist. The Davis and Anderson ONB criterion [36] is given by:

$$T_w - T_{sat} = \frac{\frac{RT_{sat}^2}{\lambda}\ln(1+\xi)}{1-\frac{RT_{sat}}{\lambda}\ln(1+\xi)} + q_{wi}\frac{y}{k_l} \tag{5-4}$$

$$y = \frac{C\sigma}{p_l} + \sqrt{(\frac{C\sigma}{p_l})^2 + \frac{2Ck_l S}{q_w}} \tag{5-5}$$

$$\xi = \frac{2\sigma C}{p_{sat} y} \tag{5-6}$$

$$C = 1 + \cos\phi \tag{5-7}$$

where q_{wi} is ONB heat flux, R is gas constant, k_l is liquid thermal conductivity, and ϕ is the angle of bubble surface with respect to the horizontal. The condition, C=1, corresponds to a hemispherical bubble nucleus being used.

For the present application, the following equation was found to give a good approximation to Equation (5-4):

$$q_{wi} = \frac{k_l \lambda \rho_v}{8C\sigma T_{sat}}\left(T_w - T_{sat}\right)^2 \tag{5-8}$$

This equation is usually equated with the following equation to obtain $T_{w,onb}$ and thus q_{wi}:

$$q=h(T_w-T_b) \tag{5-9}$$

However, for the finned annulus geometry the heat transfer coefficient and the wall temperature are not known and vary around the finned periphery. Thus it is difficult to obtain the heat flux q_{wi} and superheat $(T_{w,onb}-T_{sat})$ required to cause the ONB. Therefore, the present single-phase predictions of heat fluxes and wall temperatures are used in conjunction with the Davis and Anderson criterion to predict the ONB.

The values used to evaluate the Davis and Anderson criterion are:

- ϕ=90° corresponding to a hemisphere,
- the latent heat of vaporization λ and vapour density ρ_v evaluated at T_{sat} corresponding to the pressure at the plane of ONB,
- the surface tension of liquid with respect to vapour σ evaluated at T_{sat}.

The present model simulated the temperature and heat flux distributions. As illustrated in Figure 5.4, the ONB point is defined as the intersection of the calculated temperature and heat flux at the sheath midway between the fins with those of the Davis and Anderson criterion, Equation (5-4). The model supplied the successive calculations of superheat at various powers until the ONB point, i.e., the intersection point, was found.

The following calculation procedure was used to determine the ONB point using the Davis and Anderson criterion:

1. Establish a relationship (a graph) between mass flow rate and -dp/dz by pre-simulating a number of isothermal cases for a given finned annulus geometry,
2. Determine volumetric heat generation rate (element power/heater tube volume),

dT/dz (from the energy balance using mass flow and C_p) for each ONB datum point,

3. Specify a temperature boundary condition at a selected node in the calculation domain,

4. Select a corresponding value of -dp/dz for a given mass flow rate from the graph of step 1, and refine -dp/dz, if necessary, based on the relationship of $\Delta p \propto \dot{m}^2$,

5. Iterate simulations until the calculated T_b agrees with the experimental T_b by adjusting the temperature boundary condition of step 3,

6. Run the single-phase model with fixed mass flow rate and pressure but with variable power to compare with the Davis and Anderson criterion (Equation (5.4)). Repeat step 5 for each power until the calculated T_b converges to the experimental T_b. Plot superheat (T_w-T_{sat}) versus heat flux at the sheath to obtain T_{sh} and power at the intersection with the Davis and Anderson criterion. An example is shown in Figure 5.4.

The above procedure was repeated for all 25 ONB data points for the 8-finned element in a 17-mm glass tube (D_h=7.3 mm).

Figure 5.5 shows comparison of the experimental and calculated ONB powers with flow velocity. The agreement appears quite good except for a few high velocity points. For these data points, the detection of ONB occurred up to 15% higher power than the calculated ONB. The measured wall temperatures are also lower than the calculated temperatures. As bubble size gets smaller with increasing flow, it is possible that visual observation may have missed the first bubble until a higher power. The actual wall temperature would have dropped due to improved boiling heat transfer rate.

Figure 5.6 shows comparison of the experimental and calculated ONB powers with subcooling. The calculated ONB powers are about 6% less than the experimental values.

Figure 5.7 shows comparison of the experimental and calculated ONB powers with pressure. The experimental data followed the expected parametric trends except for one outlier at W=4.1 m/s (shown in Figure 5.7).

The arithmetic mean deviation ($\bar{e}$) and the root-mean-square deviation (RMS) for all 25 ONB data points are -10% and 13%, respectively, where

$$e=\frac{Predicted\ value-Experimental\ value}{Experimental\ value} \qquad (5\text{-}10)$$

$$\bar{e}=\frac{\sum_{j=1}^{m} e_j}{m} \qquad (5\text{-}11)$$

$$RMS=\sqrt{\frac{\sum_{j=1}^{m} e_j^2}{m}} \qquad (5\text{-}12)$$

The predicted ONB powers were consistently less than the measured ONB powers.

The predicted ONB shown in Figures 5.5 to 5.7 follow the parametric trends observed from the finned annulus ONB results such that the power at ONB increased with increasing:

- flow velocity, or

- ΔT_{sub}, or
- pressure.

The present model together with the Davis and Anderson ONB criterion predicted slightly less powers and higher superheat (T_w-T_{sat}) at the ONB than the experimental values. To improve the agreement, the model should calculate higher pressure losses (higher C_f) and thus higher Nu than the current values for a given dp/dz.

A sensitivity study of input parameters on the ONB heat flux was made. As illustrated in Figure 5.8, the ONB heat flux decreased slightly by increasing the contact angle from 30° (C=1.866) to 90° (C=1). Although the use of smaller contact angle thus brought the agreement with the experiments closer at high flows, this may not be physically plausible since a higher flow would make the superheat layer in which bubbles occupy thinner and suppress the bubbles, thus increasing the contact angle (90° as in a hemisphere).

The evaluation of surface tension at T_{sh} rather than at T_{sat} reduced $T_{w,onb}$ very little in the order of 0.1°C (surface tension increases with reducing temperature). Sensitivity cases were also run with such variables as Pr_t (the reference value of Pr_t=0.9) and a_t (the reference value of 0.8 used to calculate the mixing length in the circumferential direction). The use of Pr_t=0.8 rather than the reference value reduced T_{sh} and T_{ft} by a few degrees Celsius. The use of a_t=1.0 rather than the reference value reduced T_{sh} and T_{ft} by about a degree Celsius.

5.2 Parametric Study of Fins

In the present study, the performance of internal finning is evaluated for:

(1) constant average flow velocity (W), and

(2) constant mass flow rate ($\dot{m}$).

The geometric and flow parameters in Table 4.1 were used. Details of the procedure and input is given in Section 4.4.1.

As detailed in the previous section, the ONB heat flux was determined at the intersection between successive single-phase predictions with those of Davis and Anderson criterion. The heat flux was defined by:

$$q''=\frac{Heater\ power}{Area\ of\ unfinned\ annulus\ r_i} \qquad (5\text{-}13)$$

This definition was used to facilitate the comparison of heat transfer rates for various internally finned annuli with respect to the annulus geometry of the same r_i and r_o.

Figures 5.9 and 5.10 show the ONB heat fluxes by varying fin height and number of fins for constant mass flow rate and constant velocity. The figures show that the ONB heat flux increased with increasing fin height or number of fins. The comparison between Figures 5.9 and 5.10 shows that the increase of the ONB heat flux is more pronounced with low flows.

5.3 Summary

The present analysis with Hsu's model did not predict well the experimental ONB when the limiting thermal layer thickness was evaluated as the first layer of a constant temperature slope from the wall. His model posed some difficulties of defining consistently the limiting thermal layer thickness and the bulk fluid temperature. Thus, the present study recommends and used Davis and Anderson's criterion.

The present model with the Davis and Anderson ONB criterion predicted consistently less power and higher superheat (T_w-T_{sat}) at the ONB than the experimental values. The sensitivity studies indicated that the agreement can be improved by: lowering Pr_t, or increasing a_1, or reducing the bubble contact angle particularly at high flows. However, the optimization of these parameters was deemed unnecessary since the predictions were quite good for low flows (less than about 4 m/s) and the discrepancy at high flows may have been caused by some uncertainties in visually obtaining the ONB points particularly at high flows and high pressures. Overall, the present prediction with Davis and Anderson's ONB criterion predicted quite well the experimental finned annulus ONB data except for a few high flow data. The predicted ONB results followed the parametric trends for the finned annulus ONB data such that the ONB power increased with increasing flow velocity, ΔT_{sub}, or pressure.

Both the experiment and the present prediction of ONB indicated that the highest wall temperature occurred at the sheath midway between two fins and is consistent with the ONB occurring there. The parametric study showed that the ONB heat flux increased with fin height and number of fins. The increase of the ONB heat flux is more pronounced with low flows.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

A study was made of turbulent fluid flow and heat transfer in finned passages. The governing conservation equations of momentum and energy were formulated with a turbulence closure model based on the classical mixing length theory that has been widely used for the tube and annulus geometries. The mixing length model was modified for a finned geometry so that a mixing length at a point can be determined by superimposing the contributions from its surrounding surfaces. The governing equations were solved using a finite element method to obtain detailed velocity and temperature distributions in finned annuli.

A number of coupled, nonlinear heat transfer and fluid flow problems in annuli and finned annuli have been simulated to establish primarily: (1) the accuracy of the finite element numerical procedure and (2) the validity of the turbulence model applied to the finned annulus geometry. The objectives were met in steps. First, to establish the accuracy of the numerical model, the model simulated the large number of experimental and analytical data for annuli in the literature. The model also simulated analytical work of Patankar et al. [20]. Secondly, to support the validity of the turbulence model, AECL data [27] for the finned annuli geometries were used. The model was then applied to predict the onset of nucleate boiling in the finned annuli and to study the geometric effects of fin height and the number of fins.

For the analysis of annuli, fully developed flow and temperature profiles were predicted for a wide range of radius ratio (r_o/r_i of 1.6 to 80.7), Reynolds number (10^4 to 10^6) and Prandtl number (0.7 to 10). The overall agreement between the present numerical results and data available in the literature is quite good in terms of eddy viscosities, location of maximum velocity, velocity profiles, friction coefficients, temperature profiles and Nusselt numbers.

For the analysis of finned annuli, the same geometric and flow conditions used by Patankar et al. were applied in the present model. The present analysis reproduced closely the local heat transfer coefficient distribution around the heated tube circumference and along the fin height, as well as the Nusselt numbers and friction coefficients.

For further analysis of finned annuli, the present model simulated AECL experiments and the results were compared with the two measured local surface temperatures: at the sheath between fins and at the fin tip. The predicted temperatures are in good agreement for low flows for a wide range of heat generation rate. For high flows, the predicted wall temperatures both at the sheath and at the fin tip are higher than the measured temperatures. This would mean the underprediction of the heat transfer rates up to 15% by the present analysis. The presence of fins caused significantly nonuniform distributions of heat transfer coefficient and temperature along the finned surface. The heat transfer coefficient was predicted to be higher over the fin tip than over the other inner periphery. The predicted wall temperature peaked at the sheath center between fins, and decreased along the fin side reaching its minimum at the edge of the fin tip.

The geometric effects of fins were also investigated by varying fin height and the number of fins for constant mass flow rates and constant flow velocities. Heat transfer in finned annuli is generally more effective than that in the unfinned annuli, particularly for low flows. However, an exception was found that a small number of tall fins (H/(r_o-r_i)=0.5 with 8 fins) is not as effective as the unfinned annuli for high flows. Pressure drop increased with the increase of fin height or number of fins for a given mass flow rate (or a given flow velocity).

The analysis was extended to predict the ONB on the internally finned annuli. The heat flux and superheat required for the ONB were predicted in conjunction with the criteria of Hsu [33], and of Davis and Anderson [36]. Hsu's criterion presented some difficulties in obtaining the thermal boundary layer thickness and the liquid temperature at the boundary. Thus, the Davis and Anderson criterion is recommended and was used. For the finned annuli, the present analysis provides the essential input to the criteria such as the thermal boundary layer thickness for the Hsu criterion and the local heat flux and superheat at various powers for the Davis and Anderson criterion.

The predicted ONB results with the Davis and Anderson criterion showed good agreement with the internally finned annulus data of AECL except for few high flow conditions. Possible reasons for the disagreement are discussed. Both the measured and predicted ONB occurred at the sheath midway between fins. The predicted ONB followed the parametric trends of the measured data such that the ONB power increased with increasing flow velocity, ΔT_{sub}, or pressure.

The ONB heat fluxes were also studied by varying fin height and number of fins for constant mass flow rate and constant velocity. The finned annuli were found to delay the ONB to higher powers than the unfinned annulus counterpart for the same flow conditions. The ONB heat flux increased with fin height and number of fins. The increase of the ONB heat flux is more pronounced with low flows.

Based on the study, it is concluded:

- Overall, agreement of the present analyses with available experiments and other previously published analyses for both annuli and finned annuli geometries seems quite reasonable,
- The classical mixing length theory frequently used for the annuli can be applied to the finned annuli with the use of few modelling improvements: (1) a superposition method for the mixing length to take into account the influence of all walls, (2) the numerically determined locations of maximum velocity locations, (3) Reichardt's expression to remedy zero shear near the maximum velocity, and (4) temperature-dependent physical properties. No adjustment was necessary for the finned annuli to the generally accepted values of the parameters $A^+=26$, κ_i of Roberts [3], $\kappa_o=0.4$ and $Pr_t=0.9$ used for the annuli,
- The present model provides a practical means to solve for a fully conjugate problem of assessing the pressure drop and heat transfer characteristics in finned passages. The actual geometry of fins and the heating conditions can be modelled

accurately.

6.2 Recommendations

Provisions made in the computer model to enable extension of the present work include:

(1) up to 10 different partial differential equations can be solved simultaneously. Each equation can consist of a number of terms such as a transient term, convection/diffusion terms and source terms. This will allow one to solve for the full two-dimensional Navier-Stokes equations (developing flow, transient flow), for two-phase flow equations or for higher-order turbulence equations.

(2) higher order elements such as eight- and nine-noded quadrilaterals can be tried to obtain a better accuracy with a smaller number of nodes.

(3) the model is limited to a two-dimensional problem. The model can be extended to solve three-dimensional problems with the addition of three-dimensional element shape functions.

(4) the present grid generation program was designed to generate structured grids for the known geometries simulated (annulus, finned annulus). It provides the element nodal coordinates and element/node connections. The model may receive this input from other commercial grid generation programs for a unstructured grid to further optimize the grid.

Detailed surface and fluid temperatures, velocity and turbulence measurements will be useful. Comprehensive experimental data for Δp and temperature for different finned geometries will further support the use of the present model for finned geometries.

REFERENCES

1. Kays, W.M. and Leung, E.Y., Heat Transfer in Annular Passages - Hydrodynamically Developed Turbulent Flow with Arbitrarily Prescribed Heat Flux, Int. J. Heat Mass Transfer, v. 6, pp. 537-557, 1963.

2. Lee, Y., Turbulent Heat Transfer from the Core Tube in Thermal Entrance Regions of Concentric Annuli, Int. J. Heat Mass Transfer, v. 11, pp. 509-522, 1967.

3. Roberts, A., A Comment on the Turbulent Flow Velocity Profile in a Concentric Annulus, Int. J. Heat Mass Transfer, v. 10, pp. 709-712, 1967.

4. Barrow, H., Lee Y. and Roberts, A., The Similarity Hypothesis Applied to Turbulent Flow in an Annulus, Int. J. Heat Mass Transfer, v. 8, pp. 1499-1505, 1965.

5. Brighton, J.A. and Jones, J.B., Fully Developed Turbulent Flow in Annuli, J. Basic Engineering, pp. 835-844, December 1964.

6. Lee, Y., Turbulent Flow and Heat Transfer in Concentric and Eccentric Annuli, Ph.D. Thesis, The University of Liverpool, 1964.

7. Lee, Y. and Park, S.D., Developing Turbulent Flow in Concentric Annuli: An Analytical and Experimental Study, Warme- und Stoffubertragung, v. 4, pp. 156-166, 1971.

8. Shigechi, T., Kawae, N and Lee, Y., Turbulent Fluid Flow and Heat Transfer in Concentric Annuli with Moving Core, Int. J. Heat Mass Transfer, v.33, pp. 2029-2037, 1990.

9. Lee, Y. and Kim, K.C., An Analysis on Effect of Transverse Convex Curvature on Turbulent Flow and Heat Transfer, Warme- und Stoffubertragung, v. 28, pp. 89-95, 1993.

10. Wilson, N.W. and Medwell, J.O., An Analysis of Heat Transfer for Fully Developed Turbulent Flow in Concentric Annuli, J. Heat Transfer, pp. 43-50, February 1968.

11. Quarmby, A., An Analysis of Turbulent Flow in Concentric Annuli, Appl. Sci. Res., v. 19, pp. 250-273, July 1968.

12. Rothfus, R.R., Monrad, C.C., Sikchi, K.G. and Heideger, W.J., Isothermal Skin Friction in Flow through Annular Sections, Industrial and Engineering Chemistry, v. 47, pp. 913-918, May 1955.

13. Stein, R.P. and Begell, W., Heat Transfer to Water in Turbulent Flow in Internally Heated Annuli, A.I.Ch.E. Journal, v. 4, p. 127, 1958.

14. Nixon, M.L., Heat Transfer to Water Flowing Turbulently in Tubes and Annuli, AECL Internal Report, CRNL-165, 1968.

15. Hasan, A., Roy, R.P. and Kalra, S.P., Velocity and Temperature Fields in Turbulent Liquid Flow through a Vertical Concentric Annular Channel, v. 35, pp. 1455-1467, 1992.

16. Prandtl, L, Report on Investigation into Developed Turbulence (Translated by D.B. Spalding), Zeitschrift für angewandten Mathematik und Mechanik, v. 5, no. 2, pp. 136-139, April 1925.

17. Deissler, R.G., Turbulent Heat Transfer and Friction in the Entrance Regions of Smooth Passages, Trans. ASME, v. 88, pp. 1221-1233, 1955.

18. Reichardt, H., Vollständige Darstellung der turbulenten Geschwindigkeitsverteilung in glatten Leitungen, Zeitschrift für angewandte Mathematik und Mechanik, v. 31, pp. 208-219, 1951.

19. van Driest, E.R., On Turbulent Flow near a Wall, J. Aero. Sci., v.23, pp. 1007-1011, 1956.

20. Patankar, S.V., Ivanovic, M. and Sparrow, E.M., Analysis of Turbulent Flow and Heat Transfer in Internally Finned Tubes and Annuli, Journal of Heat Transfer, v.101, pp. 29-37, 1979.

21. Hornby, R.P., Mistry, J. and Barrow, H., A Mixing Length Model for Turbulent Flow in Constant Cross Section Duct, Warme- und Stoffubertragung, v. 10, pp. 125-129, 1977.

22. Trupp, A.C., Lau, A.C.Y., Said, M.N.A. and Soliman, H.M., Turbulent Flow Characteristics in an Internally Finned Tube, Advances in Heat Transfer, ASME, HTD-18, pp. 11-19, 1981.

23. Trupp, A.C., Lau, A.C.Y., Said, M.N.A. and Soliman, H.M., Turbulent Flow Characteristics in an Internally Finned Tube, Joint ASME/JSME Conference, pp. 423-428, 1983.

24. Edwards, D.P., Hirsa, A. and Jensen, M.K., An Experimental Investigation of Flow in Longitudinally Finned Tubes, ASME Winter Annual Meeting, Anaheim, CA, 8-13 November 1992.

25. Said, M.N.A. and A.C. Trupp, Prediction of Turbulent Flow and Heat Transfer in Internally Finned Tubes, Chem. Eng. Commun., v.31, pp. 65-99, 1984.

26. Kim, N.H. and Webb, R.L., Analytic Prediction of the Friction and Heat Transfer for Turbulent Flow in Axial Internal Fin Tubes, J. Heat Transfer, v. 115, pp. 553-559, August 1993.

27. Hembroff, R.L., Kowalski, J.E., Spitz, K.O. and McCallum, C.K., Single-Phase and Boiling Heat Transfer Measurements Conducted in the MAPLE-X10 Heat Transfer Test Facility, MX-10-03300-232-TN, Unpublished AECL Technical Note, September 1991.

28. De Lorenzo, B. and Anderson, E.D., Heat Transfer and Pressure Drop of Liquids in Double-Pipe Fin-Tube Exchangers, Transactions of the ASME, pp. 697-702, November 1945.

29. Spitz, K.O., Kowalski, J.E., Hembroff, R.L. and Baxter, D.K., MAPLE Heat Transfer Test Facility, MX10-03300-208-TN, Unpublished AECL Technical Note, September 1989.

30. Hembroff, R.L., Single-Phase ΔP Measurements and Finned Surface Photographic Study Conducted in the Single-Pin Heat Transfer Test Facility, Unpublished AECL Data, 20 November 1995.

31. Kowalski, J.E., Lim, I.C., Shim, S.Y. and Mills, P.J., Heat Flux Distribution on a Finned Fuel Pin, Proc. of 15th CNS Annual Nuclear Simulation Symposium, Toronto, 1-2 May 1989.

32. Ivanovic, M., Prediction of Flow and Heat Transfer in Internally Finned Tubes, Ph.D. Thesis, The University of Minnesota, July 1978.

33. Hsu, Y.Y., On the Size Range of Active Nucleation Cavities on a Heating Surface, J. Heat Transfer, pp. 207-216, August 1962.

34. Han, C.Y. and Griffith, P., The Mechanism of Heat Transfer in Nucleate Pool Boiling - Part I, Bubble Initiation, Growth and Departure, Int. J. Heat Mass Transfer, v. 8, pp. 887-904, 1965.

35. Bergles, A.E. and Rohsenow, W.M., The Determination of Forced-Convection Surface-Boiling Heat Transfer, J. Heat Transfer, pp. 365-372, August 1964.

36. Davis, E.J. and Anderson, G.H, The Incipience of Nucleate Boiling in Forced Convection Flow, A.I.Ch.E. Journal, pp. 774-780, July 1966.

37. Buleev, N.I., Theoretical Model of the Mechanism of Turbulent Exchange in Fluid Flows, AERE-Trans 957, May 1963.

38. Kays, W.M. and Crawford, M.E., Convective Heat and Mass Transfer, Second Edition, McGraw Hill, New York, 1980.

39. Pletcher, R.H., Progress in Turbulent Forced Convection, J. Heat Transfer, v. 110, pp. 1129-1144, November 1988.

40. Benocci, C., Modelling of Turbulent Heat Transport A State-of-the-Art, Technical Memorandum 47, von Kármán Institute for Fluid Dynamics, Chaussée de Waterloo, 72 B-1640 Rhode Saint Genèse, Belgium, April 1991.

41. Spalding, D.B., Turbulence Models; a Lecture Course, CFD/82/4, Imperial College of Science and Technology, London, UK, October 1982.

42. Rodi, W., Turbulence Models and Their Application in Hydraulics - A State of the Art Review, International Association for Hydraulic Research, 1980.

43. Harlow, F.H. and Nakayama, P.I., Transport of Turbulence Energy Decay Rate, Los Alamos Scientific Laboratory Report LA-3854, Los Alamos, NM, 1968.

44. Jones, W.P. and Launder, B.E., The Prediction of Laminarization with a Two-Equation Model of Turbulence, Int. J. Heat Mass Transfer, v. 15, pp. 301-314, 1972.

45. Launder, B.E., On the Computation of Convective Heat Transfer in Complex Turbulent Flows, J. Heat Transfer, v. 110, pp. 1112-1128, November 1988.

46. Herrero, J. et al., A Near-Wall k-ϵ Formulation for High Prandtl number Heat Transfer, Int. J. Heat Mass Transfer, v. 34, no. 3, pp. 711-721, 1991.

47. Launder, B.E., Numerical Computation of Convective Heat Transfer in Complex Turbulent Flows: Time to Abandon Wall Functions?, Int. J. Heat Mass Transfer, v. 27, no. 9, pp. 1485-1491, 1984.

48. Launder, B.E., Reece, G.J. and Rodi, W., Progress in the Development of a Reynolds Stress Closure, J. Fluid Mechanics, v. 68, pp. 537-566, 1975.

49. Nallasamy, M., Turbulence Models and Their Applications to the Prediction of Internal Flows: A Review, Computers and Fluids, v. 15, pp. 151-194, 1987.

50. Patel, V.C., Rodi, W. and Scheuerer, M., Turbulence Models for Near-Wall and Low Reynolds Number Flows: A Review, AIAA J., v. 23, pp. 1308-1319, 1985.

51. Rodi, W., A New Algebraic Relation for Calculating the Reynolds Stresses, ZAMM, v. 56, T219-T221, 1976.

52. Kays, W.M., Turbulent Prandtl Number - Where Are We?, Journal of Heat Transfer, v. 116, May 1994.

53. Schlichting, H., Boundary Layer Theory, 7th Edition, p. 605, McGraw Hill, New York, 1979.

54. Knudsen, J.G. and Katz, D.L., Fluid Dynamics and Heat Transfer, McGraw Hill, New York, 1958.

55. Ecer, A., Reddy, J.N., Akay, H.U., Habashi, W.G., Ward, P., and Gartling, D.K., Finite Element Method in Fluid Mechanics and Heat Transfer, volume I and II, Seminar, Purdue University, Indianapolis, March 1988.

56. Ghariban, N., Haji-Sheikh, A. and You, S.M., Pressure Drop and Heat Transfer in Turbulent Duct Flow: A Two-Parameter Variational Method, J. Heat Transfer, v. 117, pp. 289-295, May 1995.

57. Roberts, H., Finite Element Analysis of Turbulent Fluid Flow Using Primitive Variables, M. Eng. Thesis, McMaster University, December 1984.

58. Morgan, K., Hughes, T.G. and Taylor, C., Investigation of a Mixing Length and a Two-Equation Turbulence Model Utilizing the Finite-Element Method, Appl. Math. Modelling, v. 1, pp. 395-400, December 1977.

59. Reddy, J.N., An Introduction to the Finite Element Method, McGraw-Hill Book Company, 1984.

60. Zienkiewicz, O.C. and Morgan, K., Finite Elements and Approximation, John Wiley and Sons, p. 66, 1983.

61. Lapidus, L. and Pinder, G.F., Numerical Solution of Partial Differential Equations in Science and Engineering, John Wiley and Sons, 1982.

62. Reddy, J.N. and Gartling, D.K., The Finite Element Method in Heat Transfer and Fluid Dynamics, CRC Press Inc., 1994.

63. Reynolds, A.J., Turbulent Flows in Engineering, Wiley, 1974.

64. Baker, A.J., Laminar and Turbulent Boundary Layer Flow, Recent Advances in Numerical Methods in Fluids, v. 1, pp. 287-309, 1980.

65. Taylor, C., Hughes, T.G. and Morgan, K., A Numerical Analysis of Turbulent Flow in Pipes, Computers and Fluids, v. 5, pp. 191-204, 1977

66. Lamb, H., Hydrodynamics, 5th edition, Cambridge University Press, London, 1924.

67. Ivey, C.M., The Position of Maximum Velocity in Annular Flow, M.Sc. Thesis, University of Windsor, Canada, 1965.

68. Hollingsworth, D.K., Kays, W.M. and Moffat, R.J., Measurement and Prediction of the Turbulent Thermal Boundary Layer in Water on Flat and Concave Surfaces, Report Number HMT-41, Thermosciences Division, Depart. of Mech. Engr., Stanford Univ., Stanford, CA, September 1989.

69. Griffith, P. and Wallis, J.D., Chemical Engineering Progress Symposium Series, v. 56, p. 49, 1960.

70. Bergles, A.E., Blumenkrantz, A.R. and Taborek, J., Performance Evaluation Criteria for Enhanced Heat Transfer Surfaces, "Heat Transfer 1974", Proc. 5th Int. Heat Transfer Conference, Tokyo, Vol II, pp. 239-243, 3-7 September 1974.

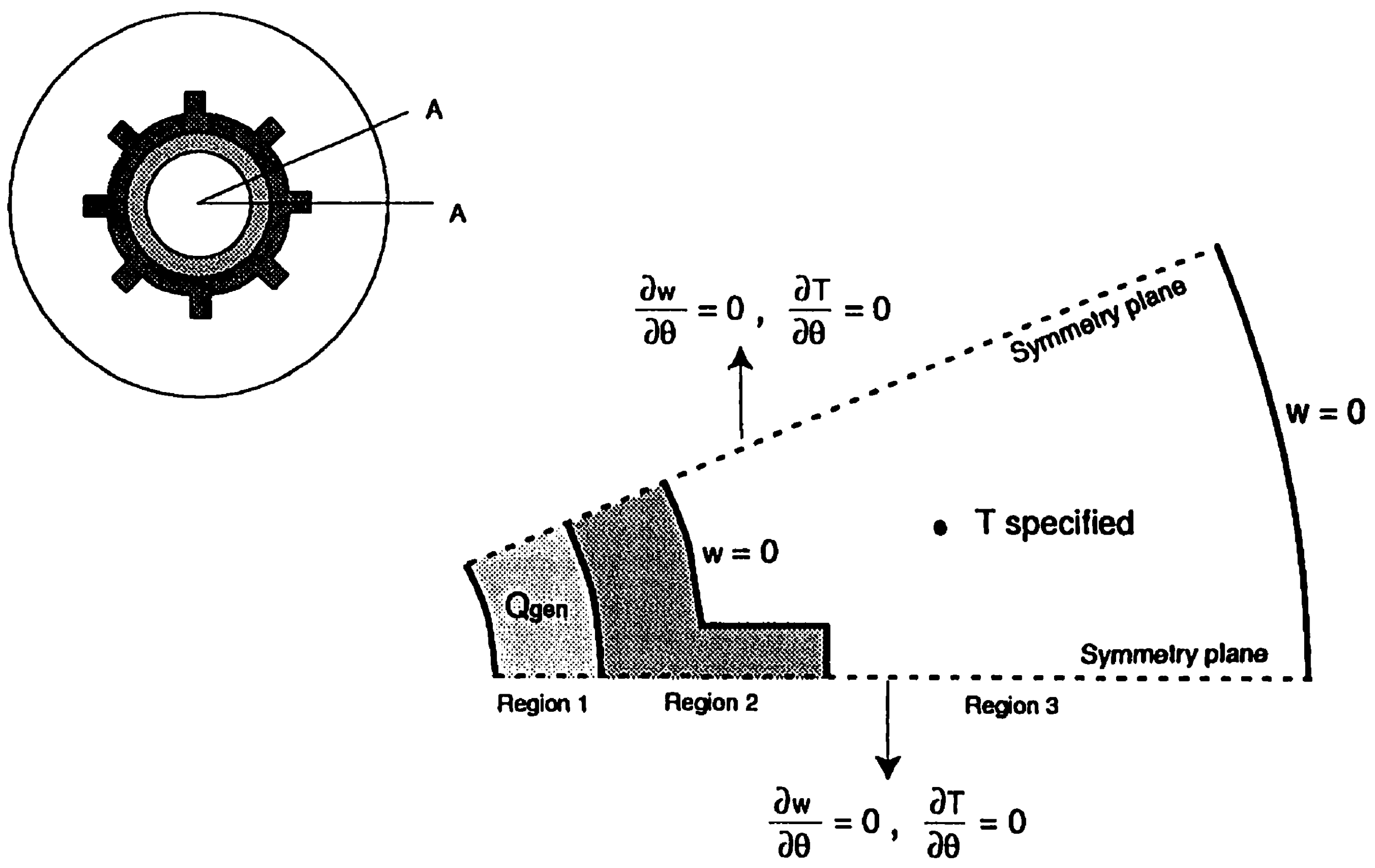

A
A
$\frac{\partial w}{\partial \theta} = 0$, $\frac{\partial T}{\partial \theta} = 0$
Symmetry plane
w = 0
T specified
w = 0
Qgen
Symmetry plane
Region 1
Region 2
Region 3
$\frac{\partial w}{\partial \theta} = 0$, $\frac{\partial T}{\partial \theta} = 0$

Figure 2.1 Statement of the problem

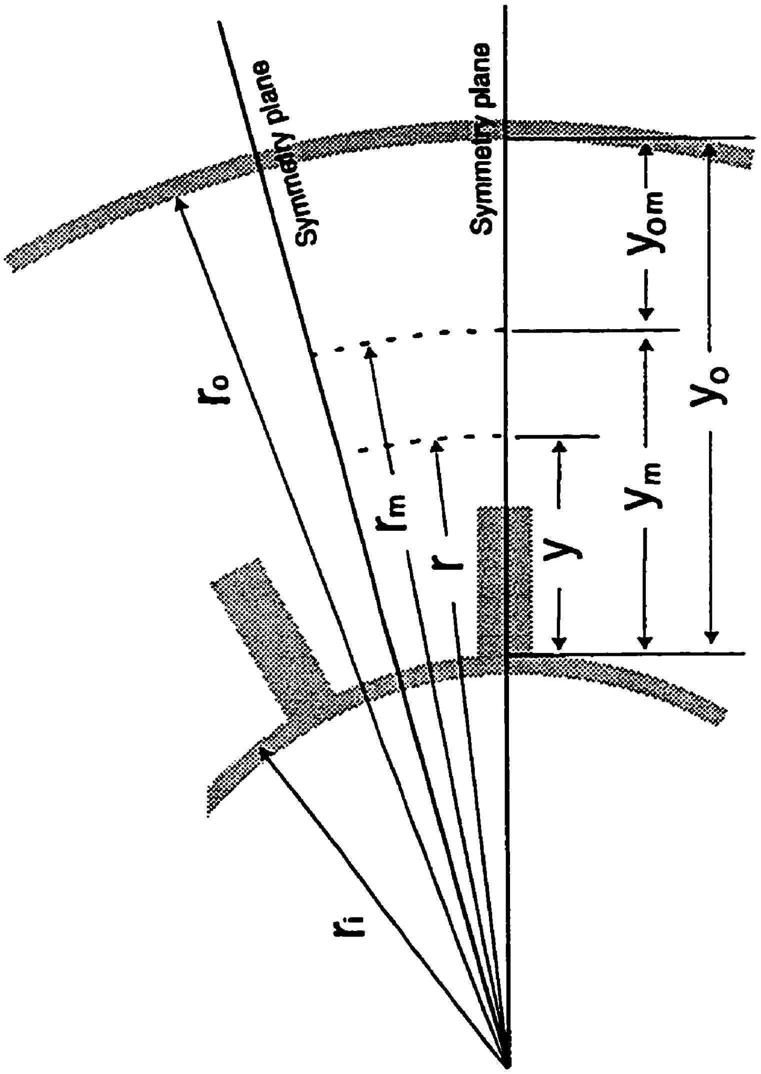
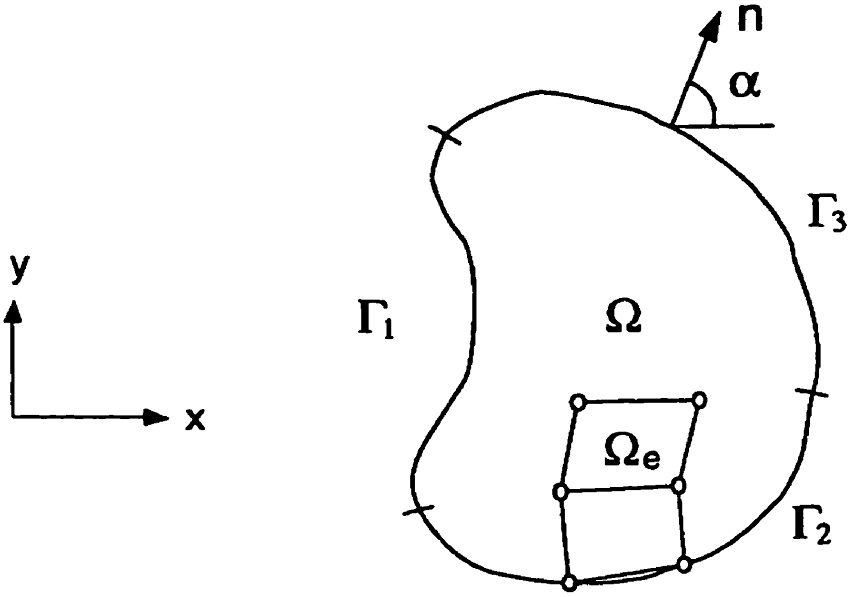
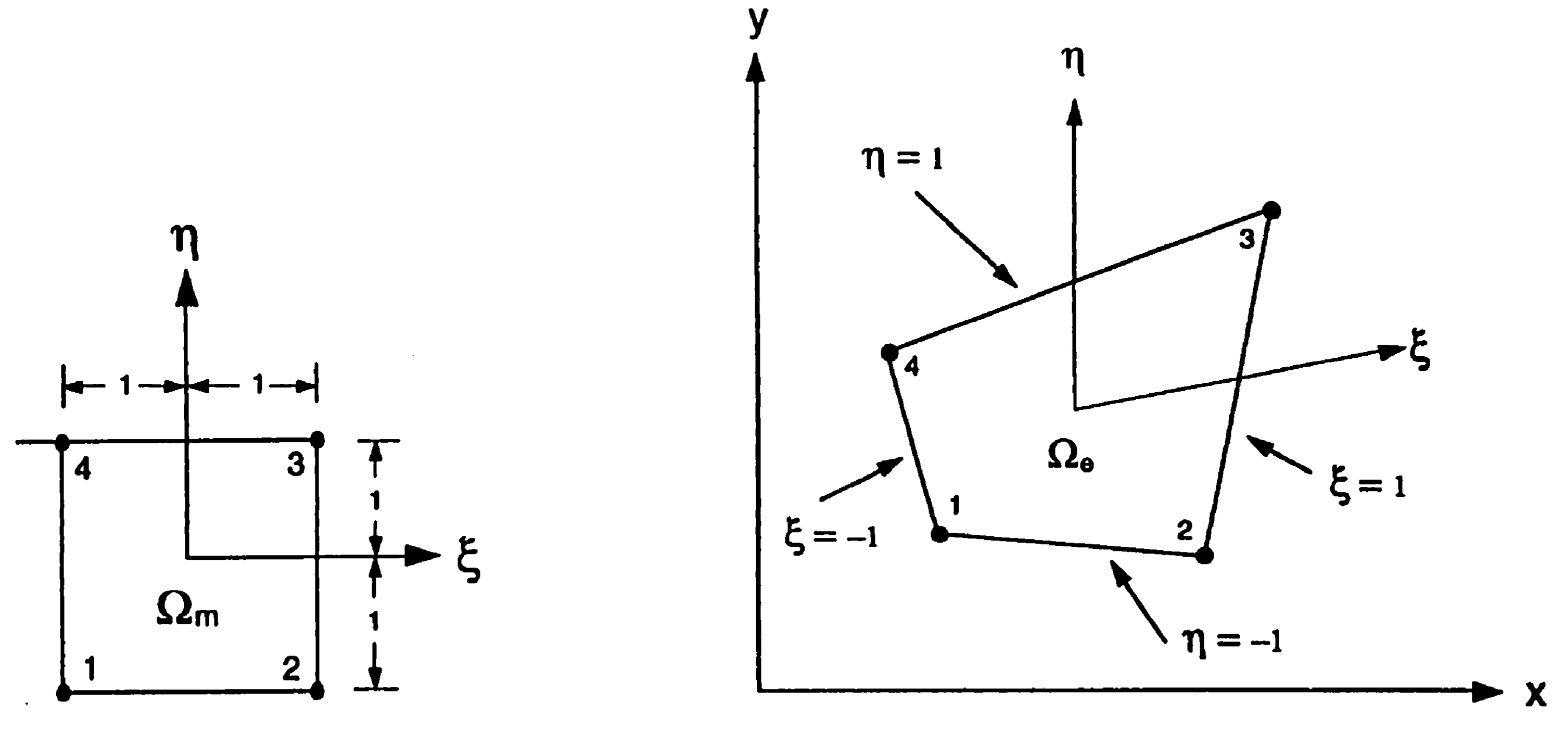


Figure 2.2 Definition of distances



$$\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3$$

Figure 3.1 Domain and its boundaries



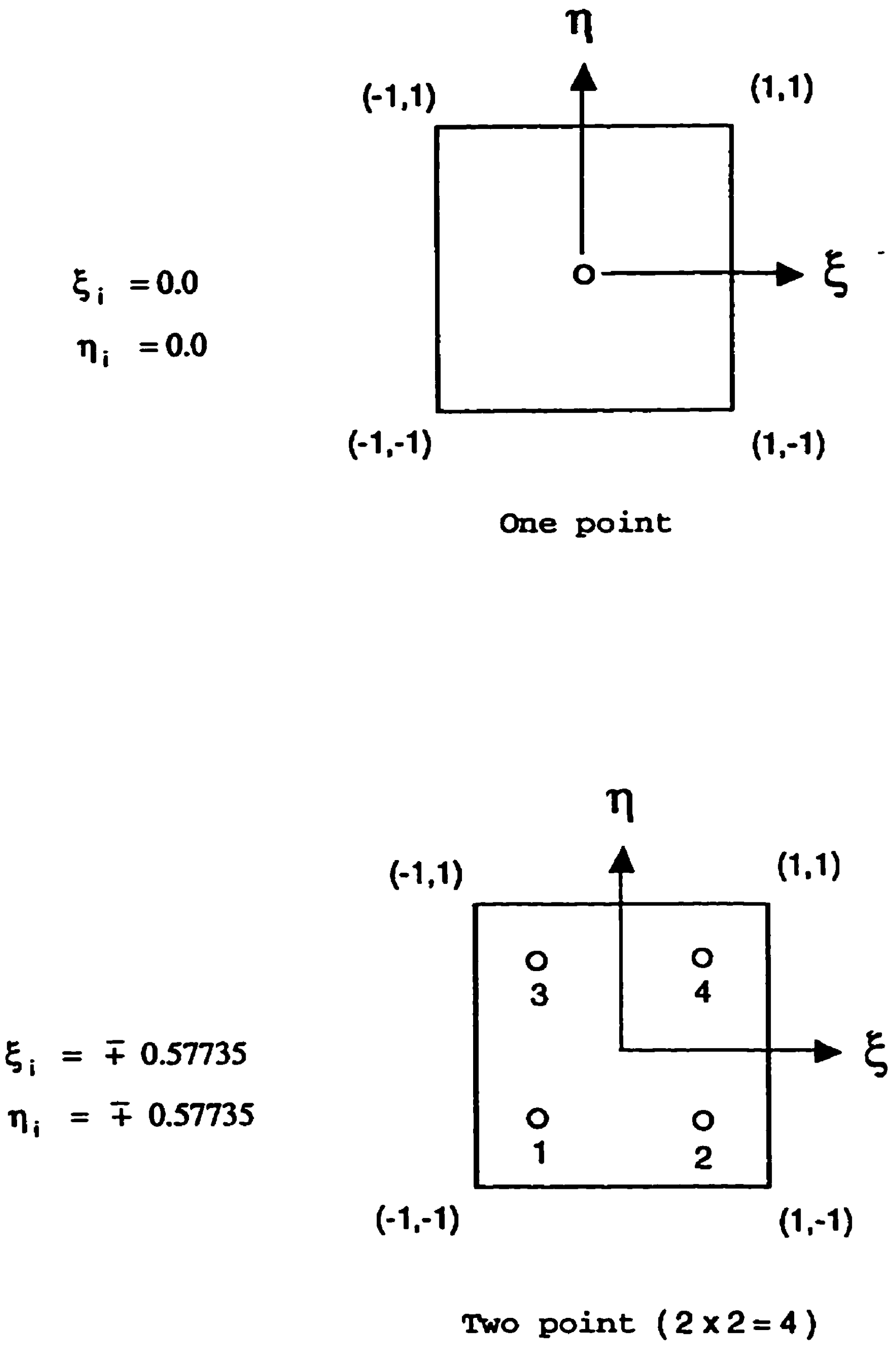
Master element

Only used for evaluating
the coefficient metrix.

Global element

Physical geometry remains the same.

Figure 3.2 Isoparametric mapping of a bilinear element



One point

Two point (2 x 2 = 4)

Figure 3.3 Location of numerical integration points in an element

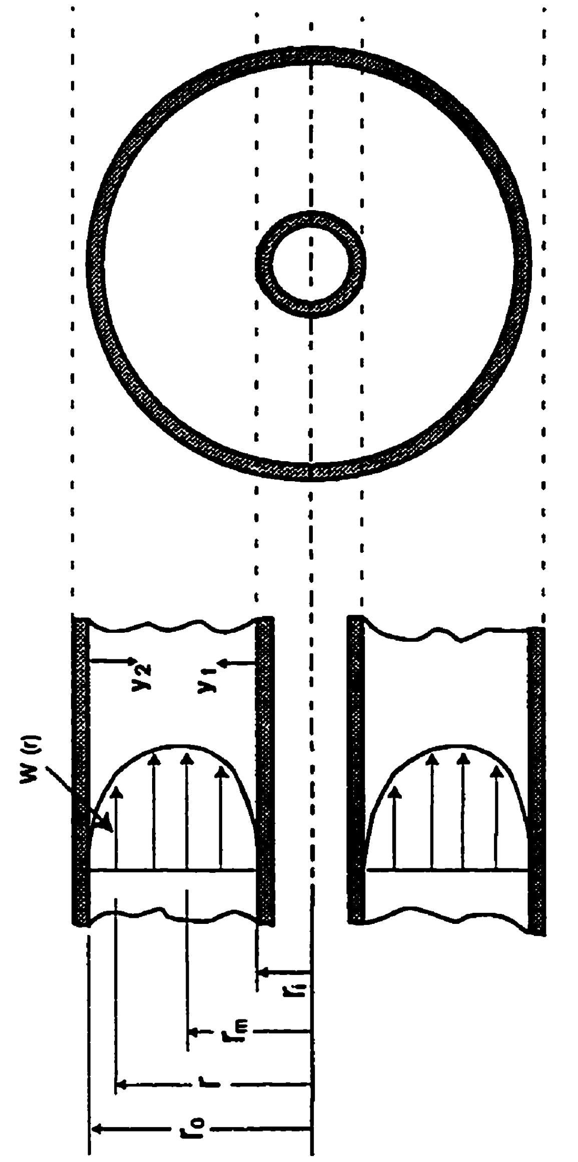


Figure 4.1 Annulus geometry

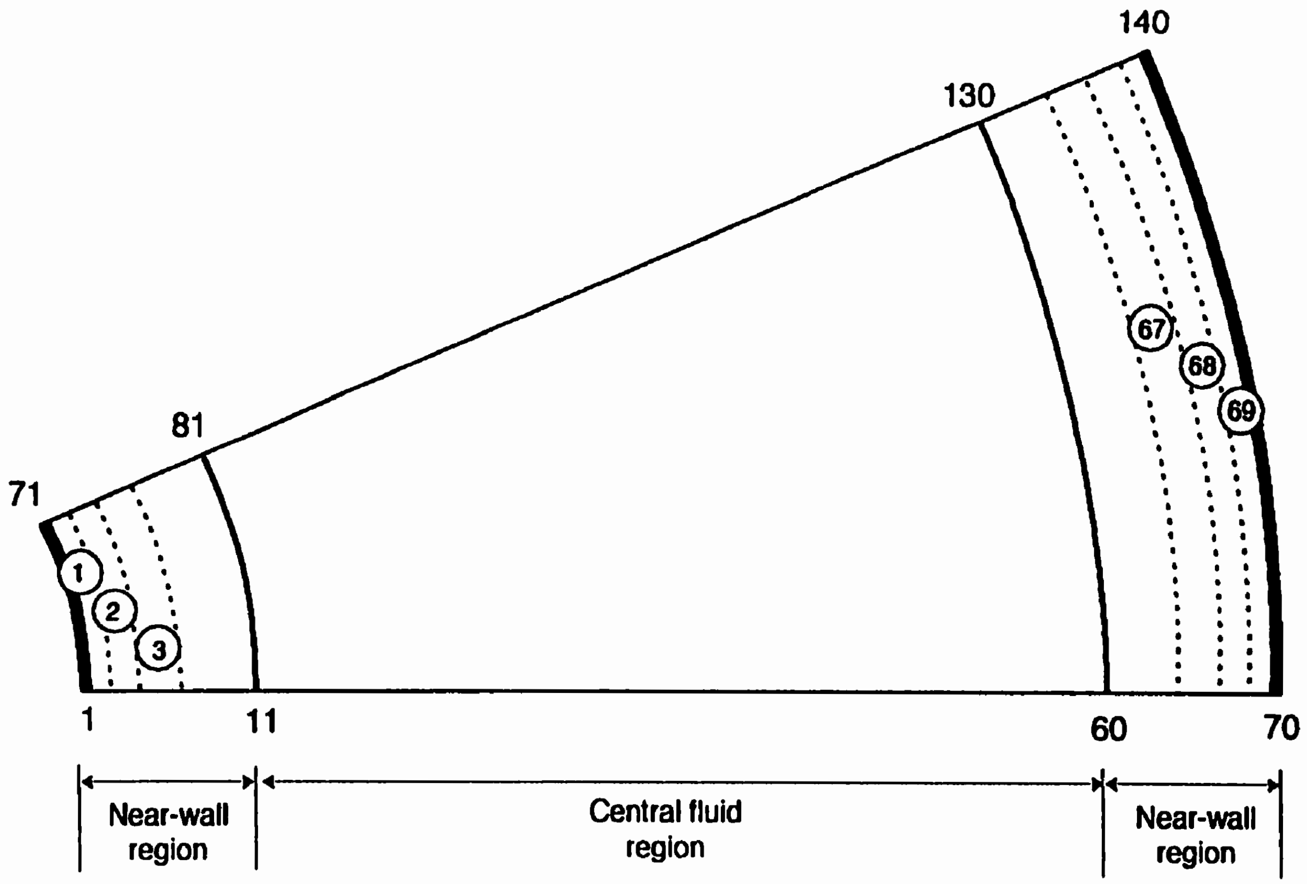


Figure 4.2 Grid used for annulus geometry
Note: Element numbers are circled,
while node numbers are not.

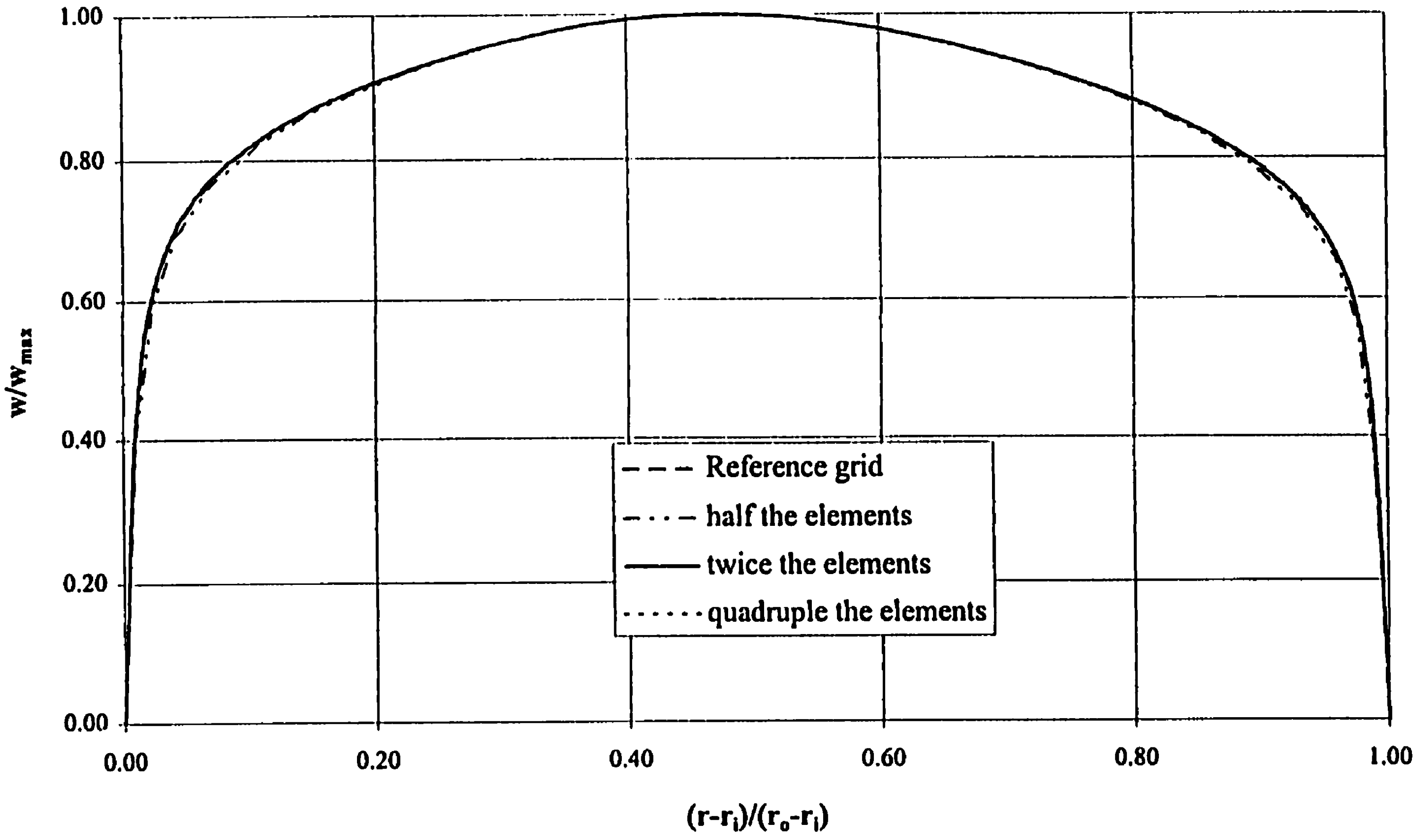


Figure 4.3 Effect of grid size on velocity profile

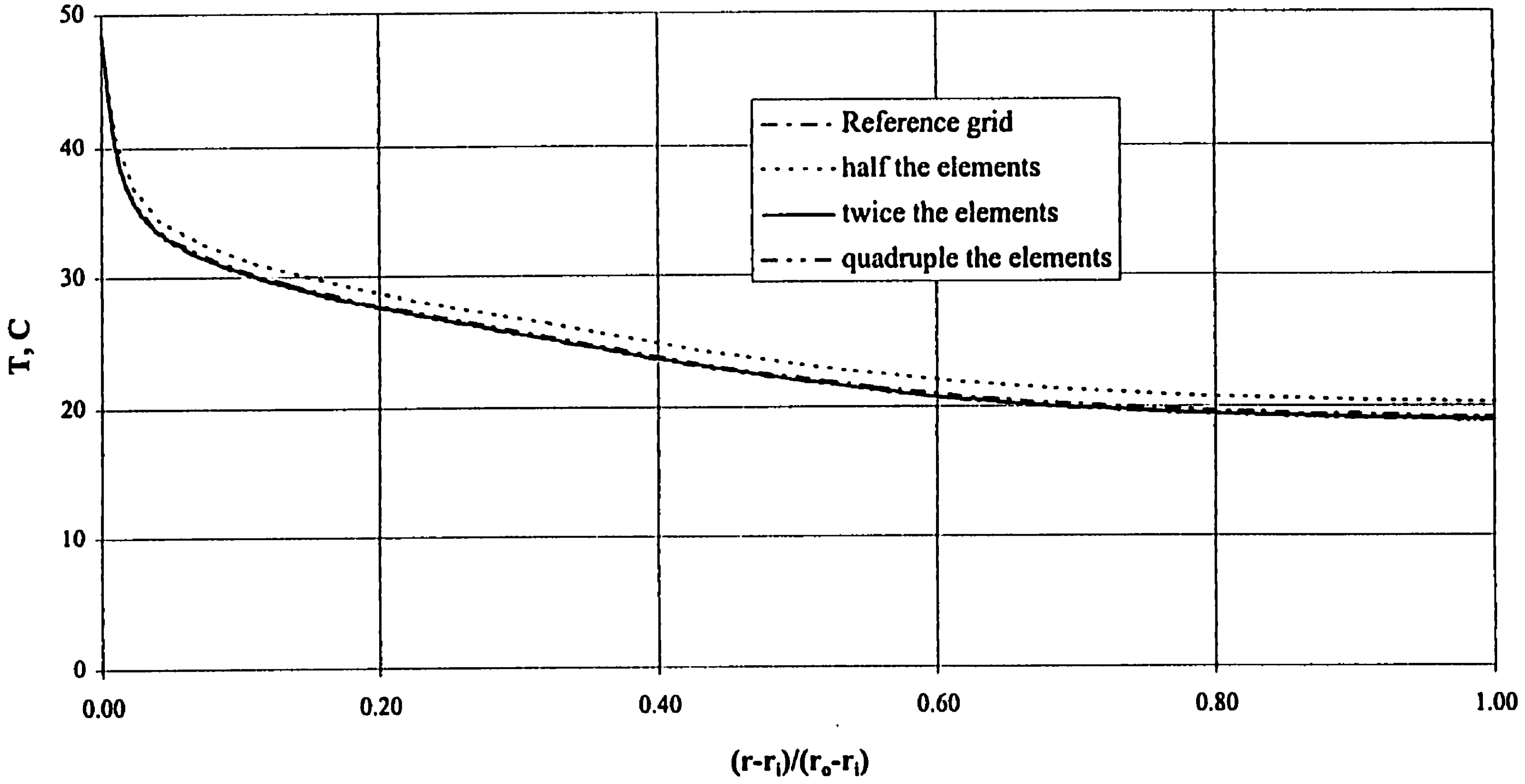


Figure 4.4 Effect of grid size on temperature profile

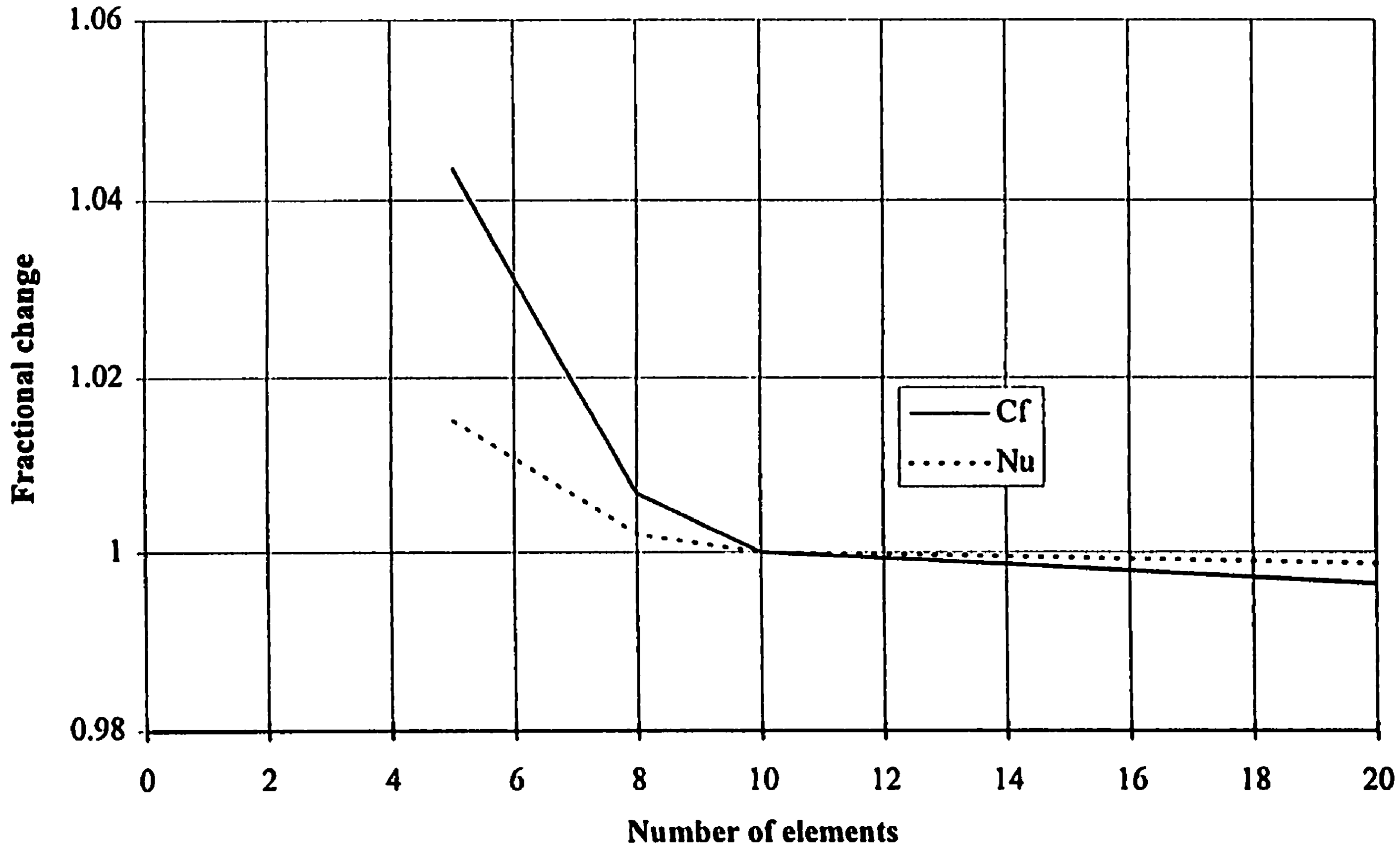


Figure 4.5 Grid convergence test for number of near-wall elements

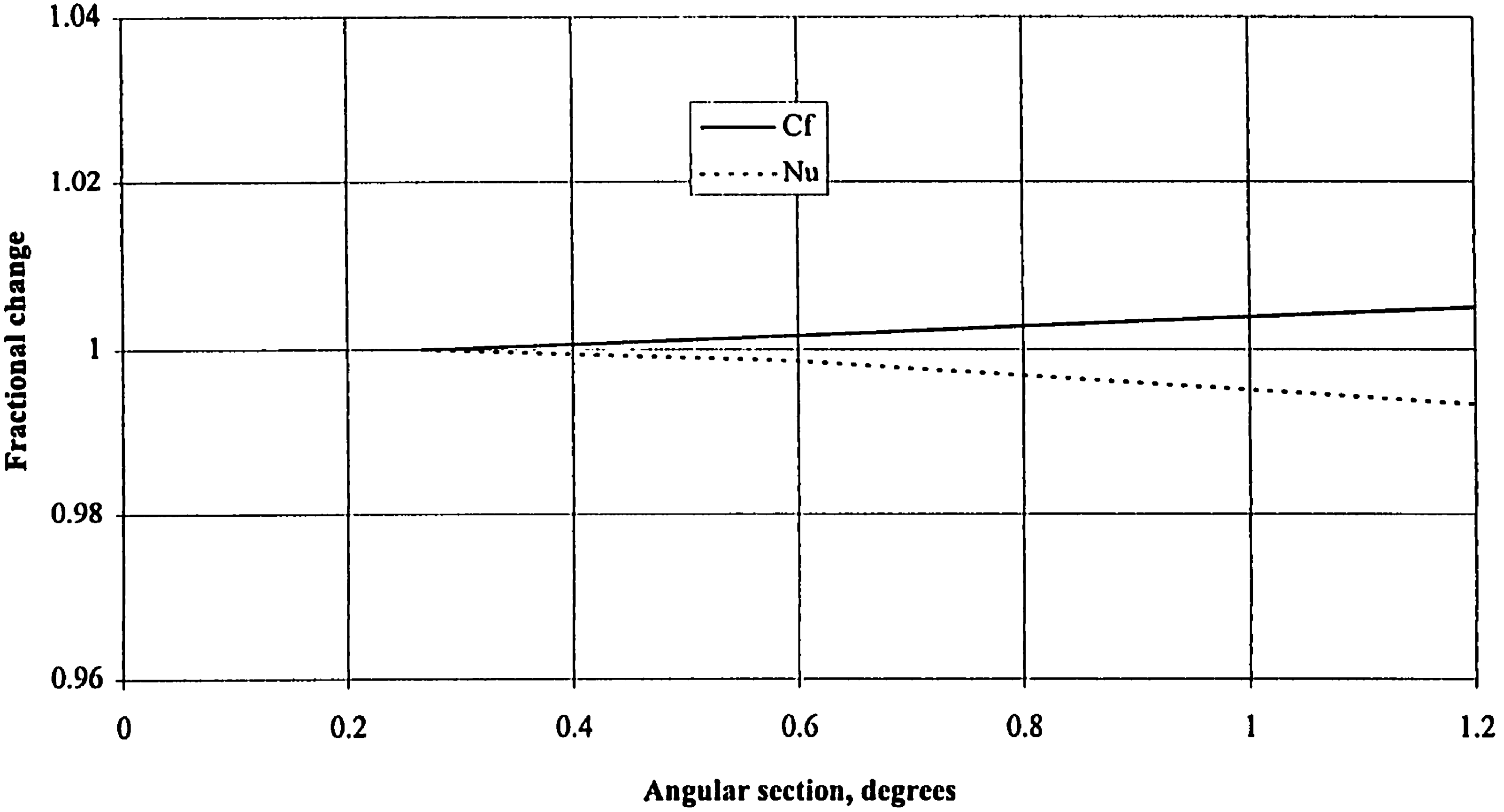


Figure 4.6 Grid convergence test for element aspect ratio

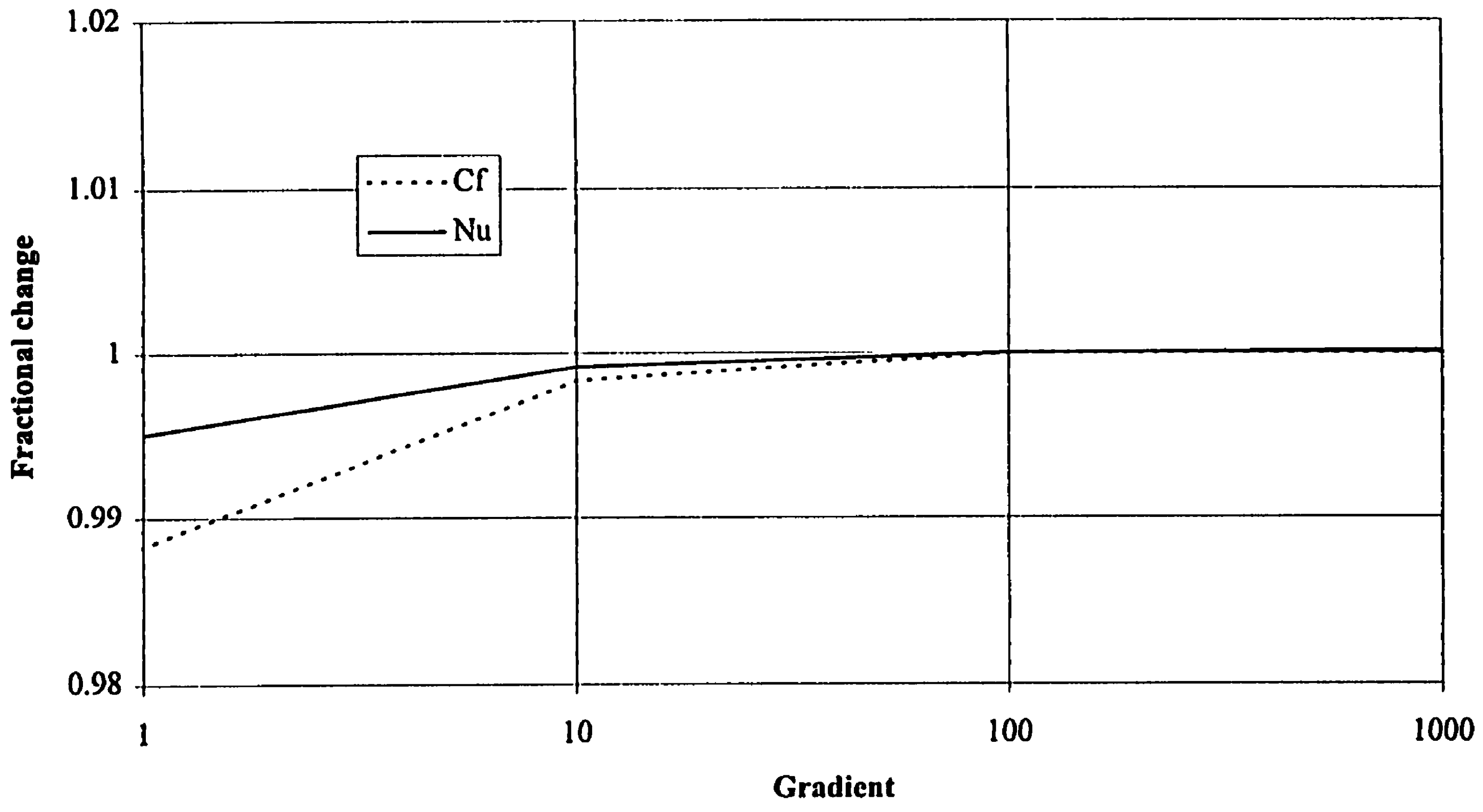


Figure 4.7 Grid convergence test for gradient of near-wall nodes

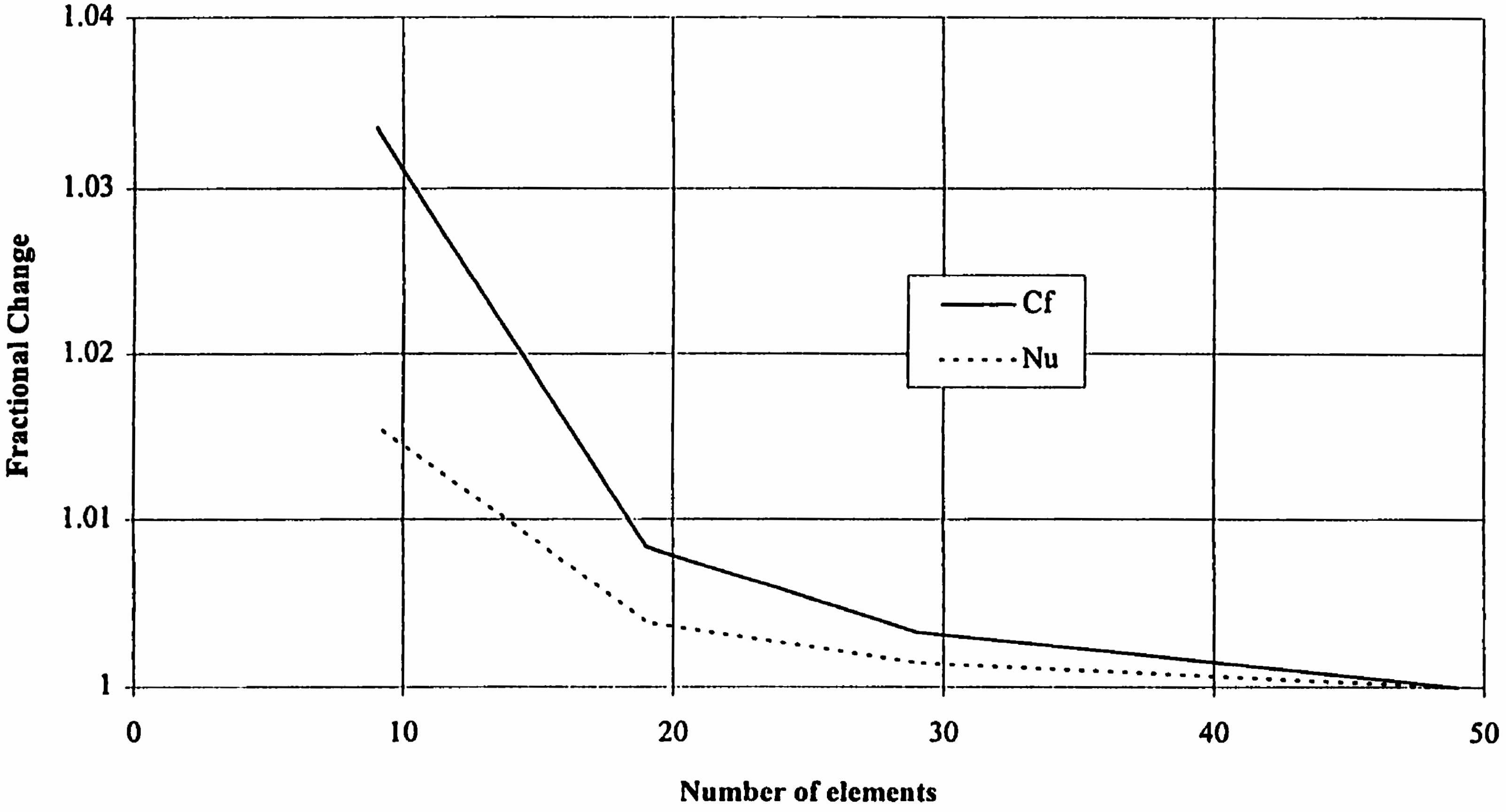


Figure 4.8 Grid convergence test for number of central-region elements

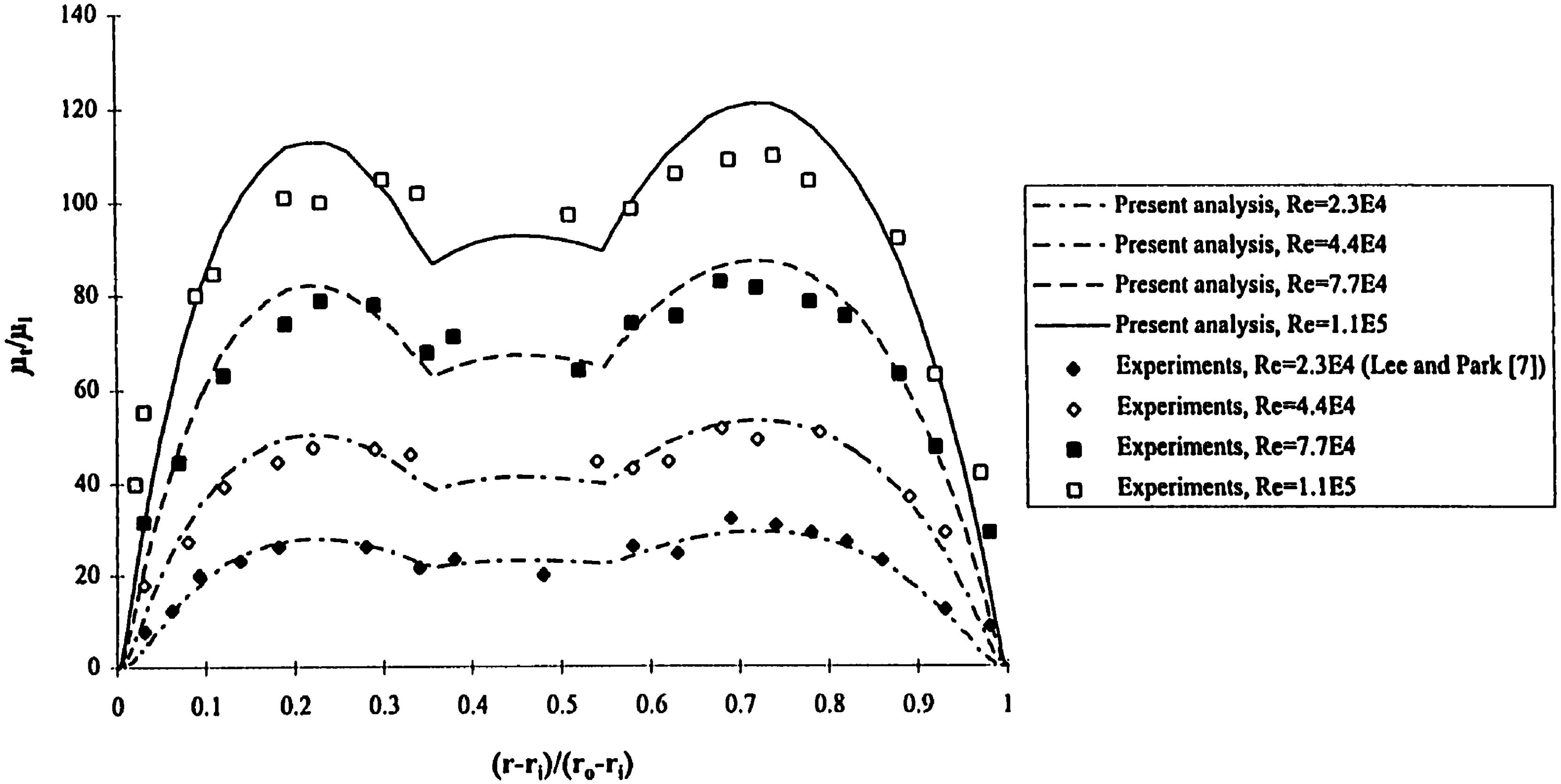


Figure 4.9 Comparison of eddy viscosities in annulus of r_o/r_i=2.31 (Present model)

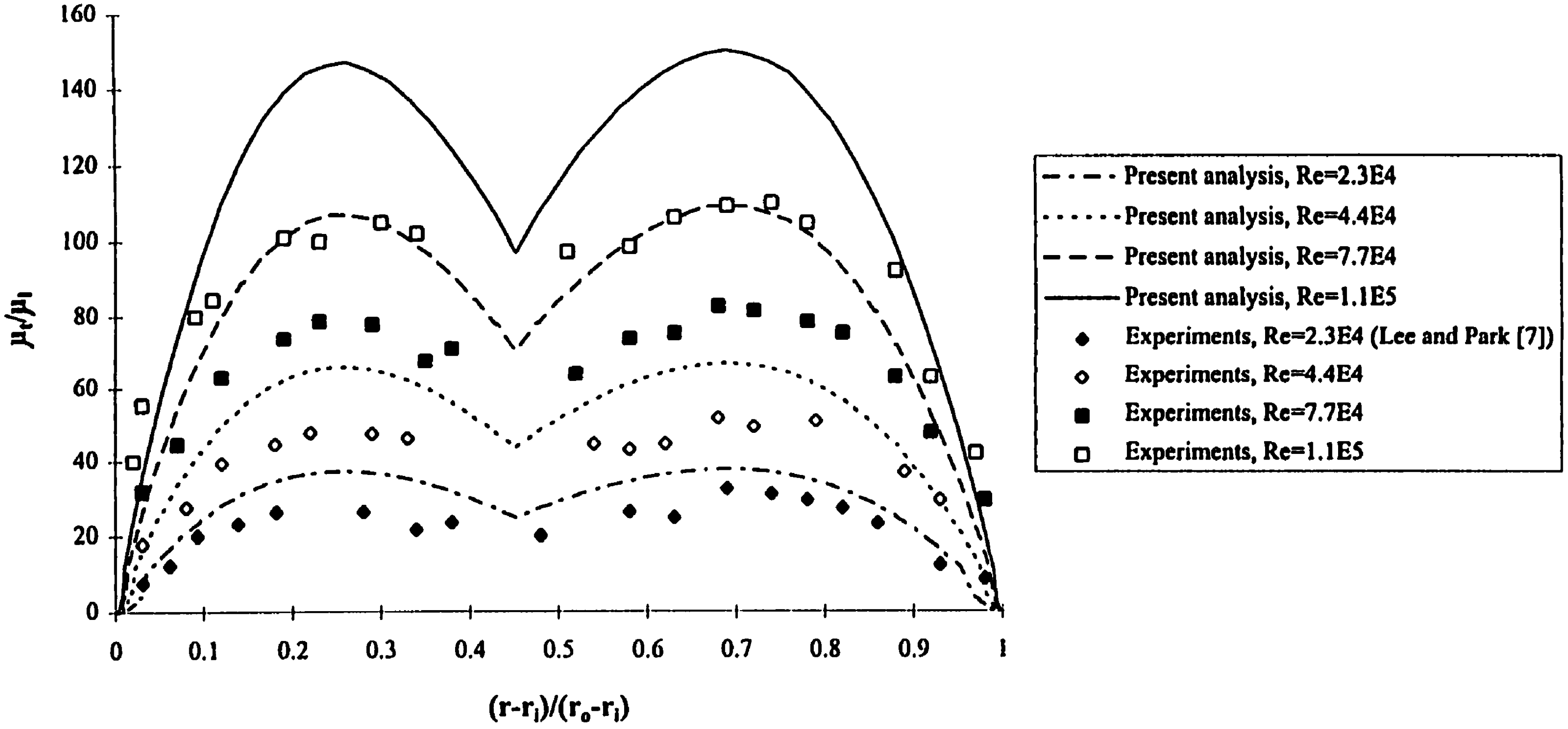


Figure 4.10 Comparison of eddy viscosities in annulus of r_o/r_i=2.31 (Reichardt model)

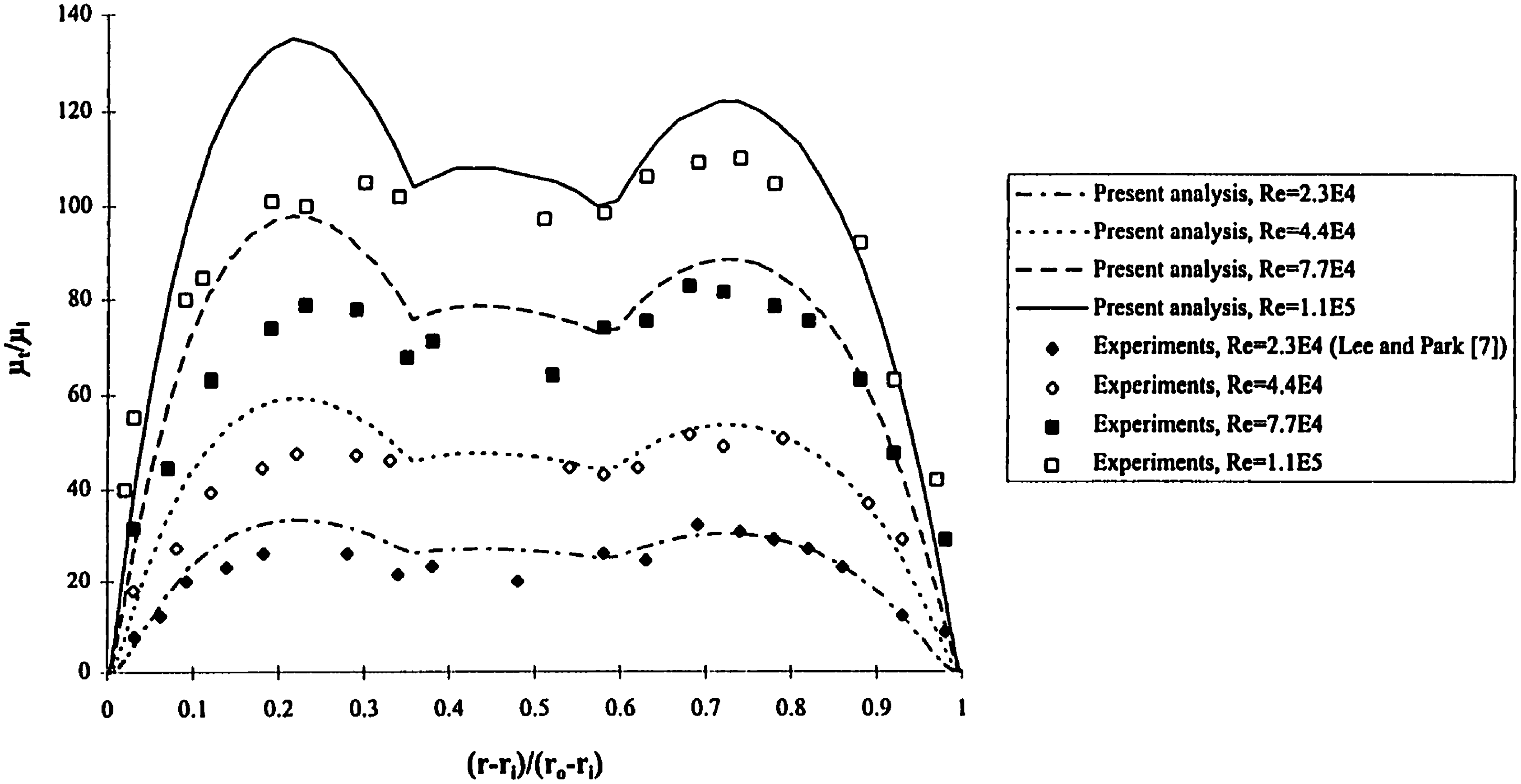


Figure 4.11 Comparison of eddy viscosities in annulus of r_o/r_i=2.31 (Present model with Kays and Leung's r_{mi})

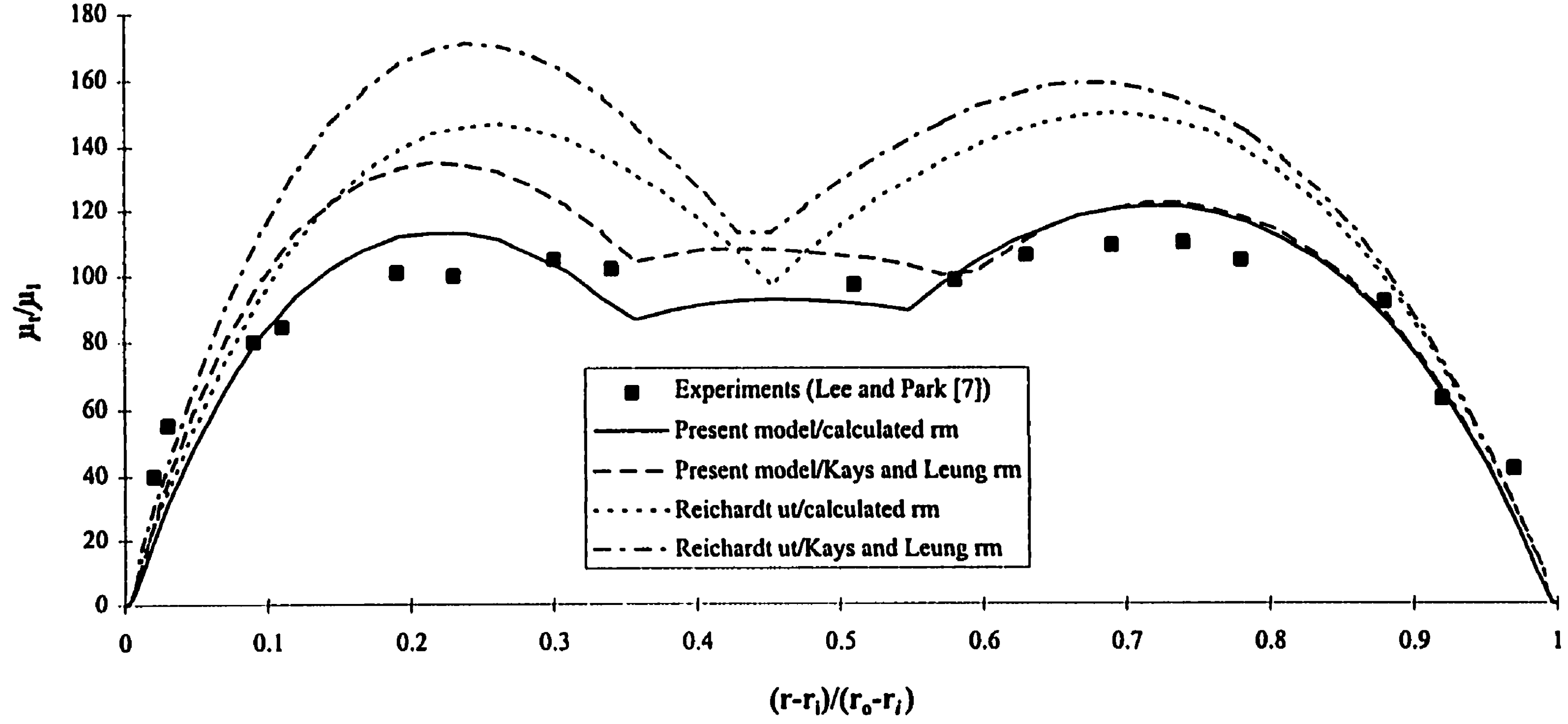


Figure 4.12 Comparison of eddy viscosities from various models for annulus of r_o/r_i=2.31 and Re=1.1E5

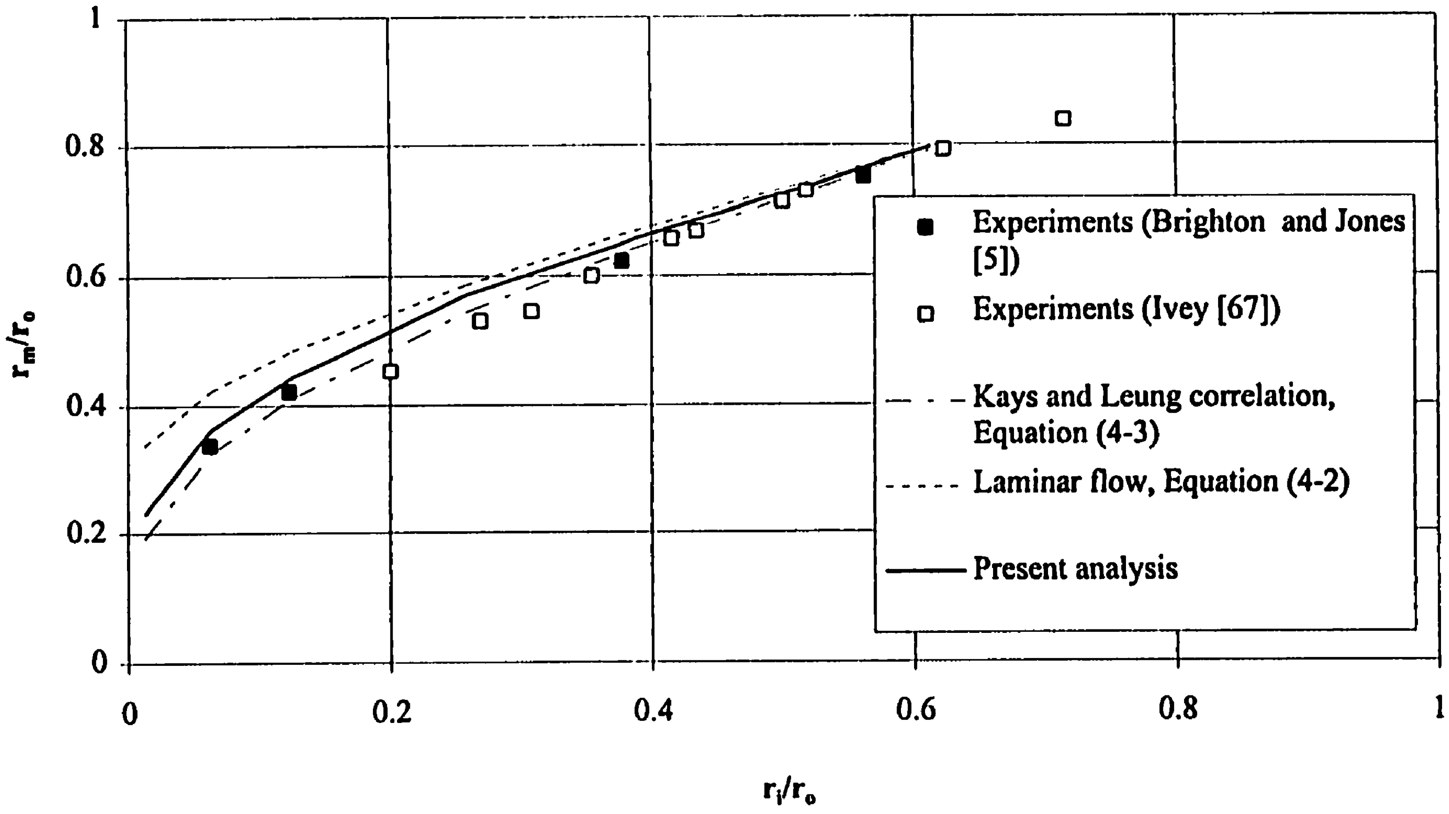


Figure 4.13 Comparison of radius of maximum velocity for various r_i/r_o

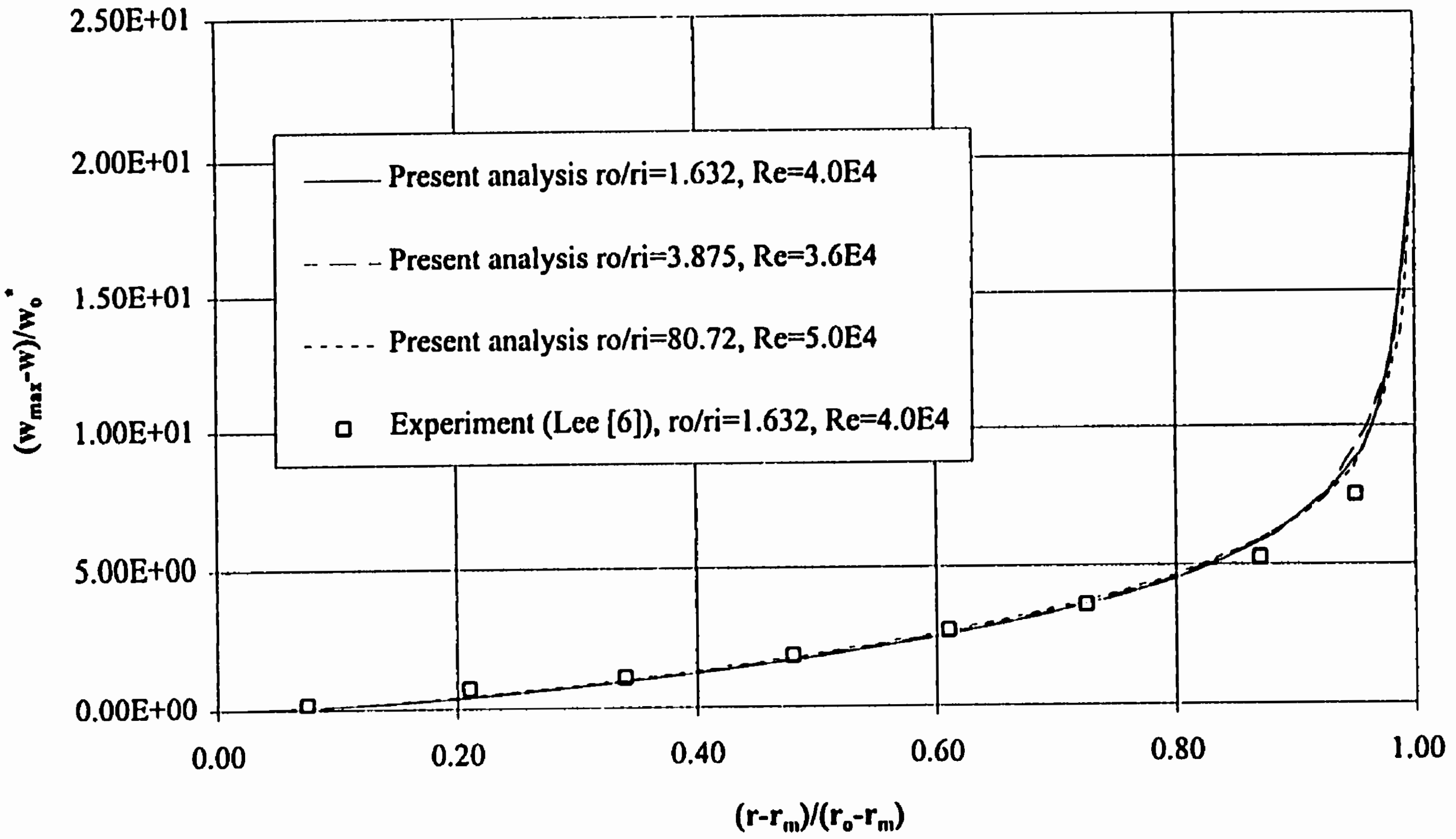


Figure 4.14 Velocity profiles outside the radius of maximum velocity

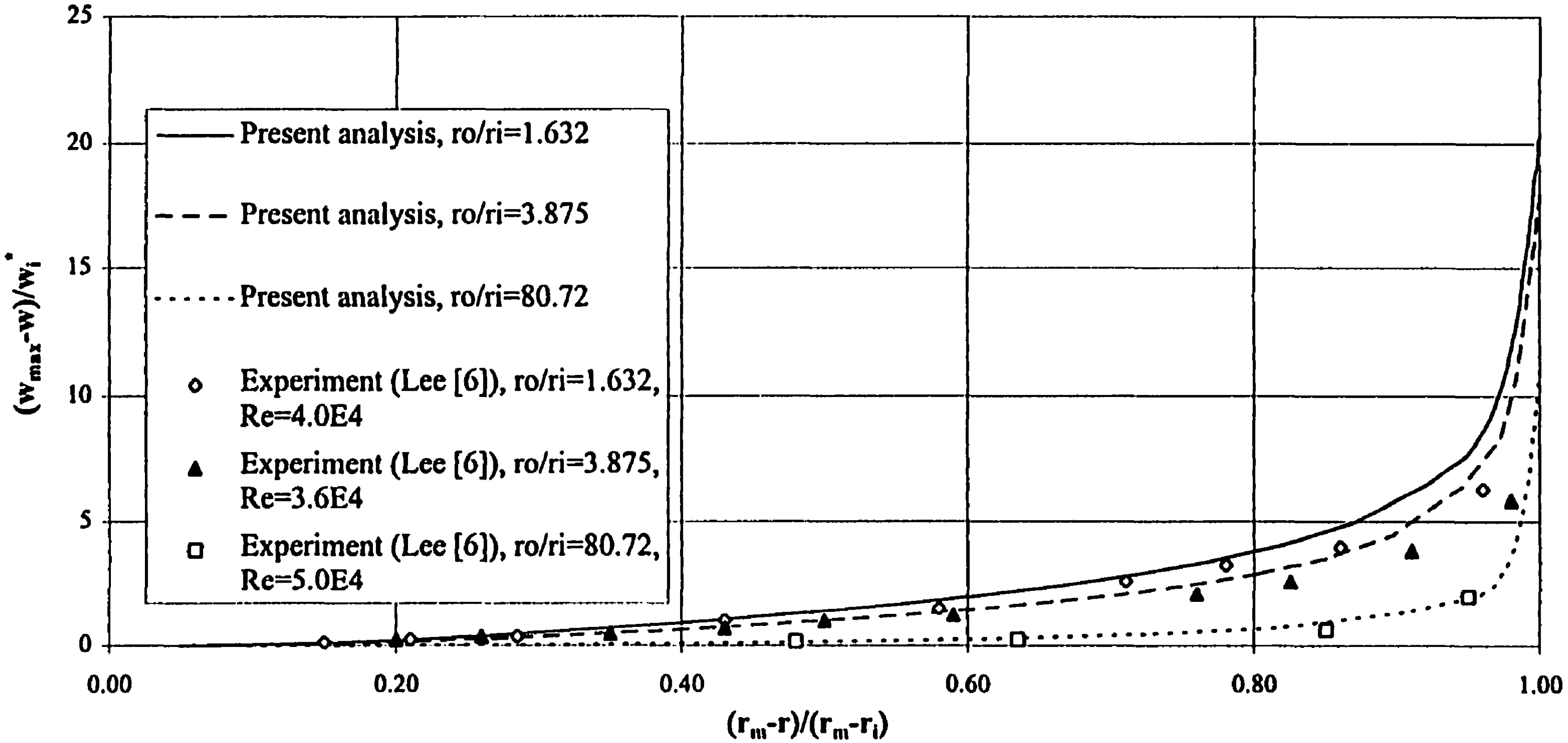


Figure 4.15 Velocity profiles inside the radius of maximum velocity

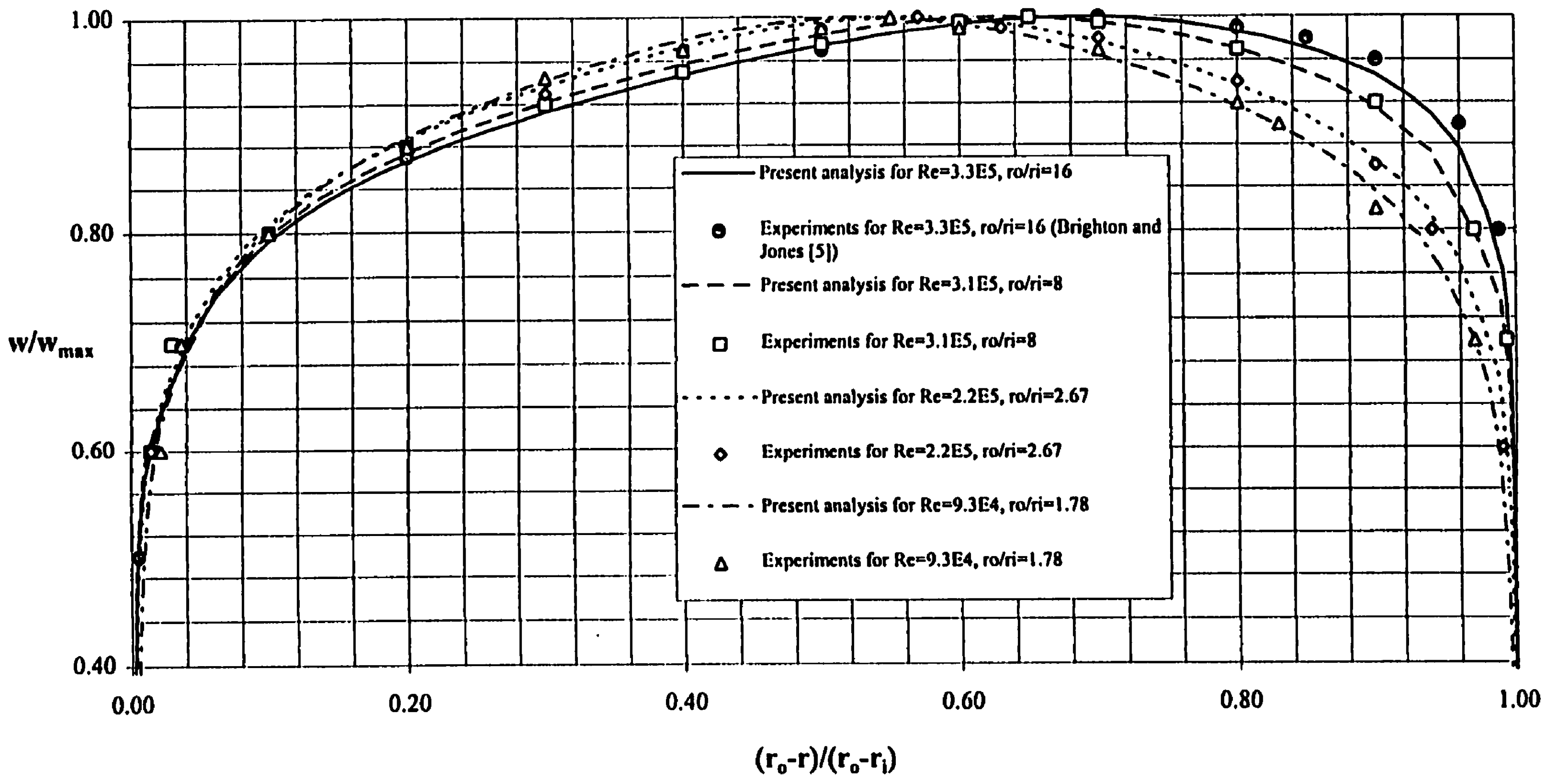


Figure 4.16 Comparison of velocity profiles

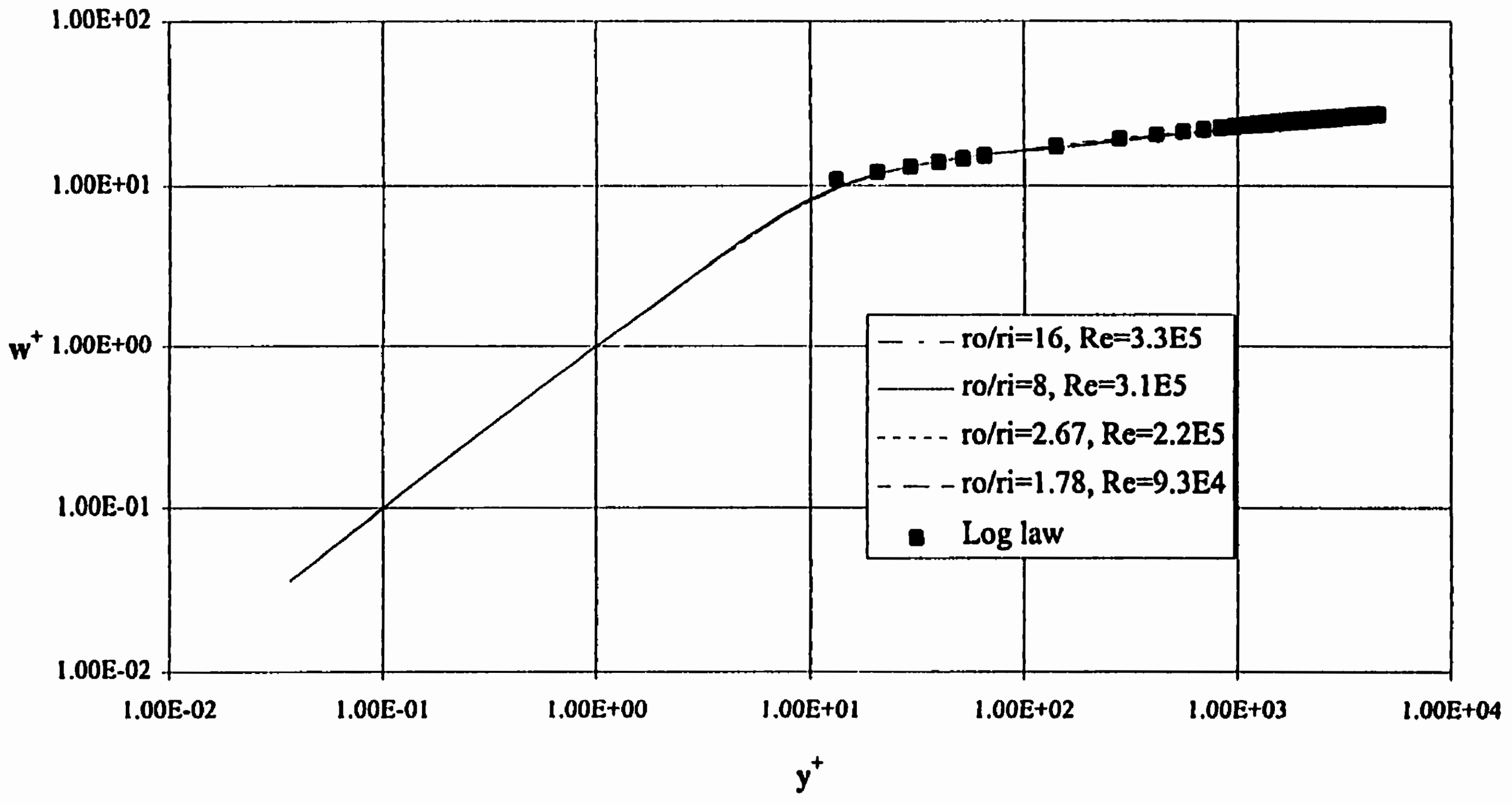


Figure 4.17 Comparison of velocity profiles outside r_m with log-law profile

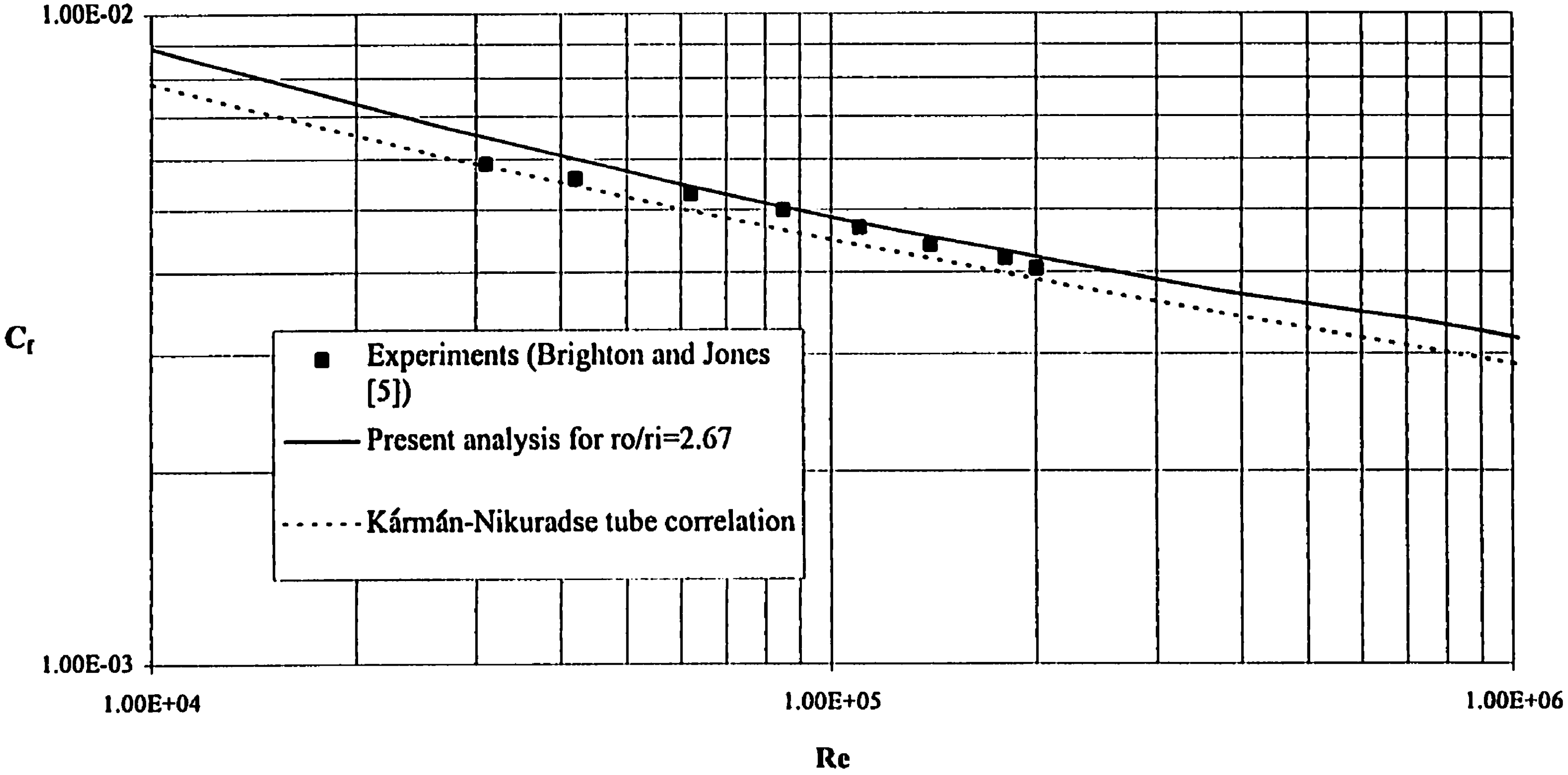


Figure 4.18 Comparison of friction coefficients

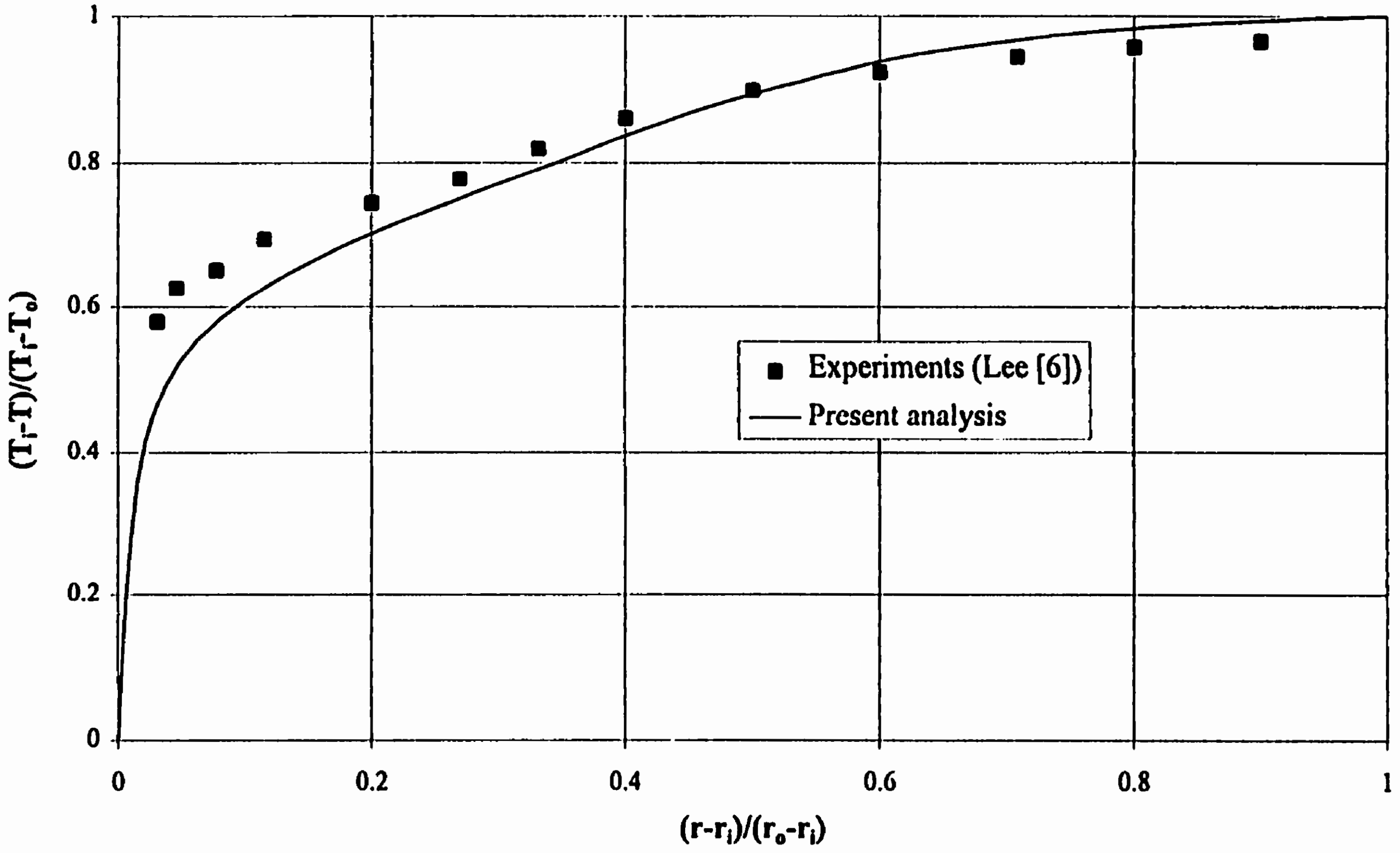


Figure 4.19 Comparison of temperature profiles for r_o/r_i=1.632, Re=4E4 and Pr=0.7

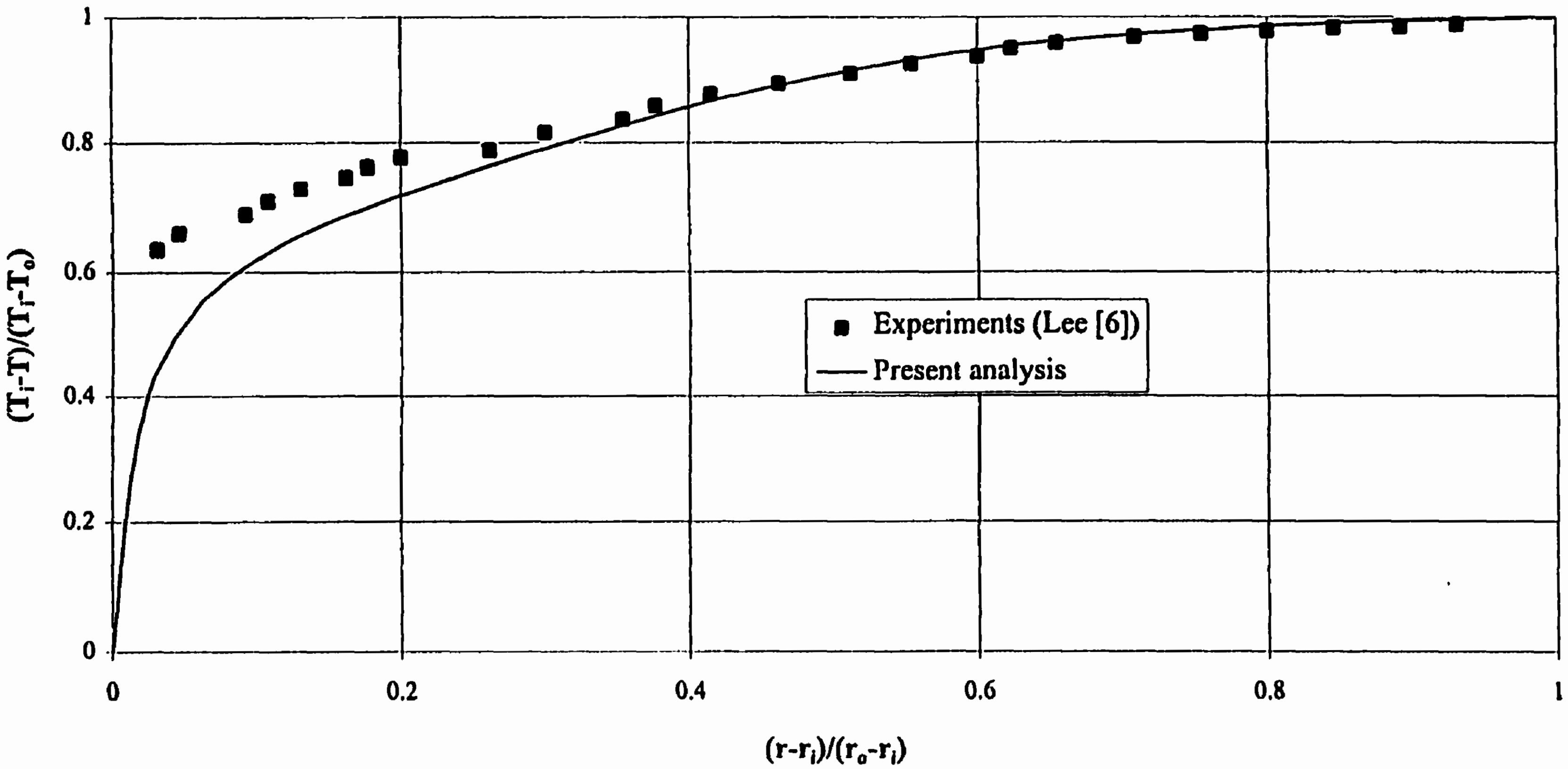


Figure 4.20 Comparison of temperature profiles for r_o/r_i=2.584, Re=2E4 and Pr=0.7

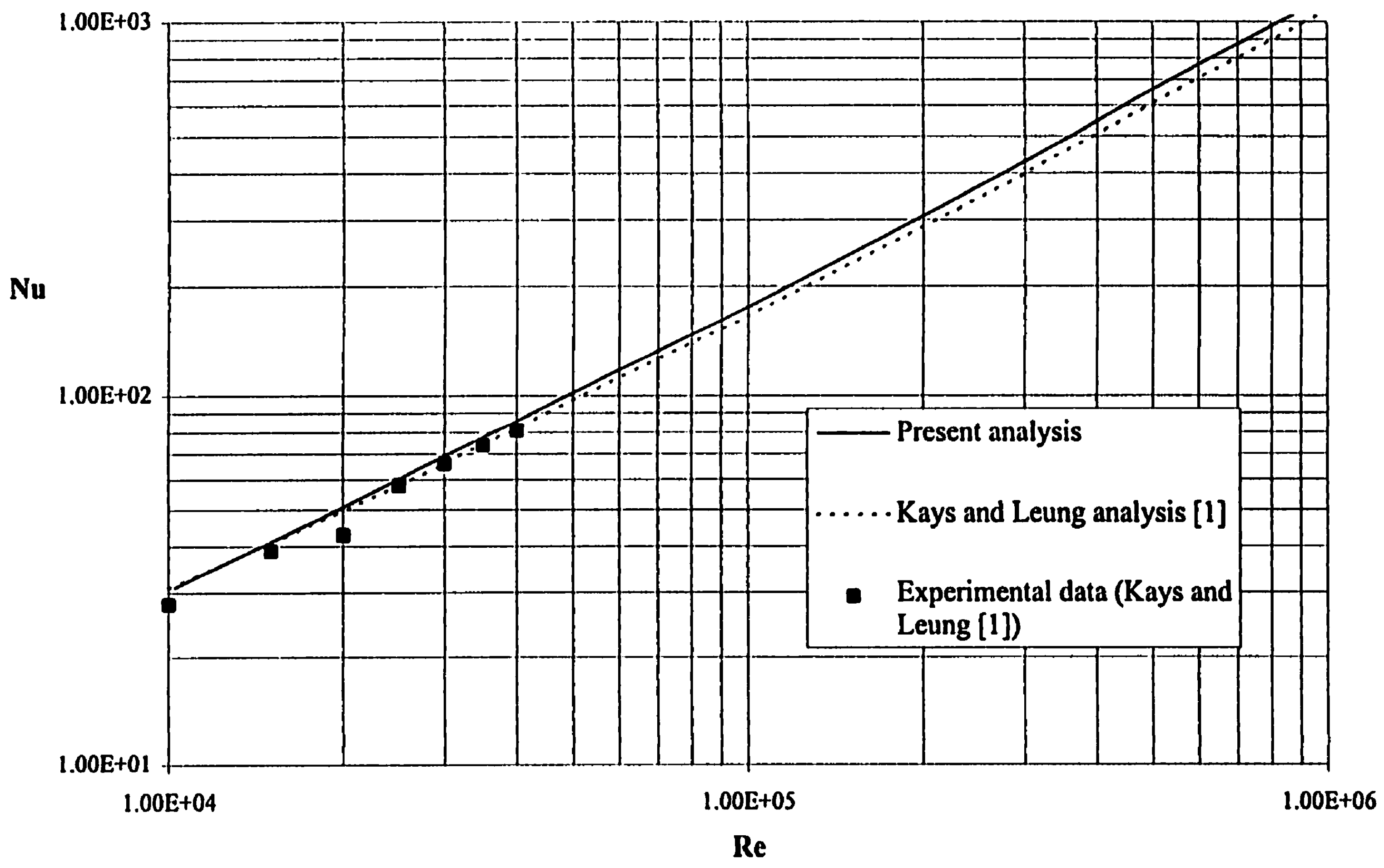


Figure 4.21 Comparison of Nusselt numbers for r_o/r_i=2 and Pr=0.7

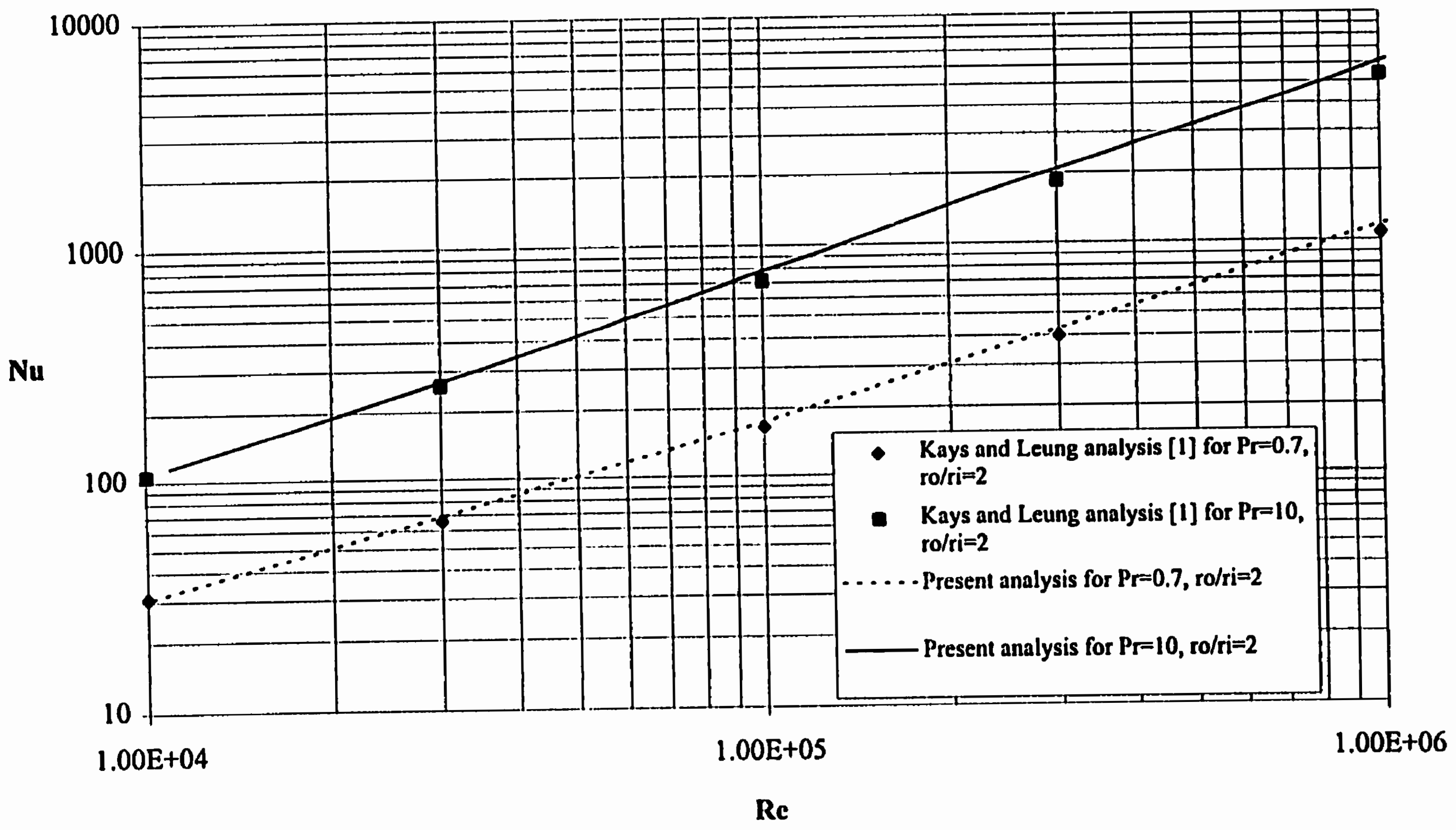


Figure 4.22 Comparison of Pr effect on Nusselt number

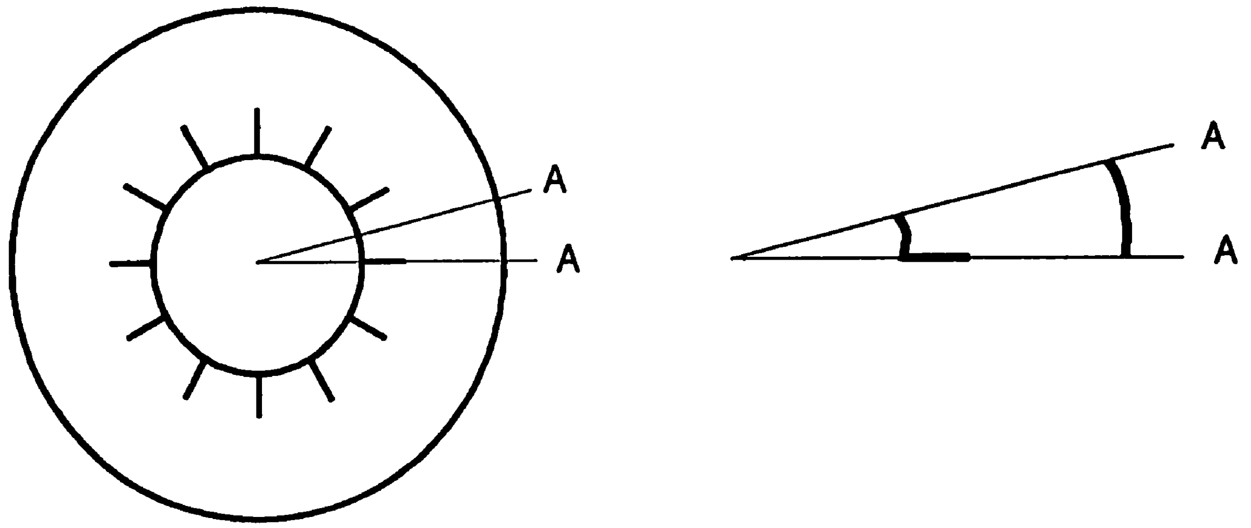


Figure 4.23 Schematic cross-section diagram of an internally finned annulus of Patankar et al.[20]

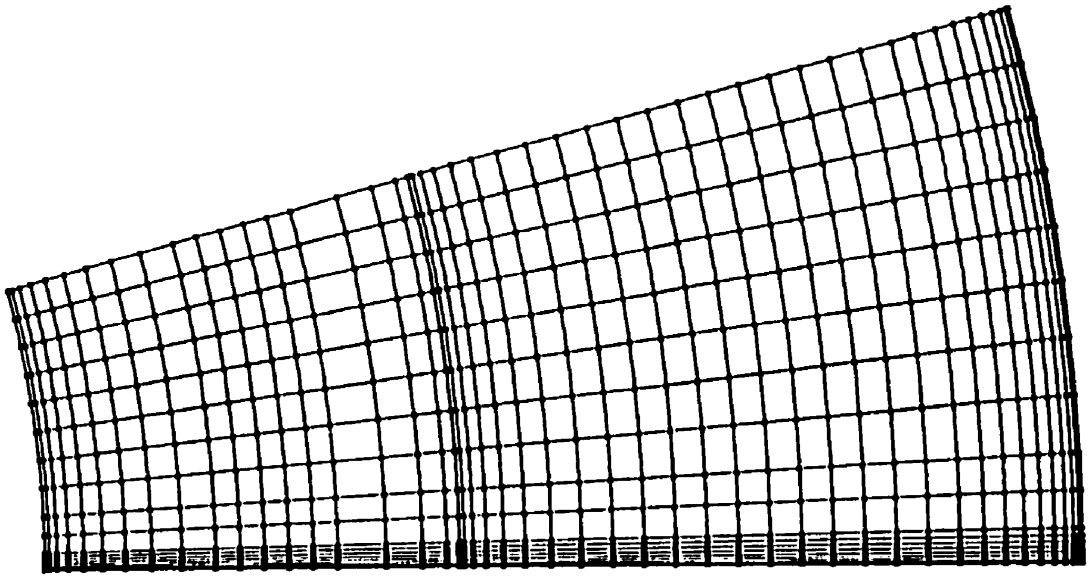


Figure 4.24 Grid used for the analysis of Patankar et al. [20] (1071 nodes and 1000 elements)

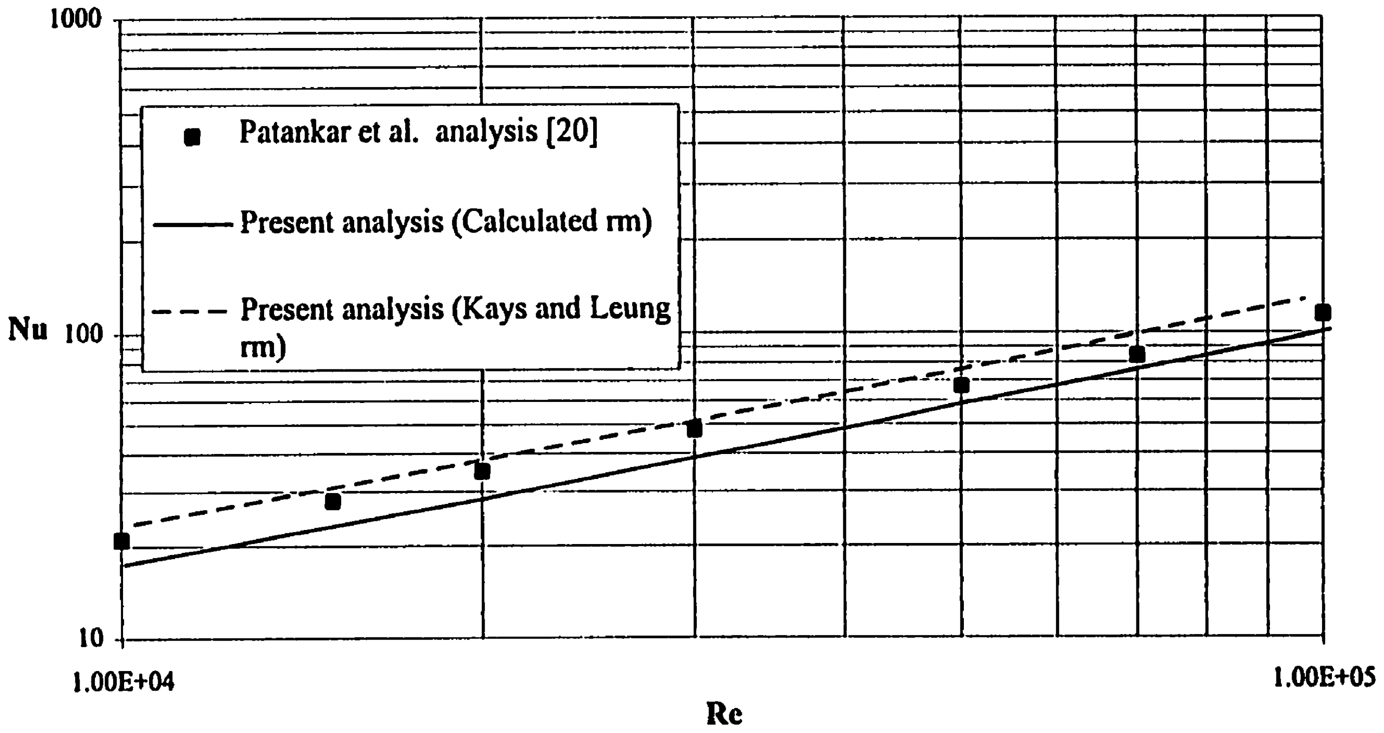


Figure 4.25 Comparison of Nu with the analysis of Patankar et al. [20]
(r_o/r_i=2, N=12, $H/(r_o-r_i)$=0.4 and Pr=0.7)

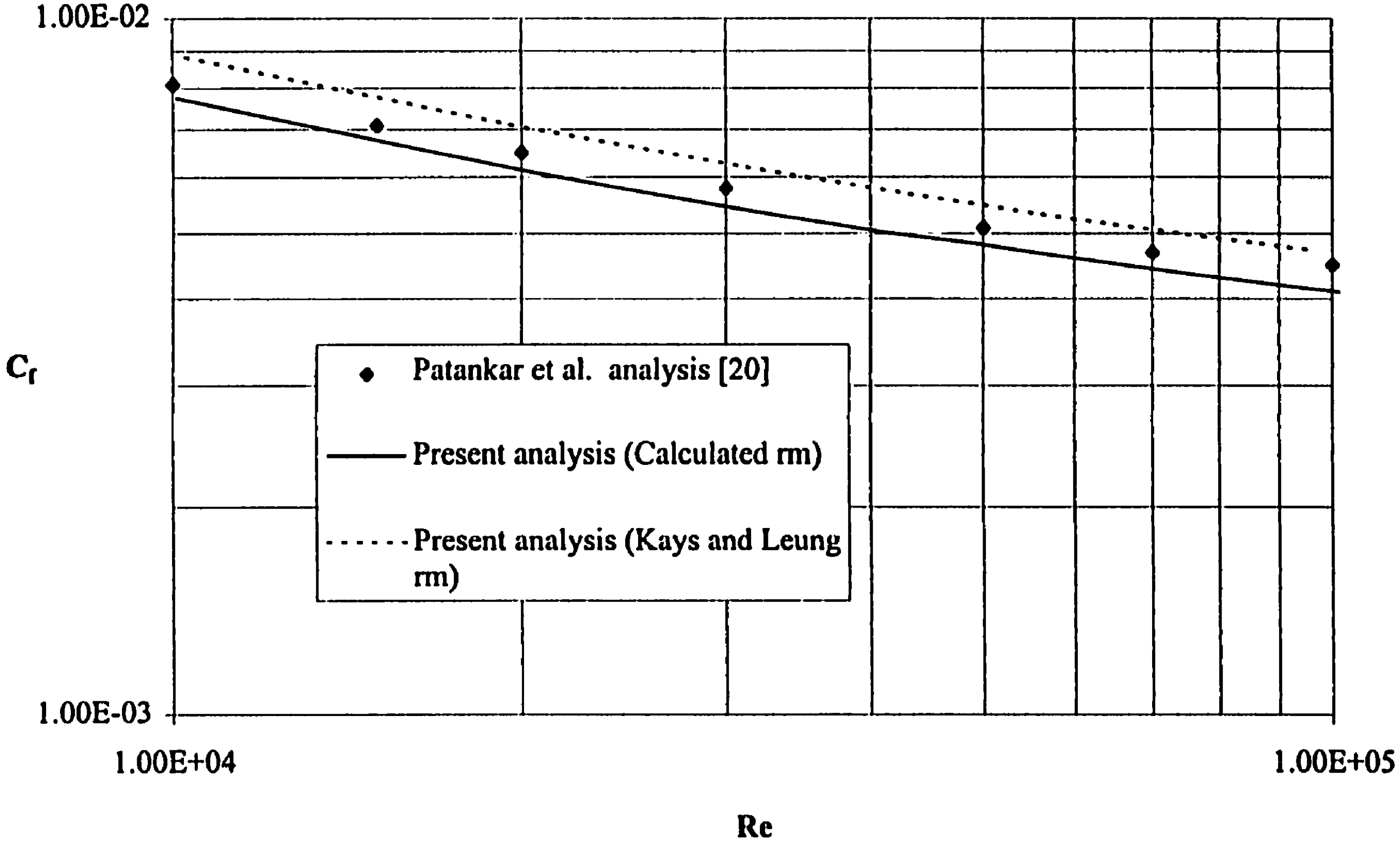


Figure 4.26 Comparison of C_f with the analysis of Patankar et al. [20] (r_o/r_i=2, N=12, H(r_o-r_i)=0.4 and Pr=0.7)

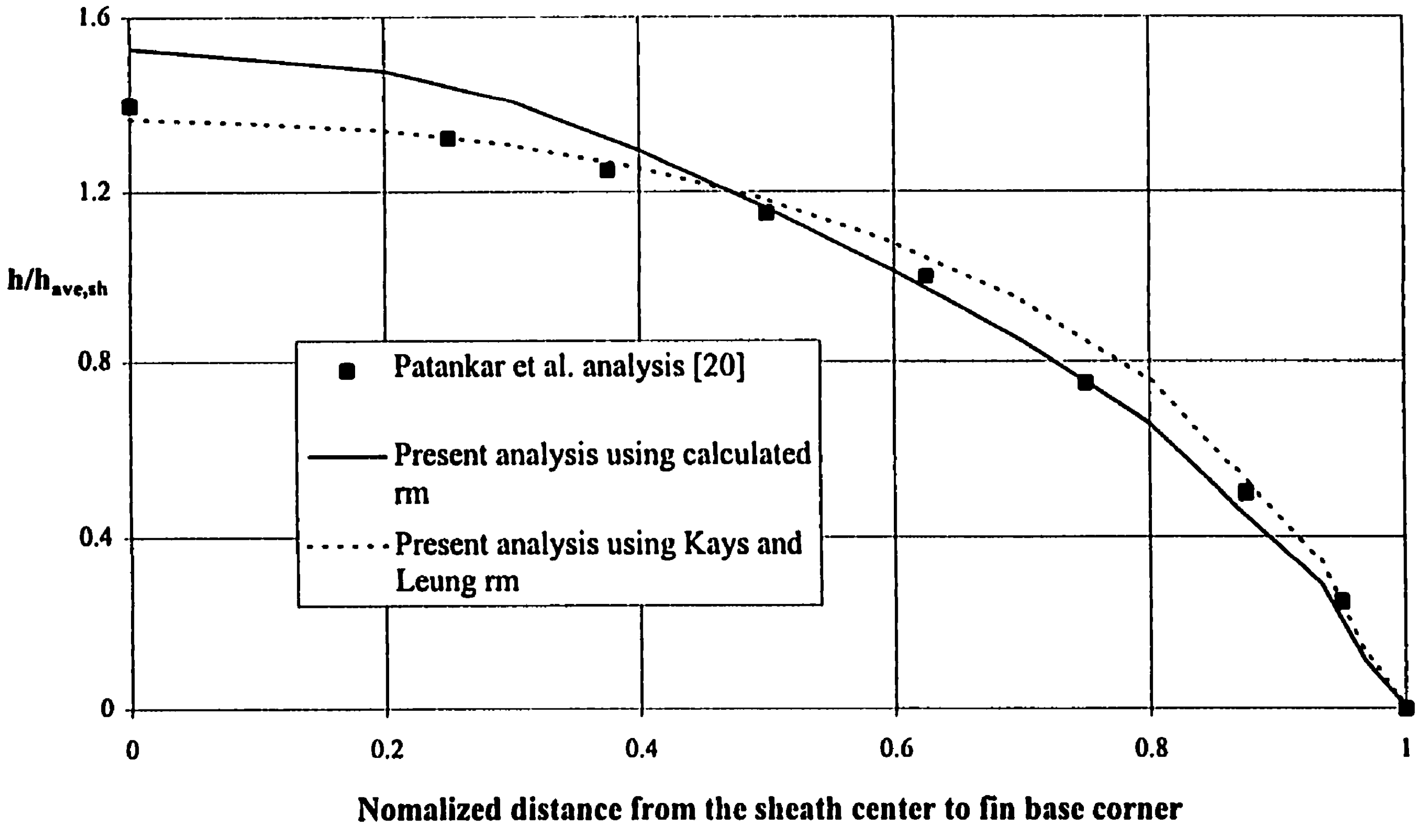


Figure 4.27 Comparison of sheath heat transfer coefficient with the analysis of Patankar et al. [20] ($r_o/r_i=2$, $N=12$, $H/(r_o-r_i)=0.4$, $Rc=1.0E4$ and $Pr=0.7$)

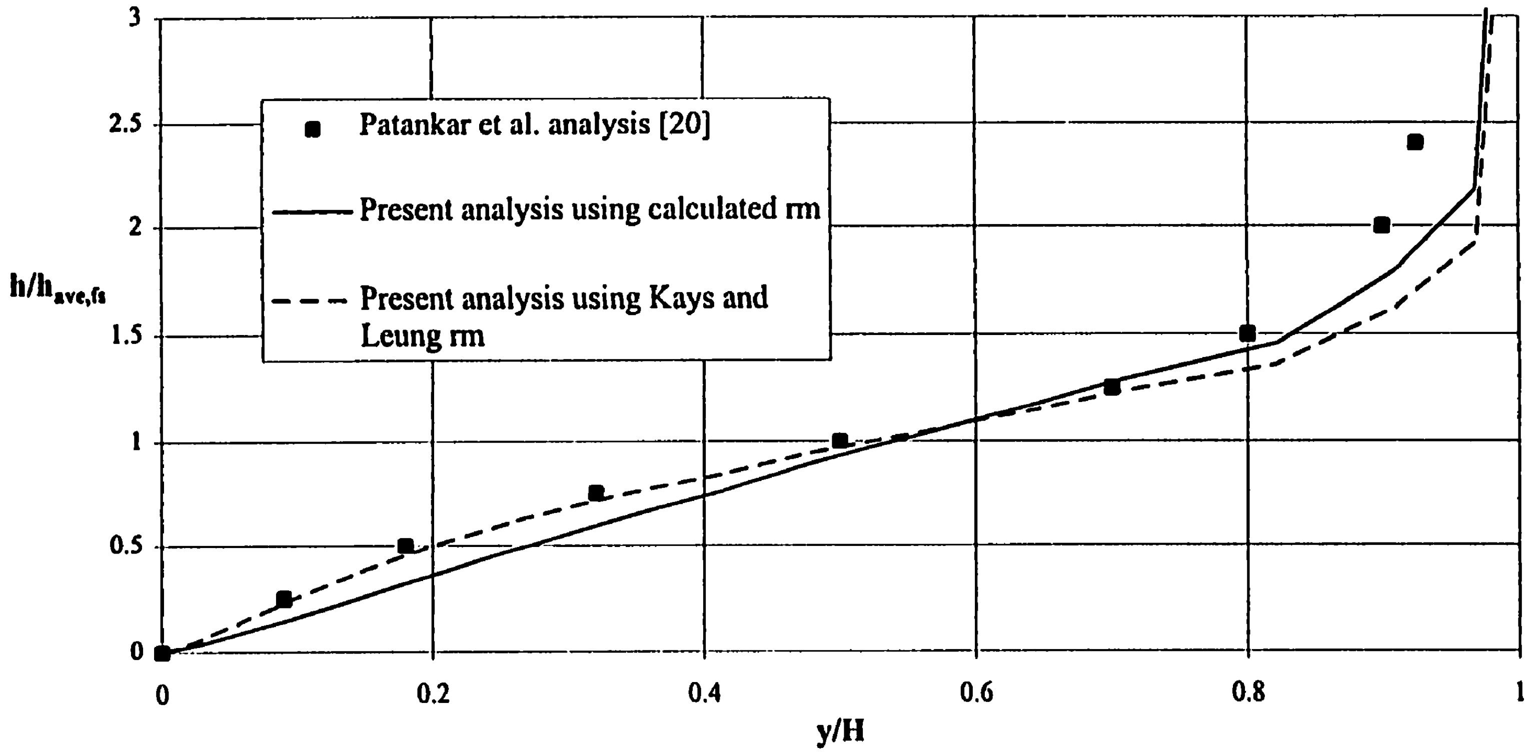


Figure 4.28 Comparison of fin side heat transfer coefficient with the analysis of Patankar et al. [20] (r_o/r_i=2, N=12, H/(r_o-r_i)=0.4, Re=1.0E4 and Pr=0.7)

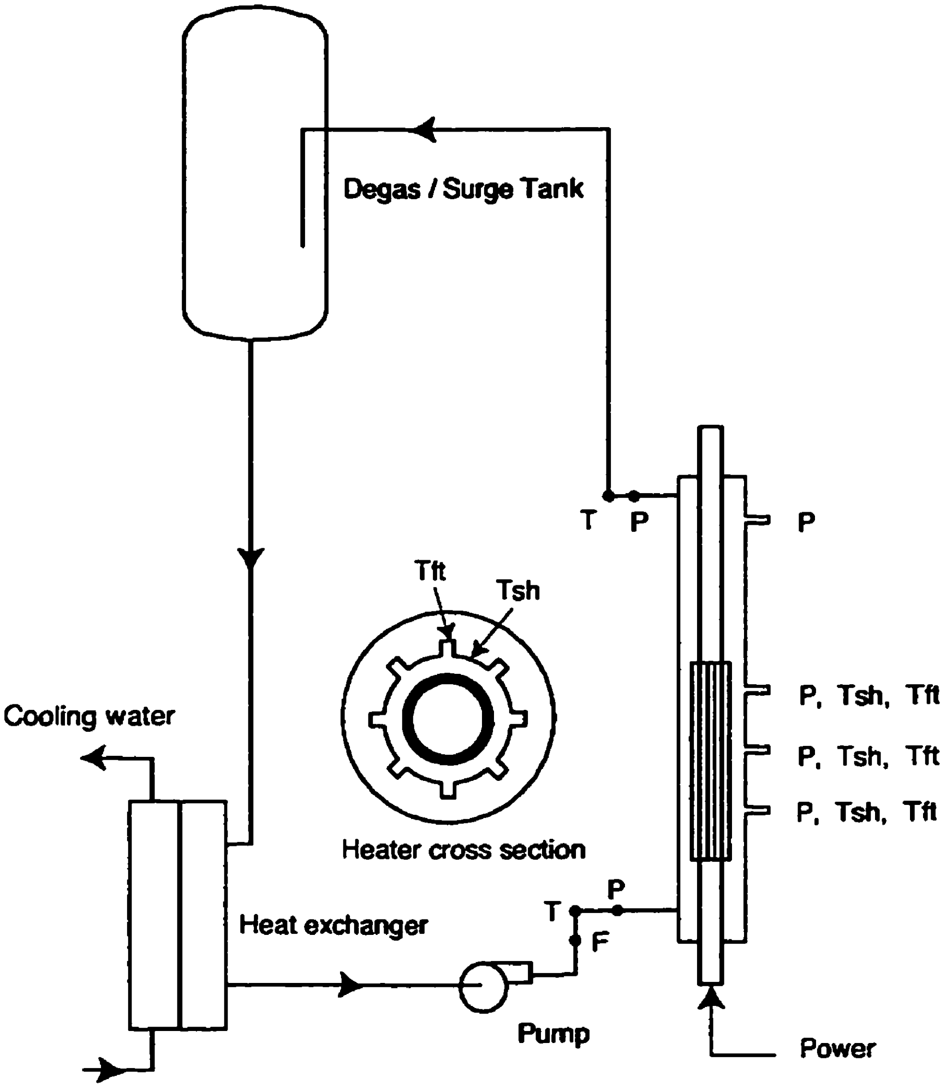


Figure 4.29 Schematic flow diagram of the test facility at AECL-WL (Reproduced from Reference 27 with minor changes)

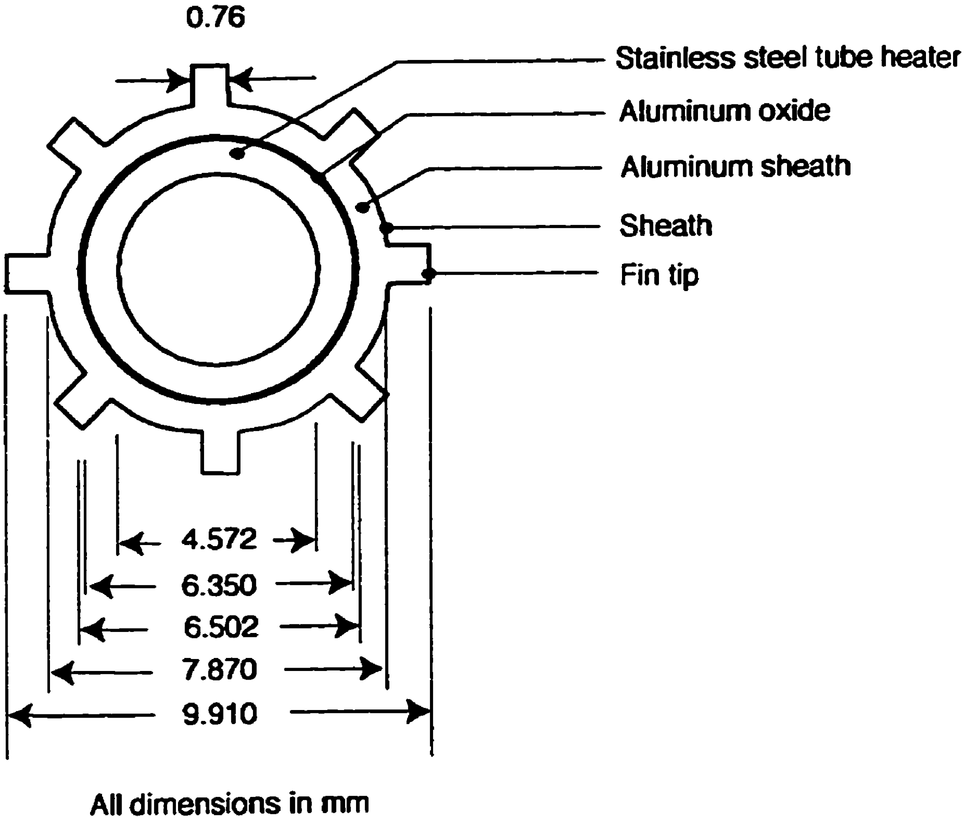


Figure 4.30 AECL finned heater geometry (Reproduced from Reference 27)

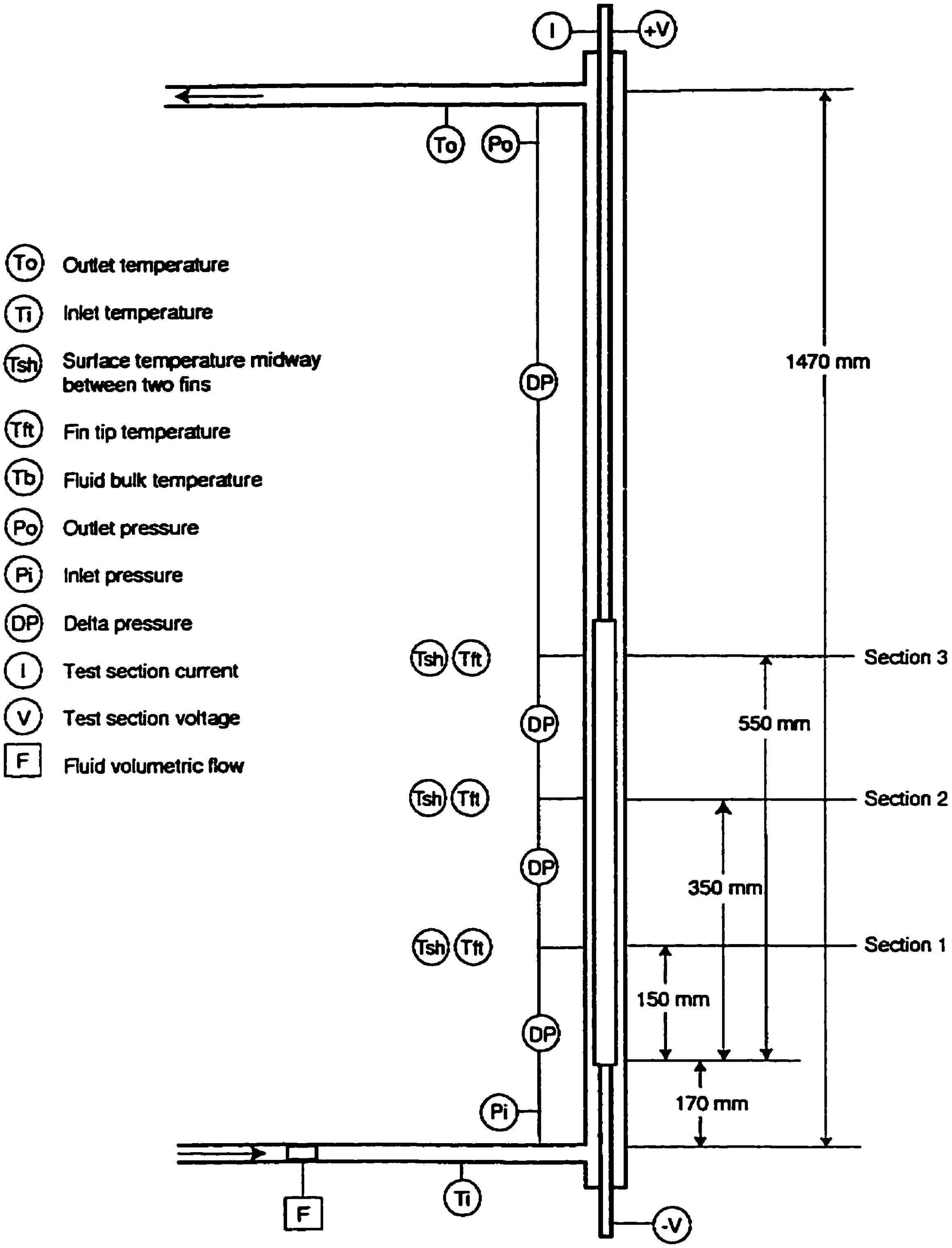


Figure 4.31 Instrumentation function and location
(Reproduced from Reference 27 with minor changes)

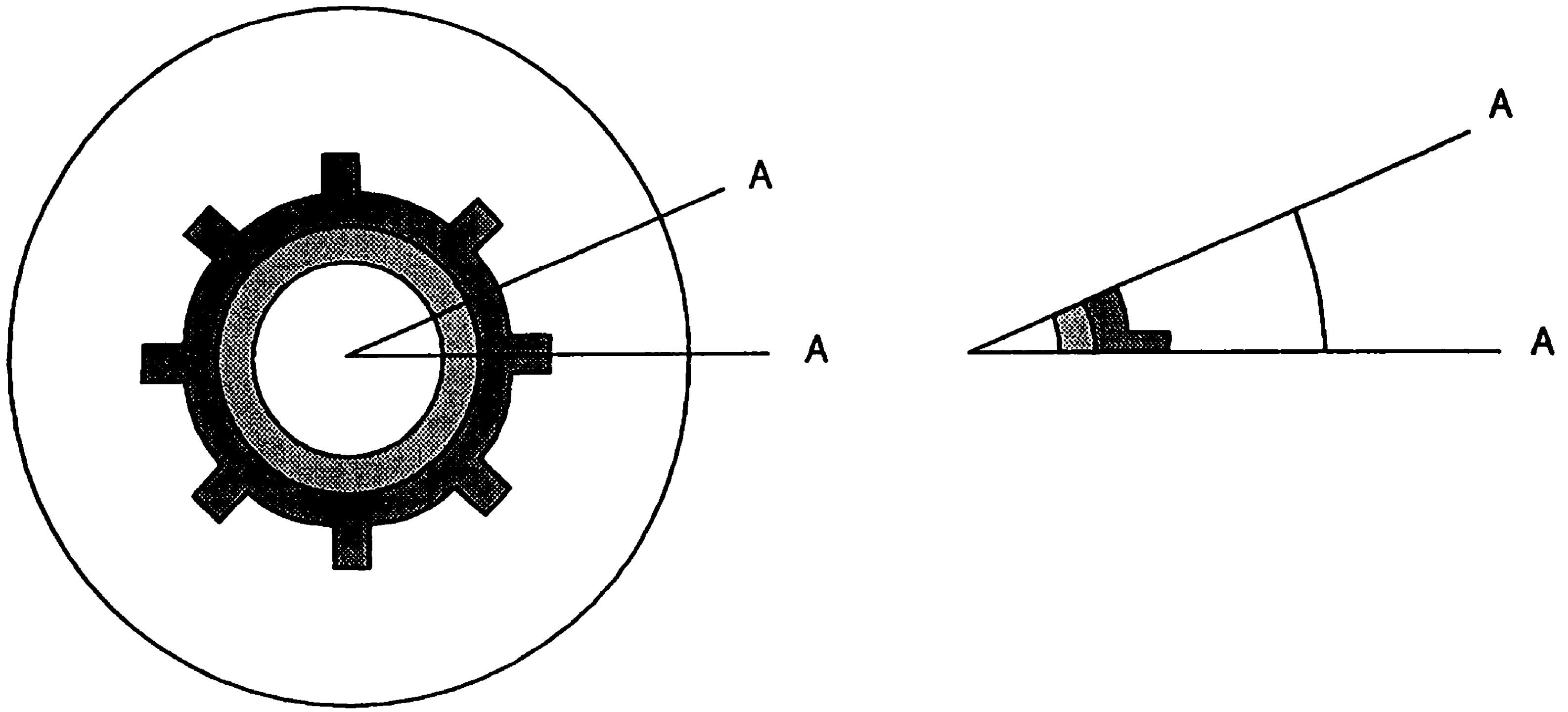


Figure 4.32 Schematic cross-section diagram of AECL finned annulus

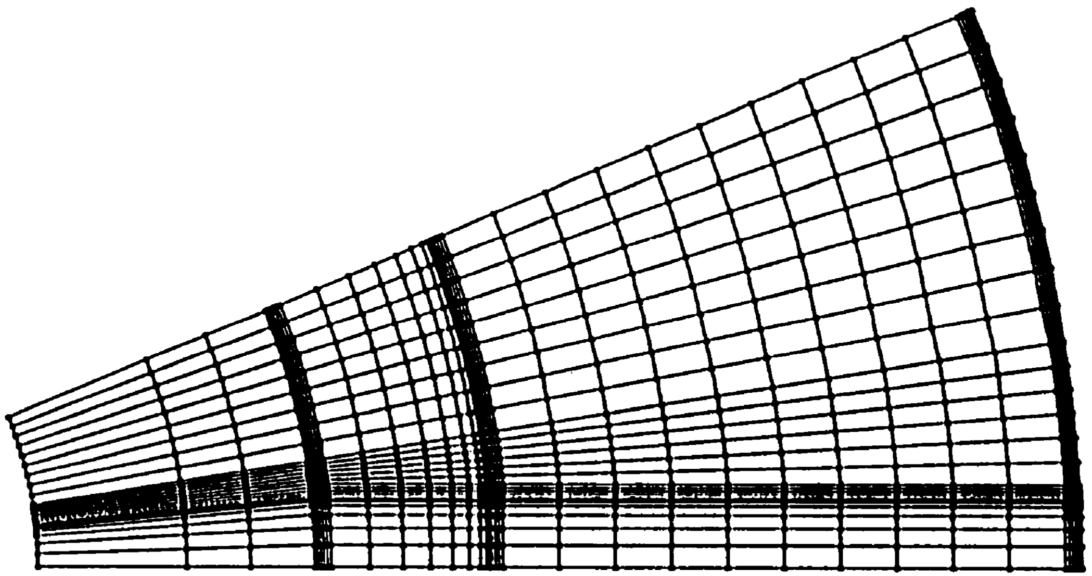


Figure 4.33 Grid used for the AECL finned annulus (1519 nodes and 1440 elements)

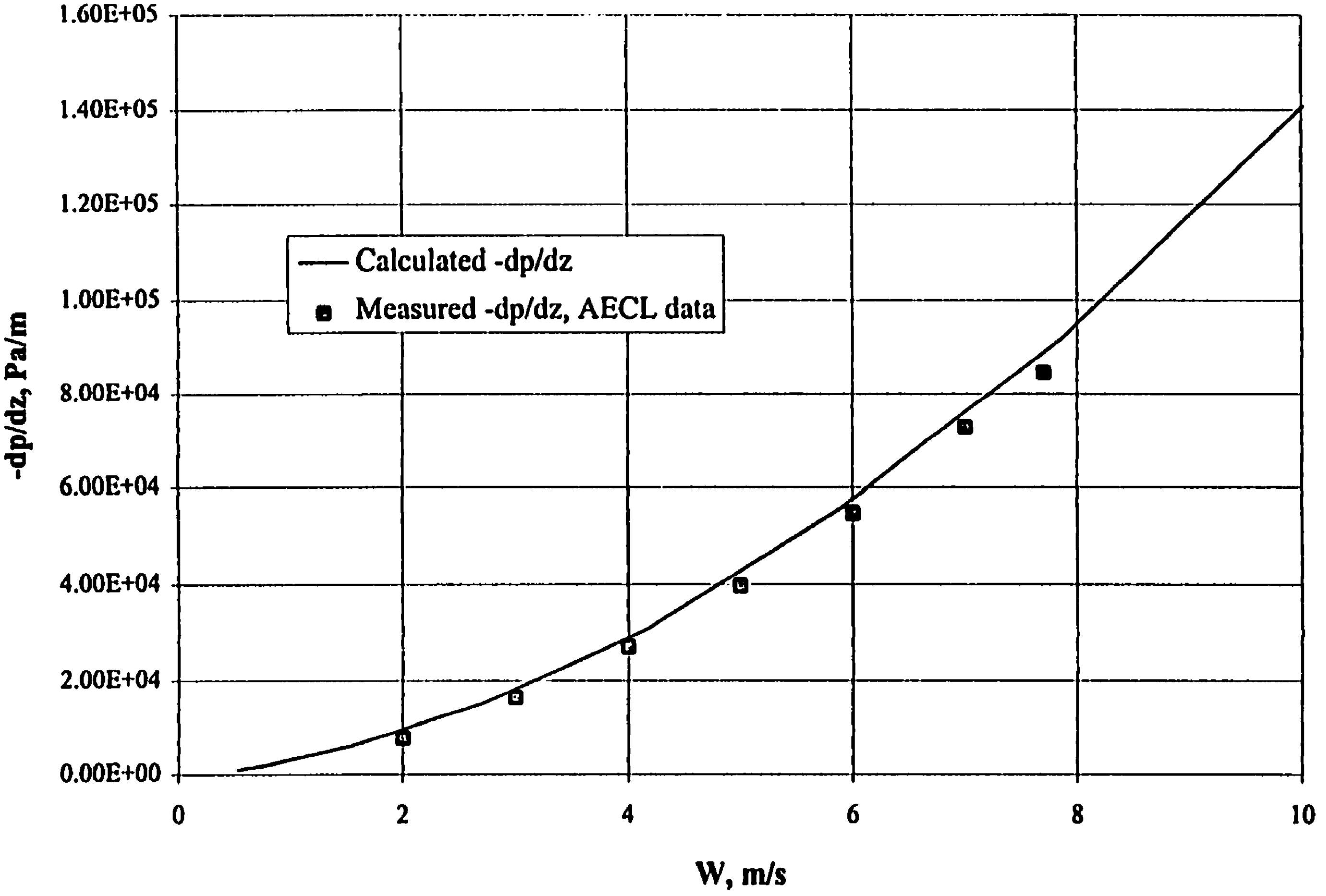


Figure 4.34 Comparison of present predictions of pressure drop with AECL data

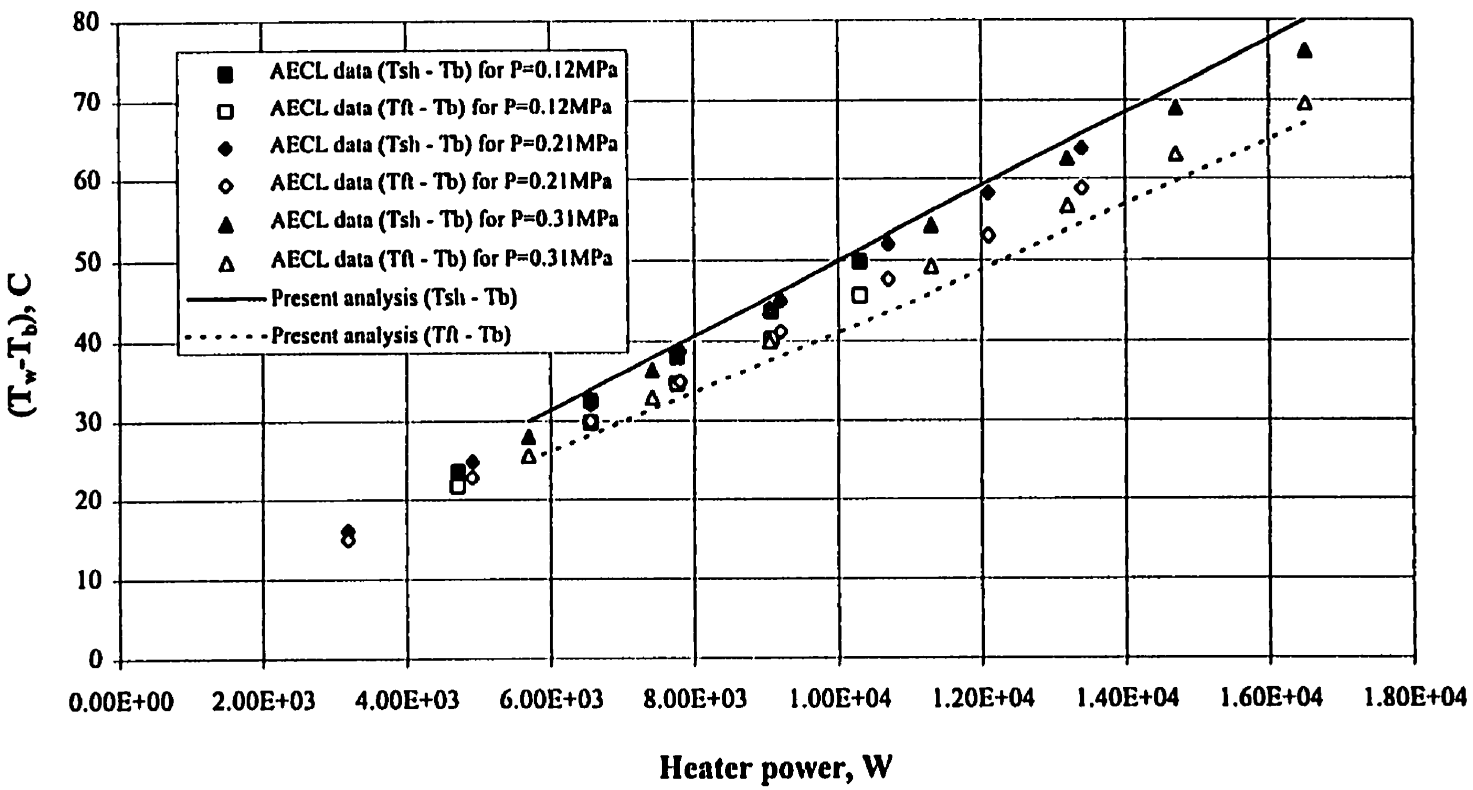


Figure 4.35 Comparison of wall temperatures with AECL data for W=1.2 m/s (Test numbers 138-157)

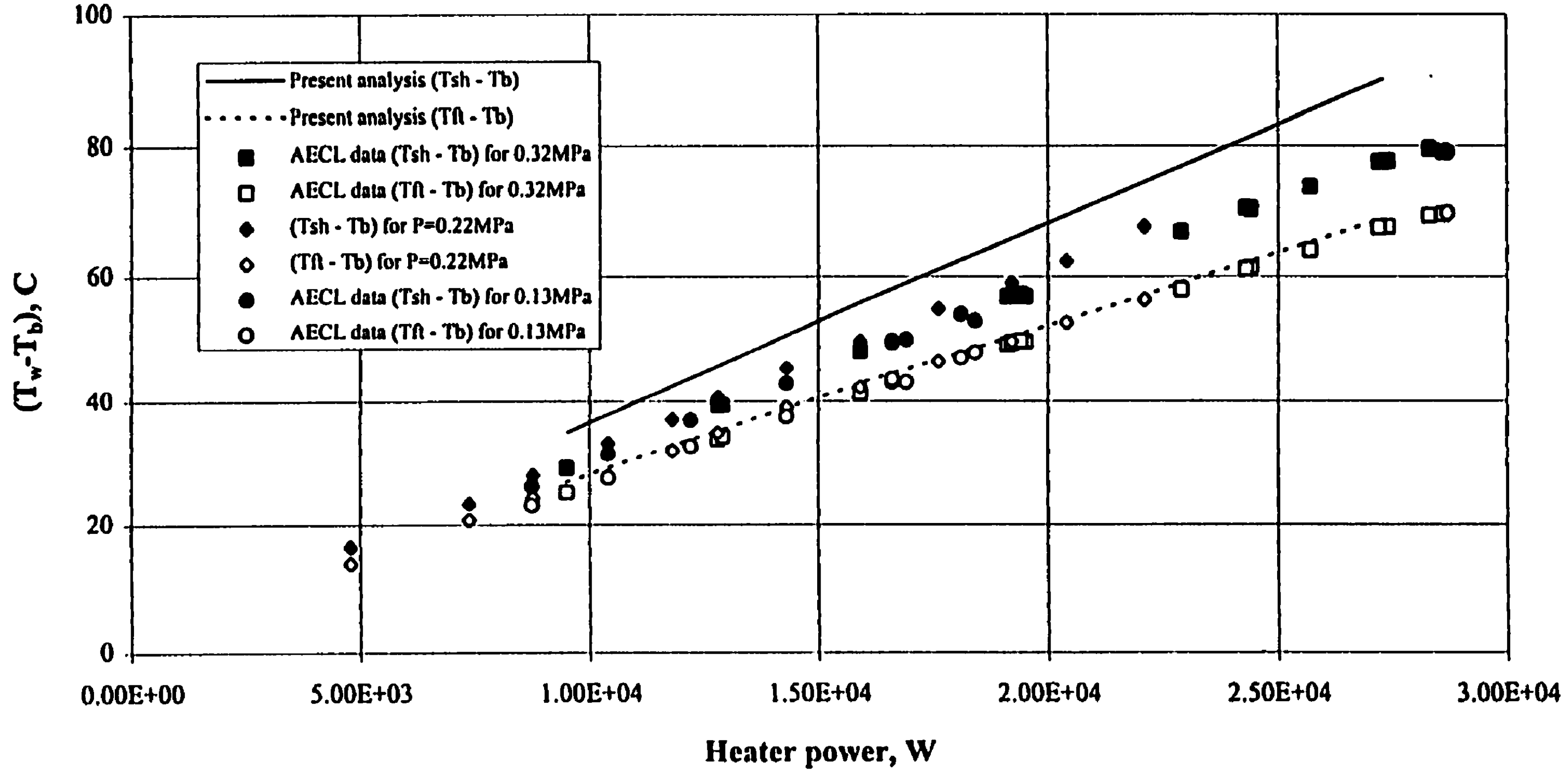


Figure 4.36 Comparison of wall temperatures with AECL data for W=2.0 m/s
(Test numbers 80-94, 95-104, 176-184, 190-202)

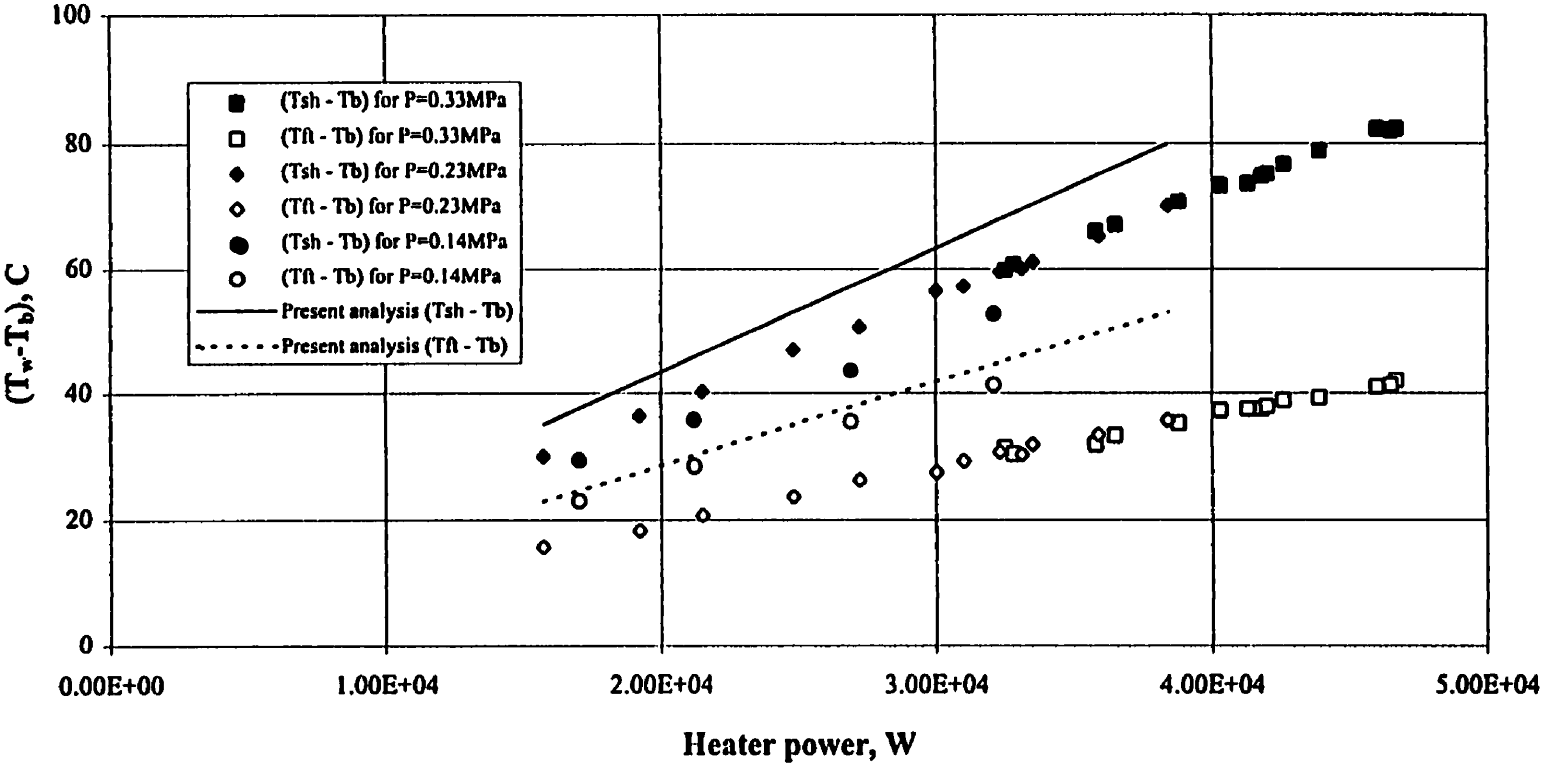


Figure 4.37 Comparison of wall temperatures with AECL data for W=4.1 m/s (Test numbers 234-263)

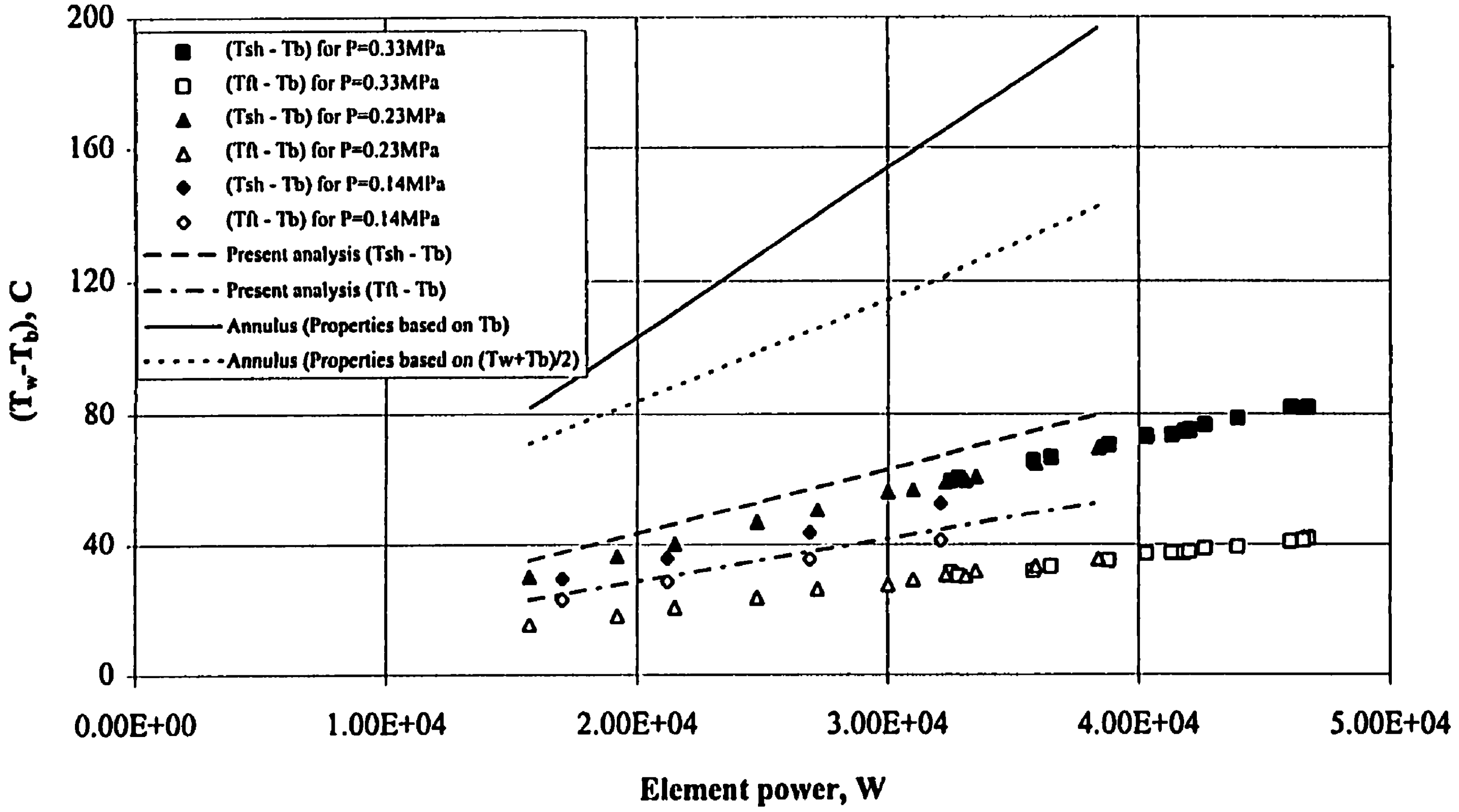


Figure 4.38 Comparison of wall temperatures with those of annulus for W=4.1 m/s (Test numbers 234-263)

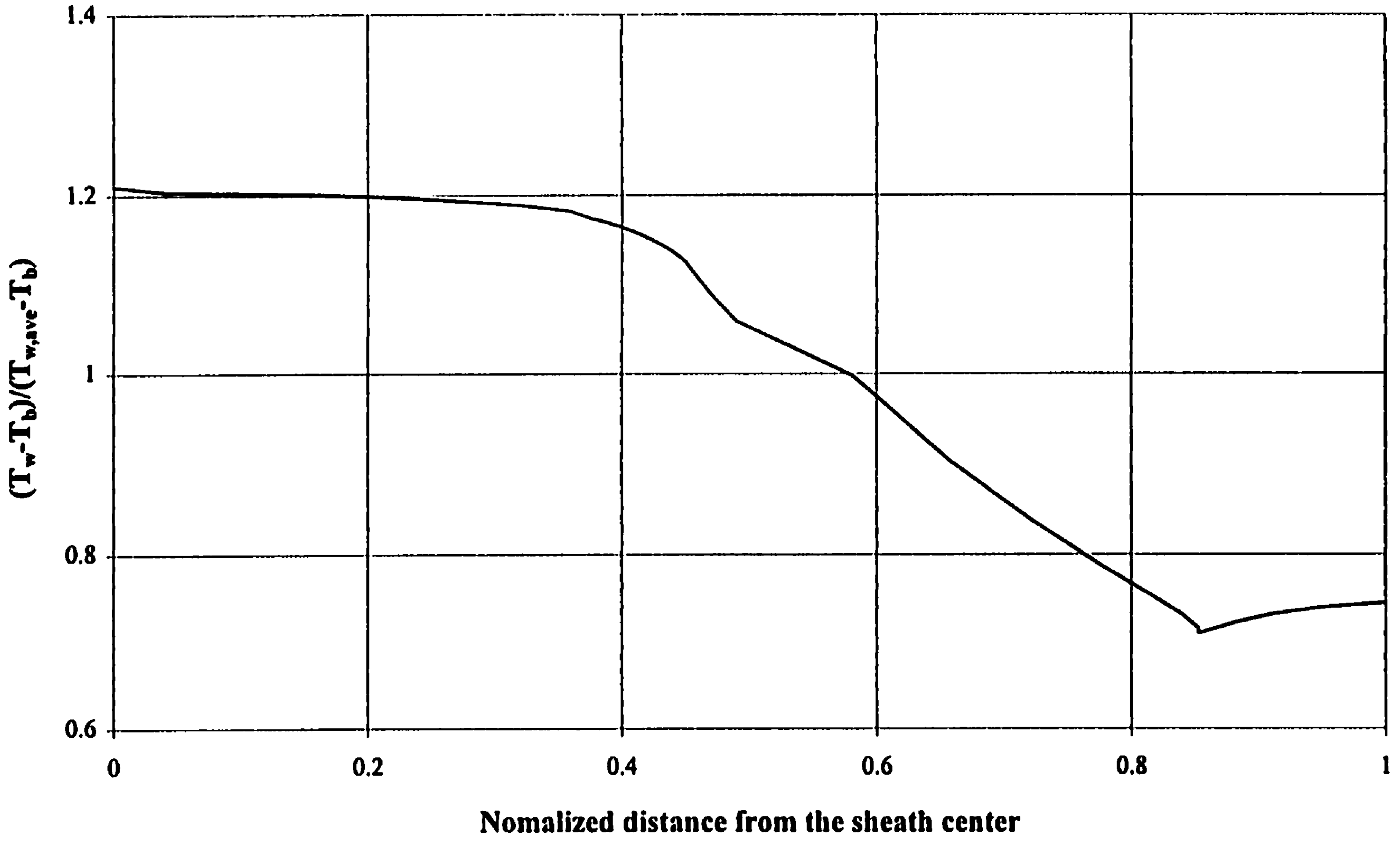


Figure 4.39 Local temperature distribution along the finned surface (Test number 277)

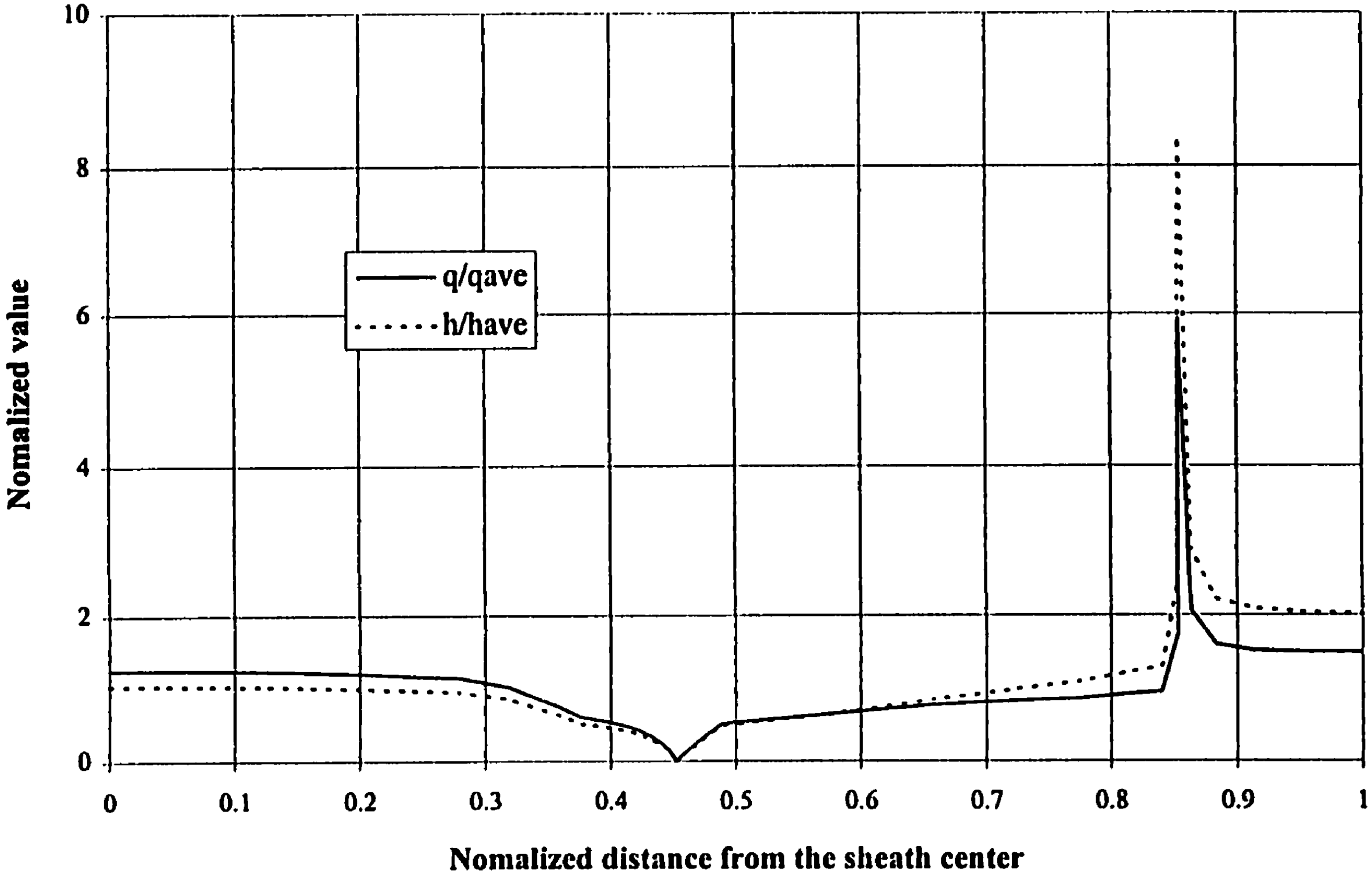


Figure 4.40 Calculated local distributions of h and q along the finned surface
(Test number 277)

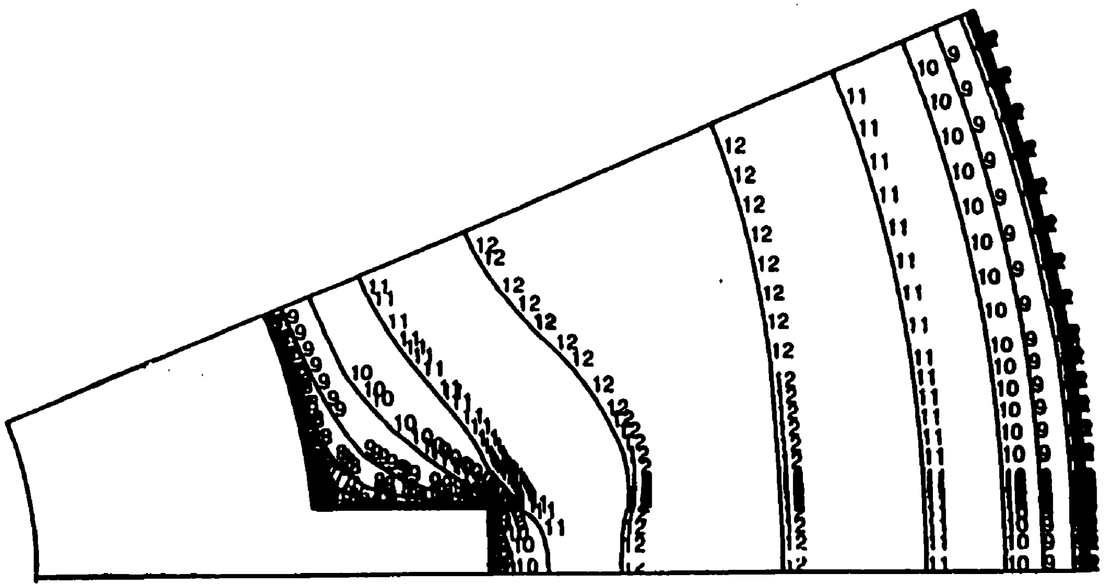


Figure 4.41 Predicted velocity distribution of test number 277 (1=0.1 m/s, 12=5.6 m/s, increment=0.5 m/s)

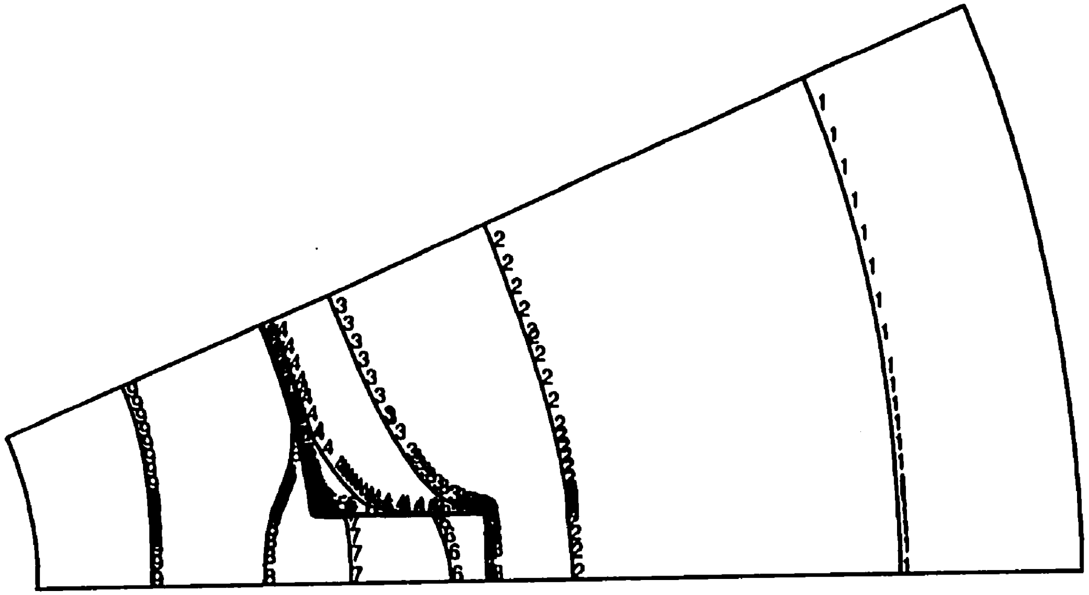


Figure 4.42 Predicted temperature distribution of test number 277 (1=50°C, 9=170°C, increment=15°C)

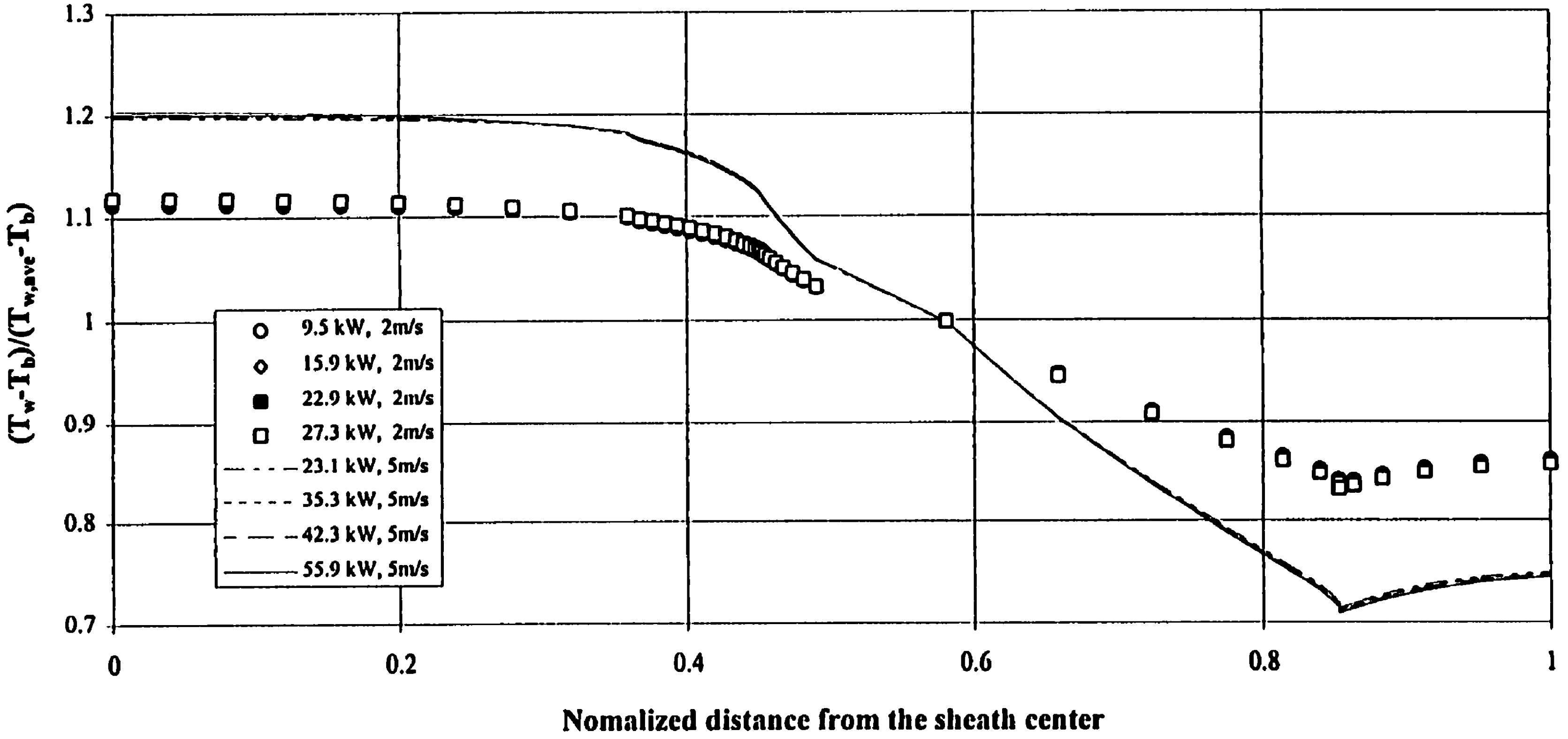


Figure 4.43 Effect of heat generation rate and velocity on temperature distribution (Test numbers 176-183, 264-277)

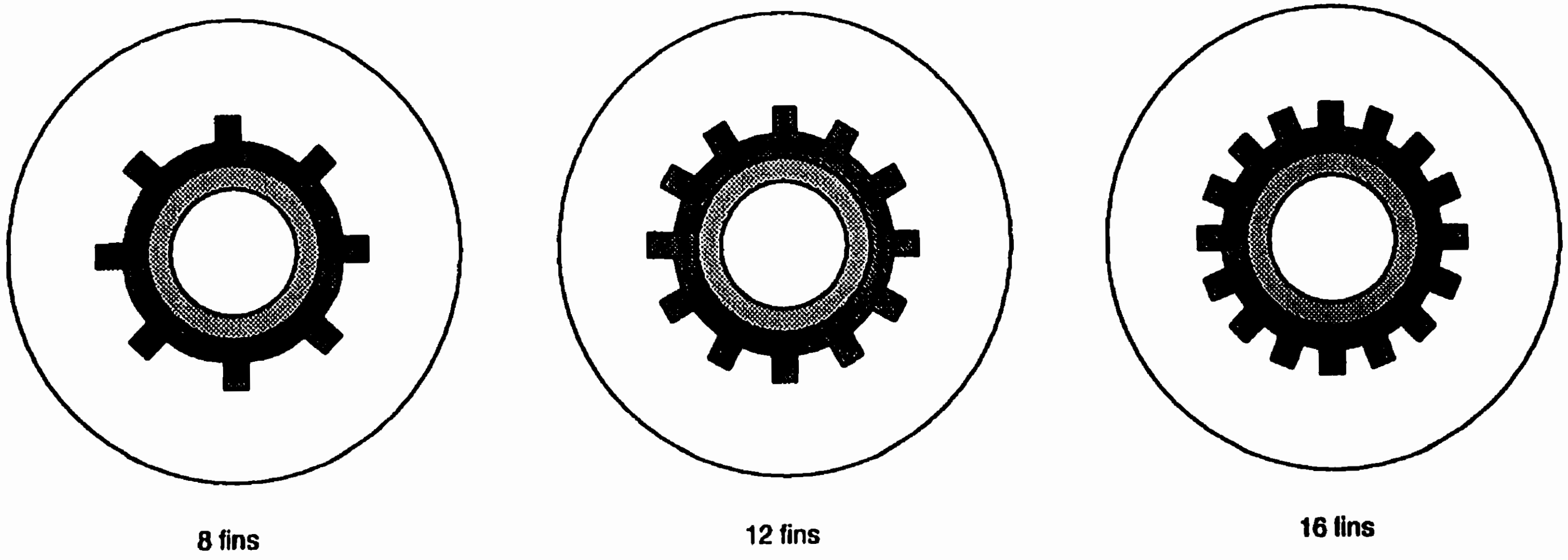


Figure 4.44 Simulated finned annulus geometries of N=8, 12 and 16 with $H/(r_o\text{-}r_i)$=0.22

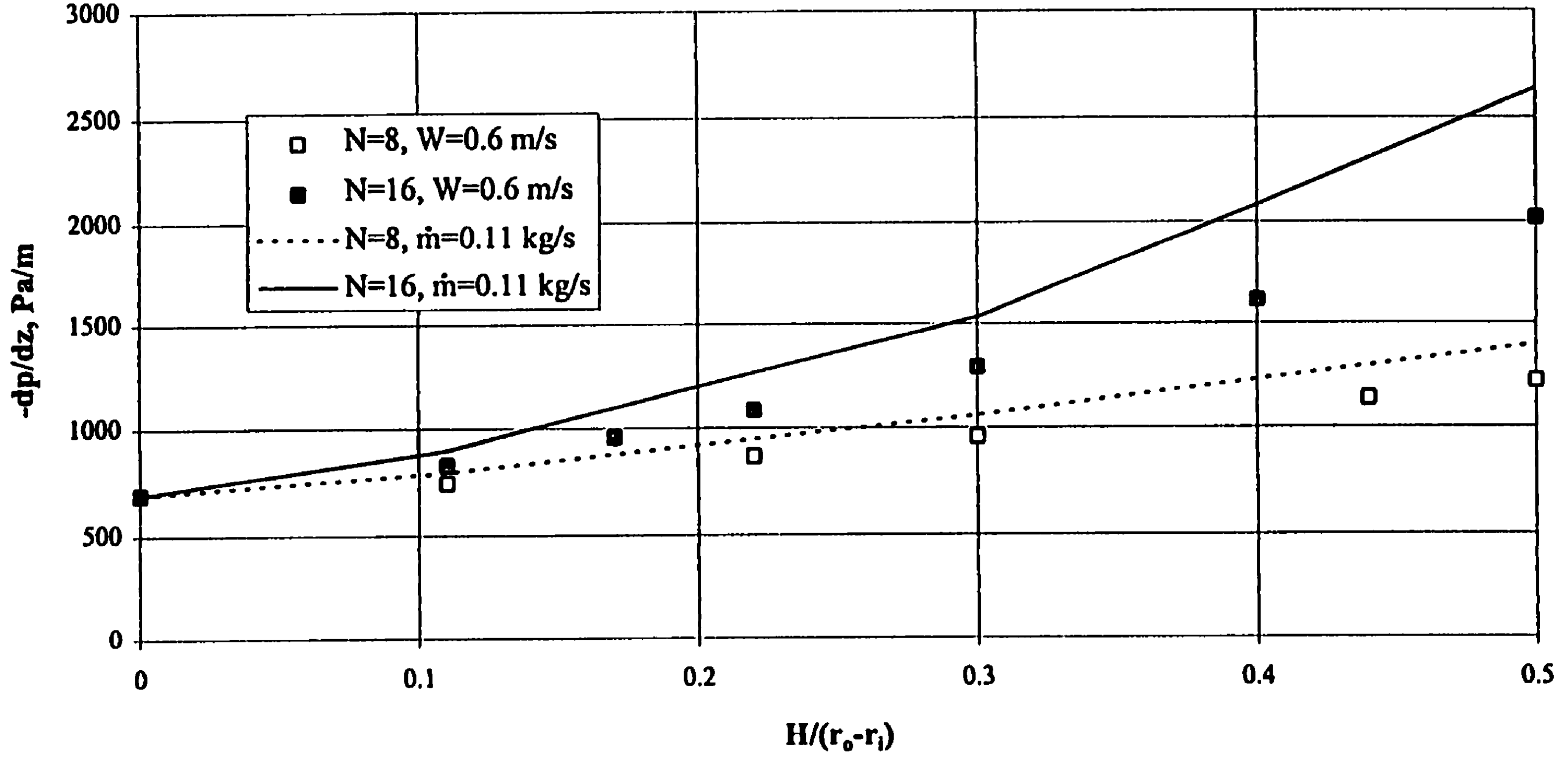


Figure 4.45 Effect of fin geometry on pressure drop for W=0.6 m/s and ṁ=0.11 kg/s

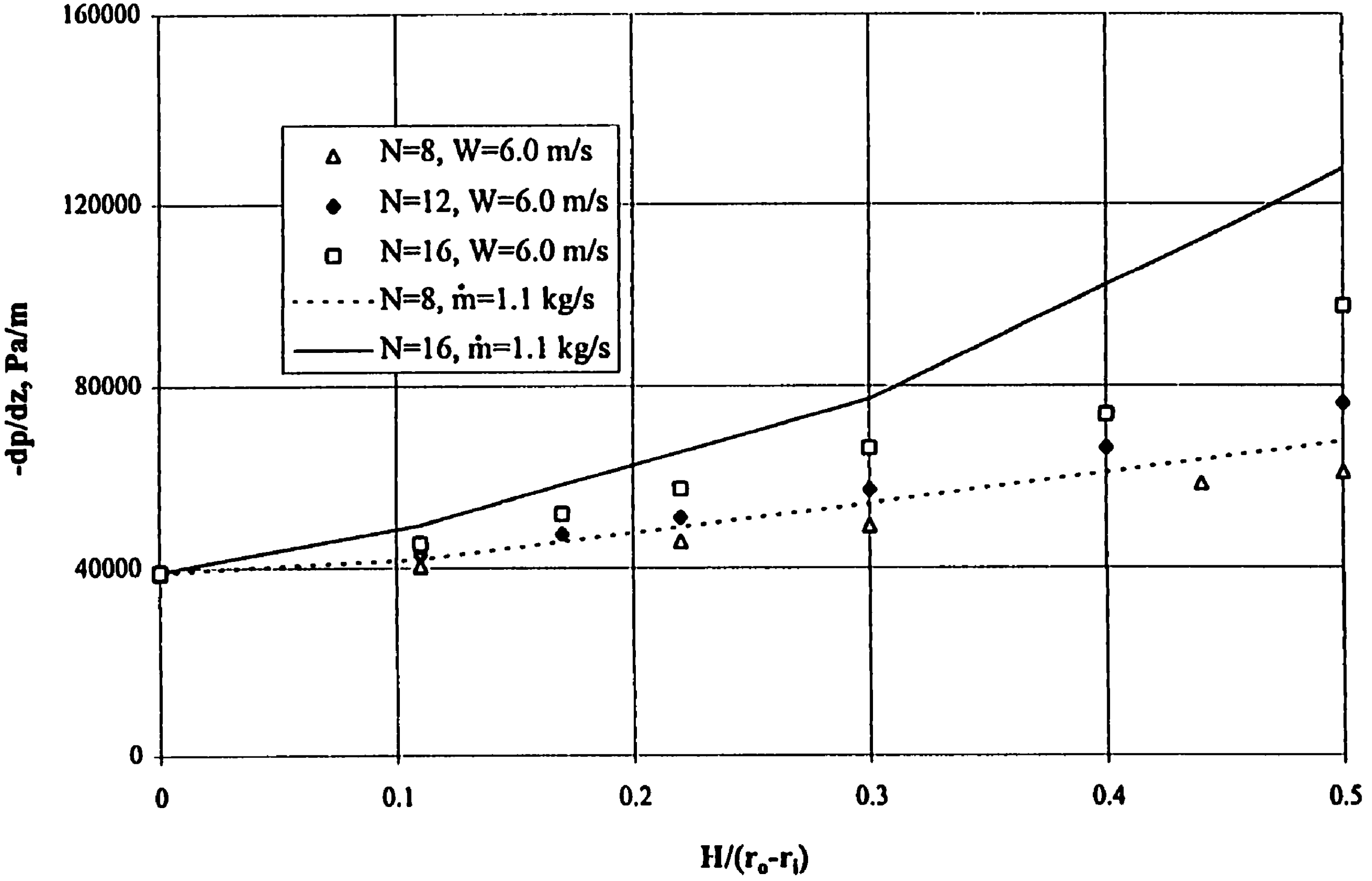


Figure 4.46 Effect of fin geometry on pressure drop for W=6.0 m/s and $\dot{m}$=1.1 kg/s

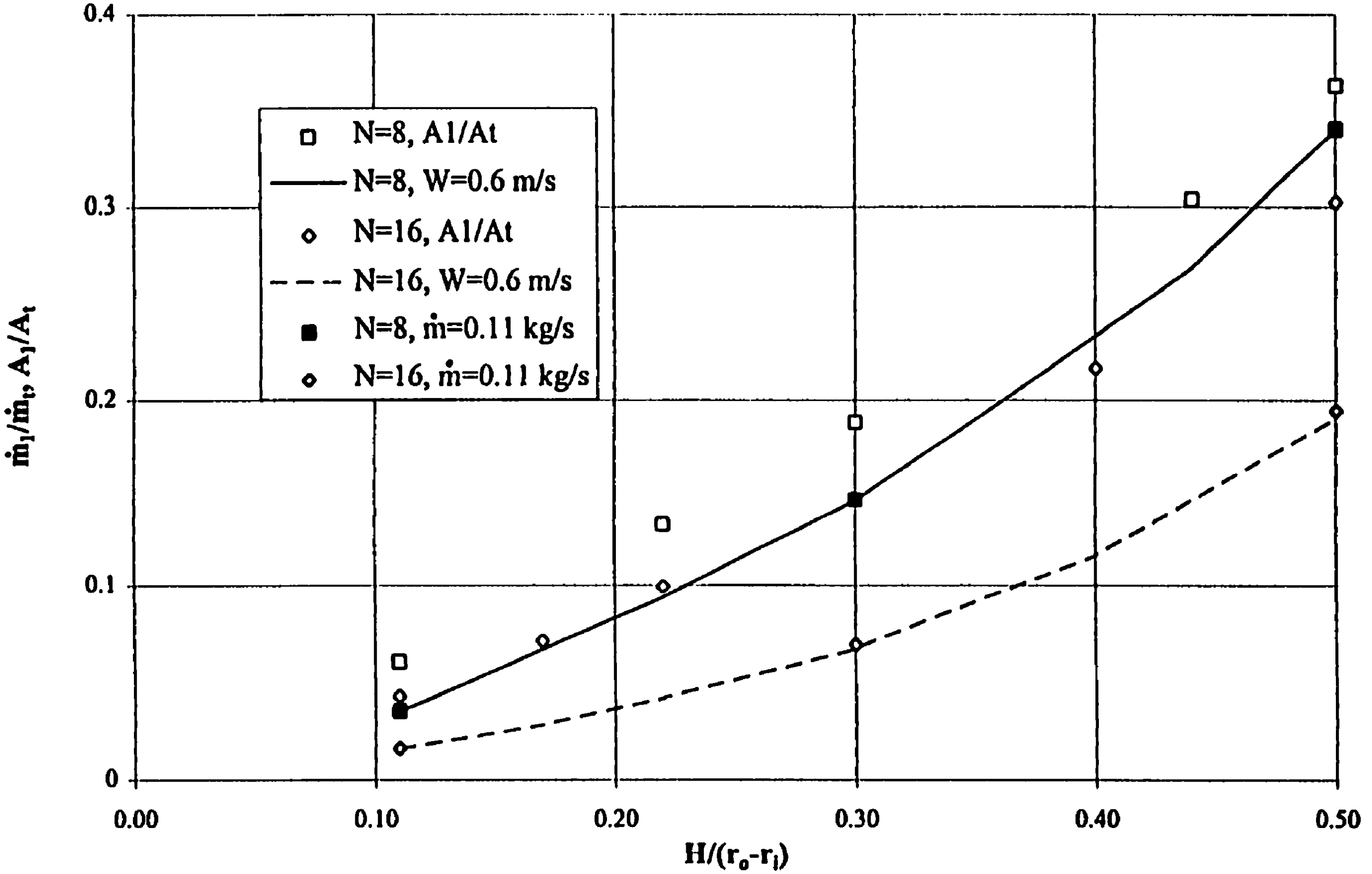


Figure 4.47 Effect of fin geometry on flow split for W=0.6 m/s and $\dot{m}$=0.11 kg/s

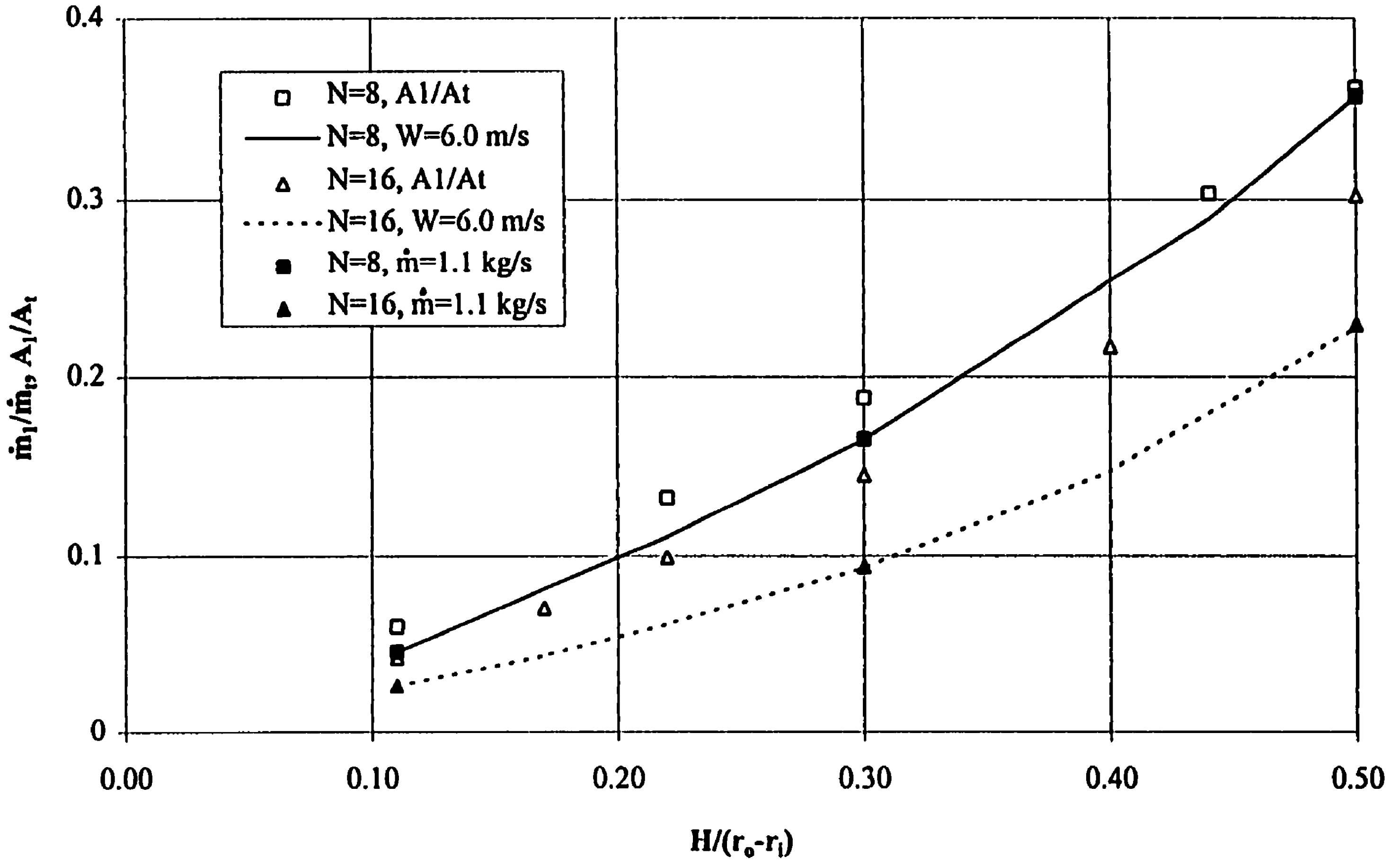


Figure 4.48 Effect of fin geometry on flow split for W=6.0 m/s and $\dot{m}$=1.1 kg/s

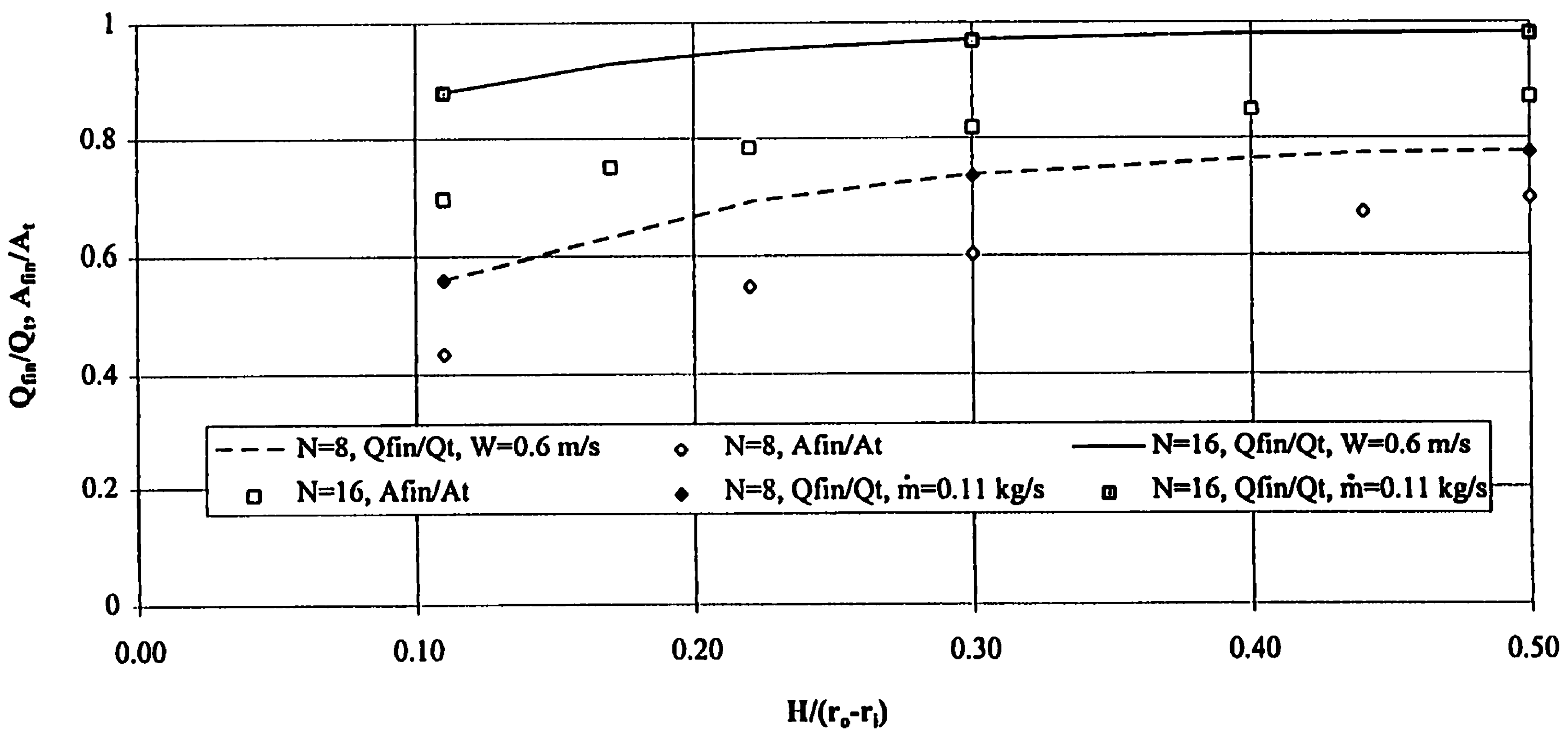


Figure 4.49 Effect of fin geometry on heat split for W=0.6 m/s and $\dot{m}$=0.11 kg/s

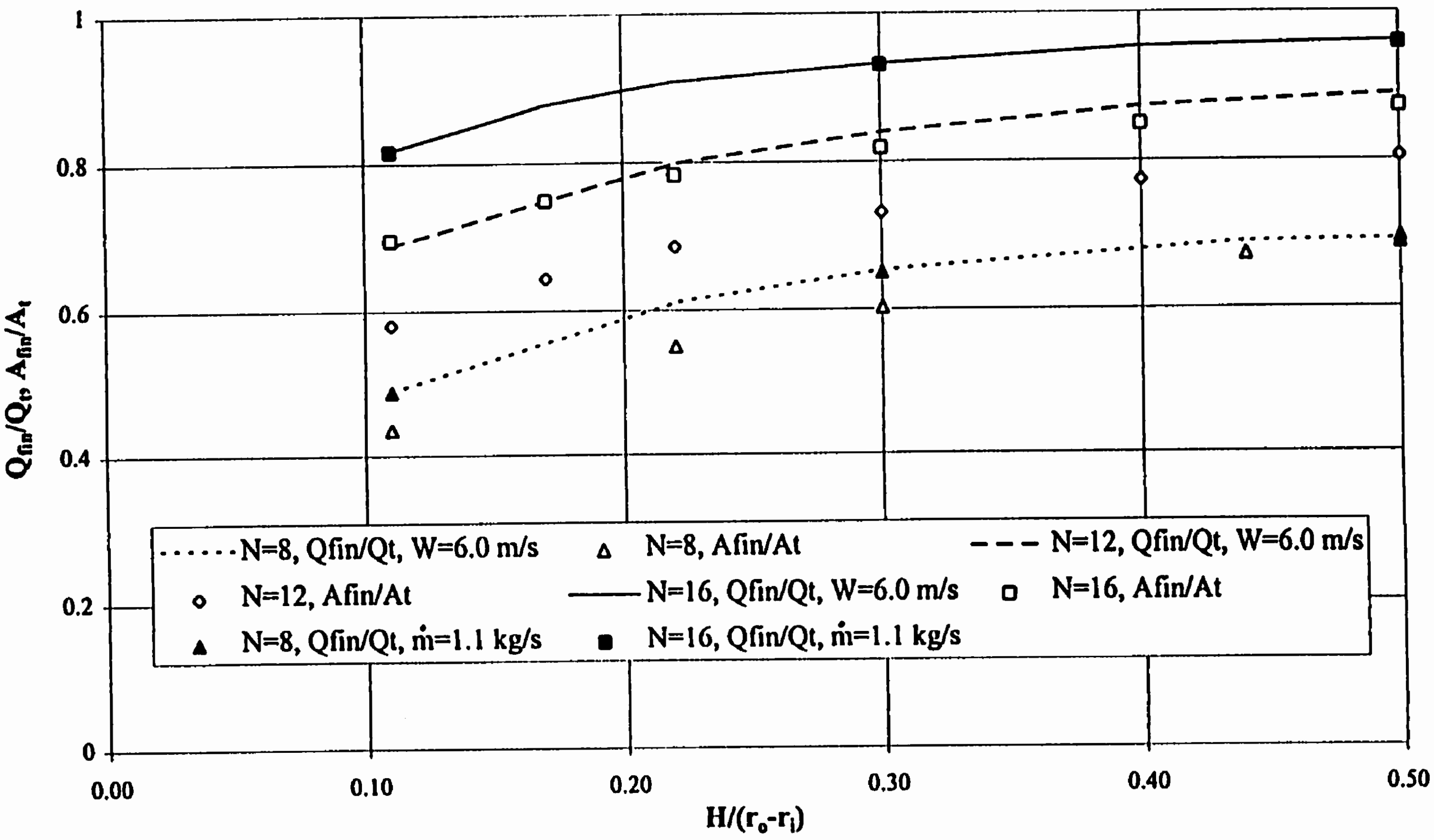


Figure 4.50 Effect of fin geometry on heat split for W=6.0 m/s and ṁ=1.1 kg/s

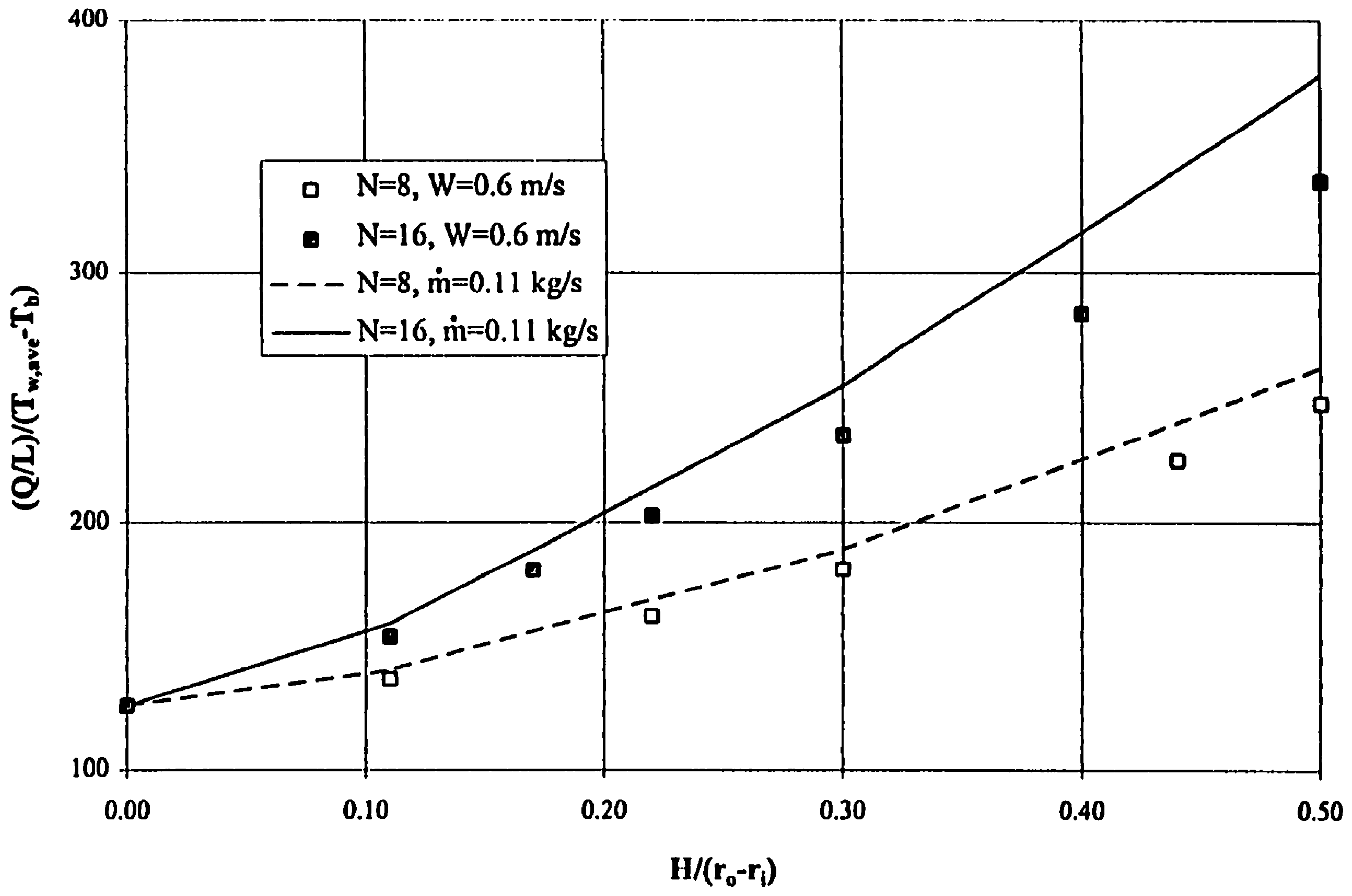


Figure 4.51 Effect of fin geometry on heat transfer rate for W=0.6 m/s and ṁ=0.11 kg/s

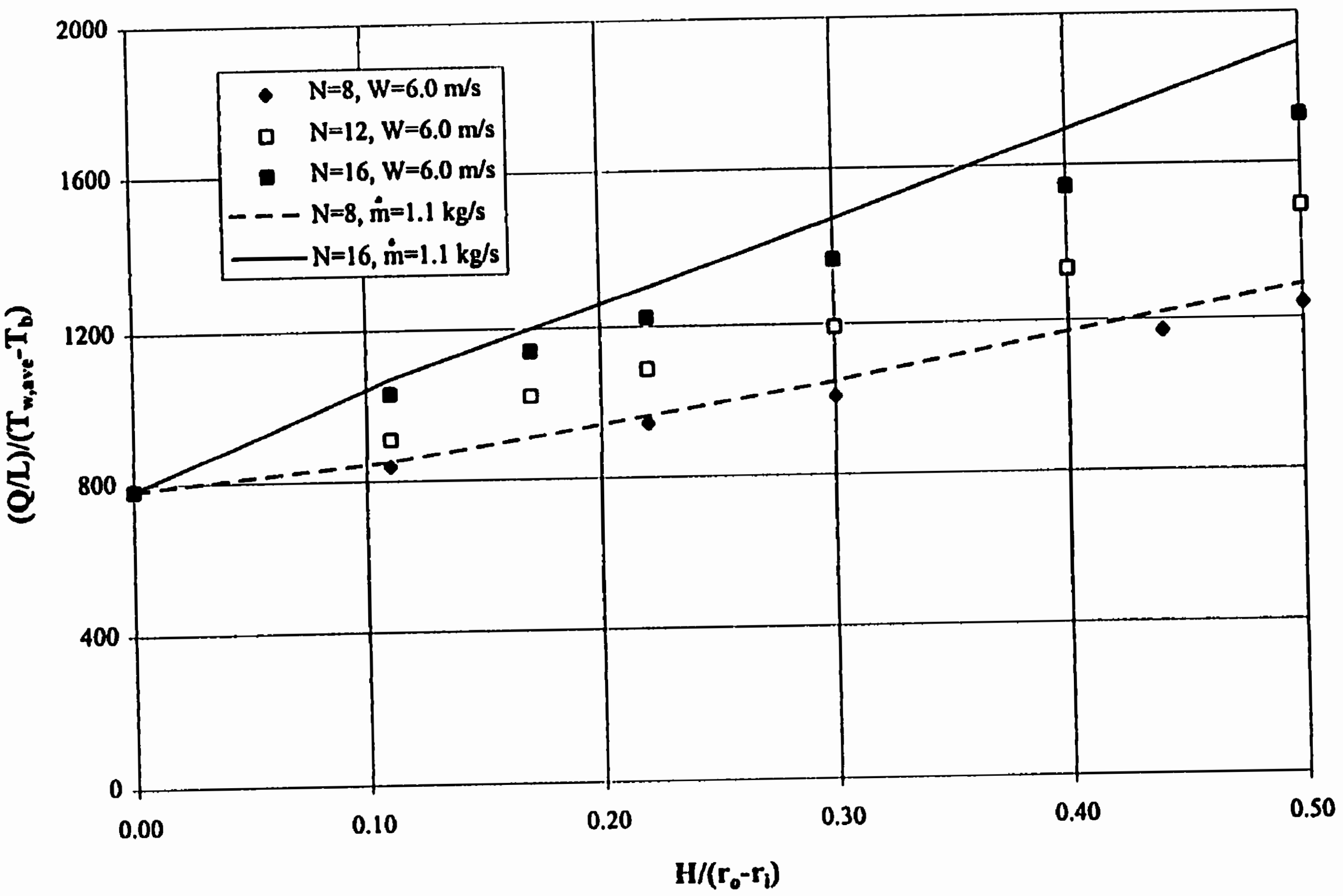


Figure 4.52 Effect of fin geometry on heat transfer rate for W=6.0 m/s and $\dot{m}$=1.1 kg/s

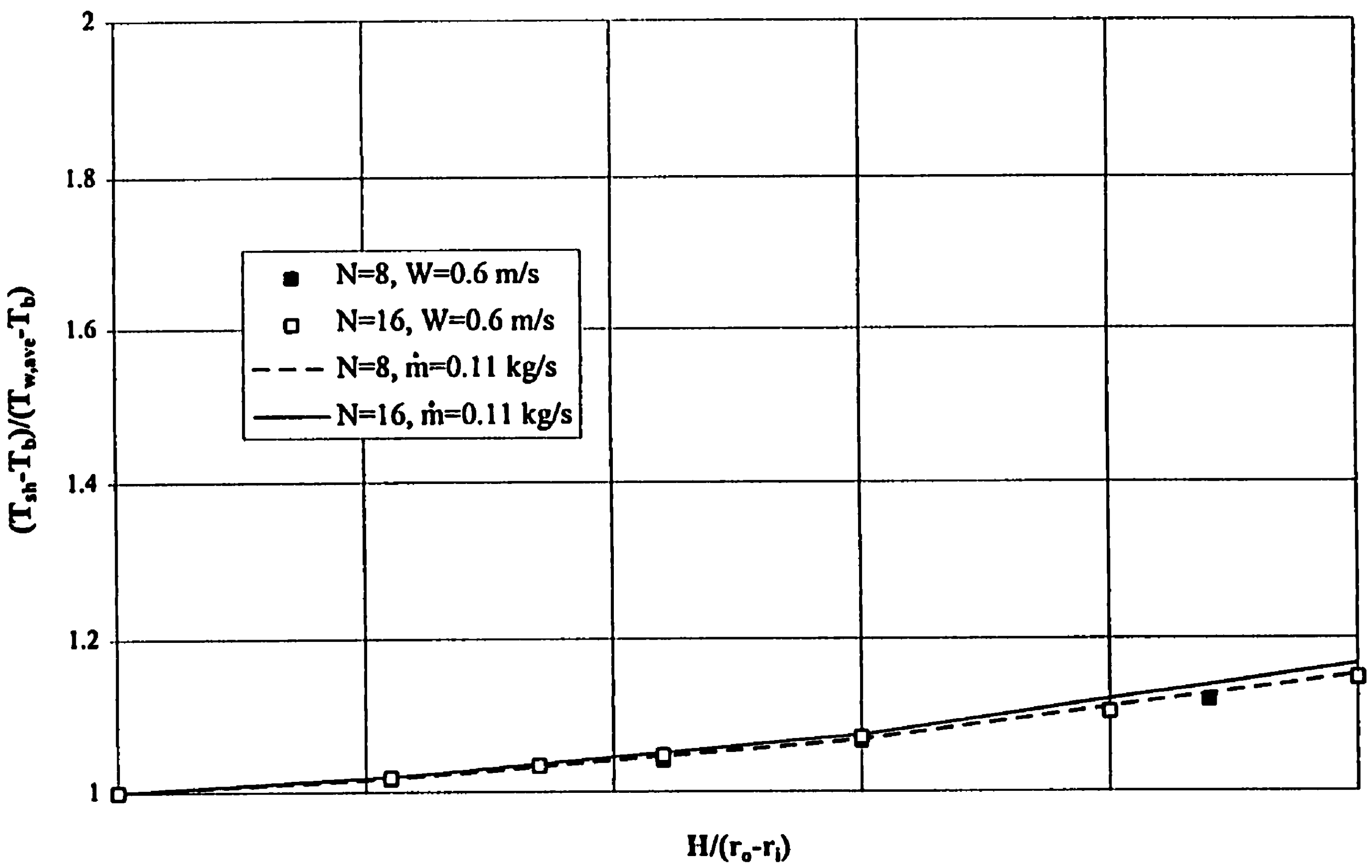


Figure 4.53 Effect of fin geometry on wall temperature for W=0.6 m/s and $\dot{m}$=0.11 kg/s

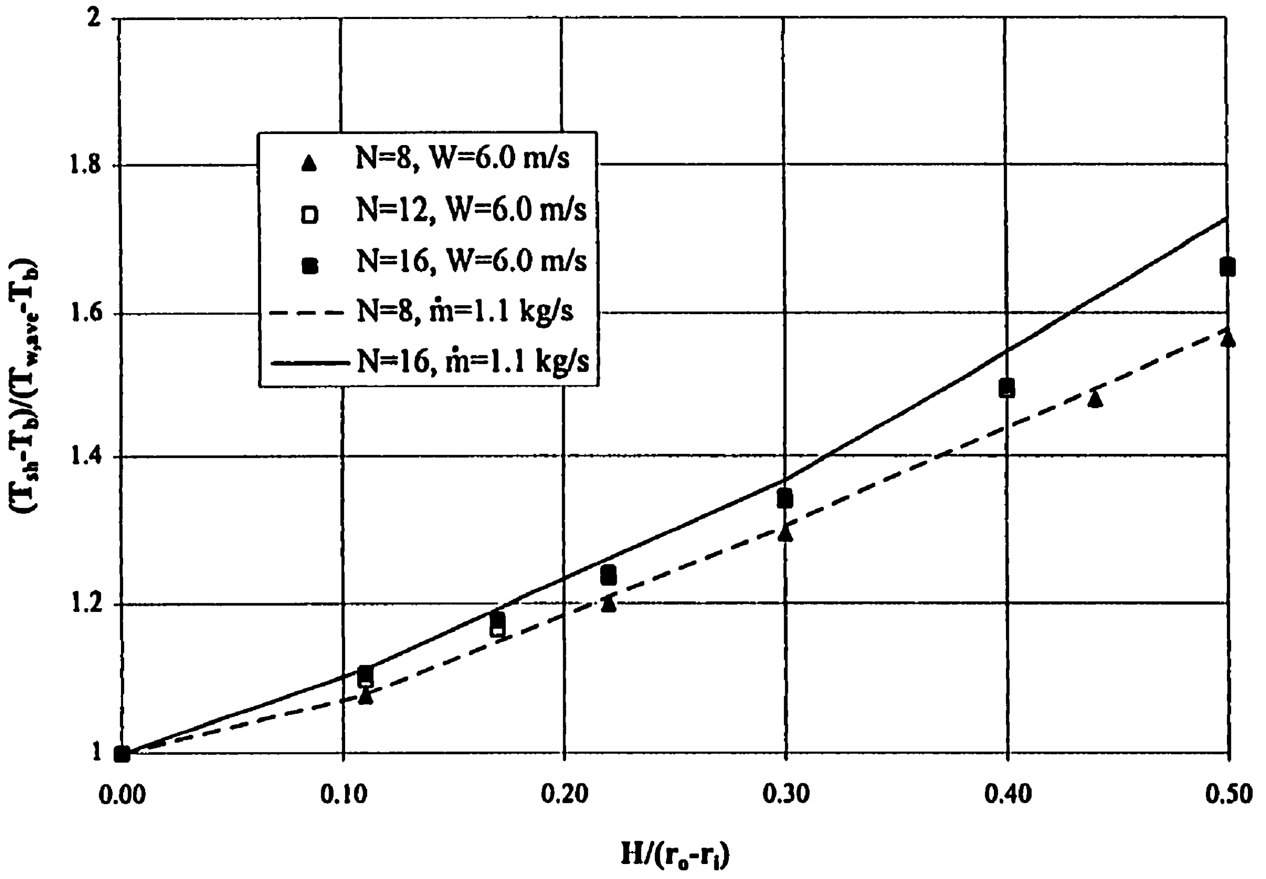
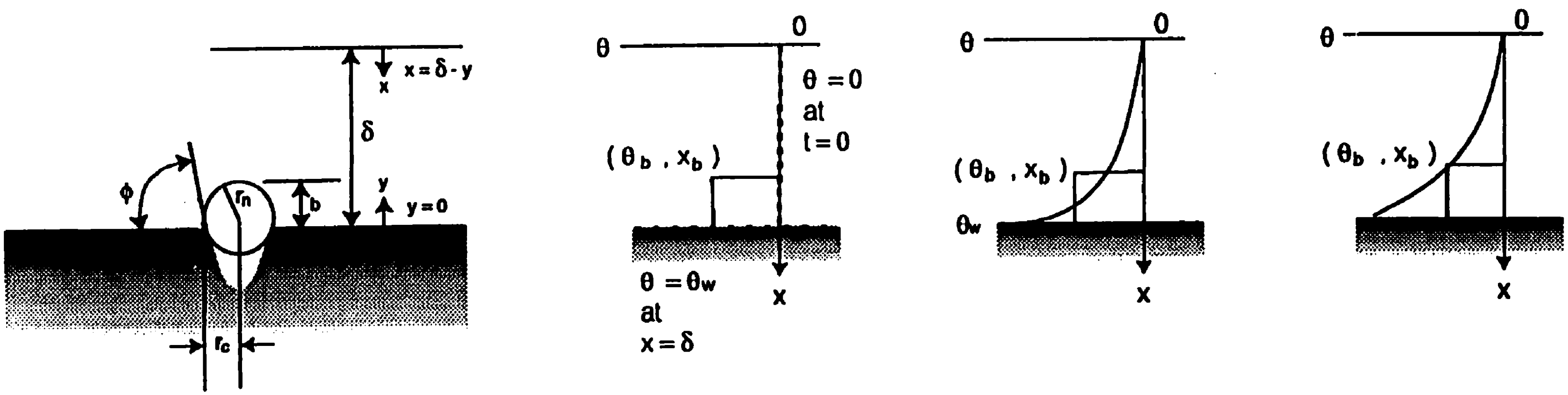


Figure 4.54 Effect of fin geometry on wall temperature for W=6.0 m/s and $\dot{m}$=1.1 kg/s



(a) Bubble nucleus at cavity mouth.

(b) Temperature profile at t = 0.
Beginning of waiting period
θ = 0 at all x except at wall.

(c) Temperature profile at
$0 < t < t_w$.
During waiting period the
surrounding liquid is still
cooler than bubble interior.

(d) End of waiting period;
θ(x) curve passes
point (θ_b, x_b).

Figure 5.1 Model for waiting period and temperature profiles (Reproduced from Reference 33)

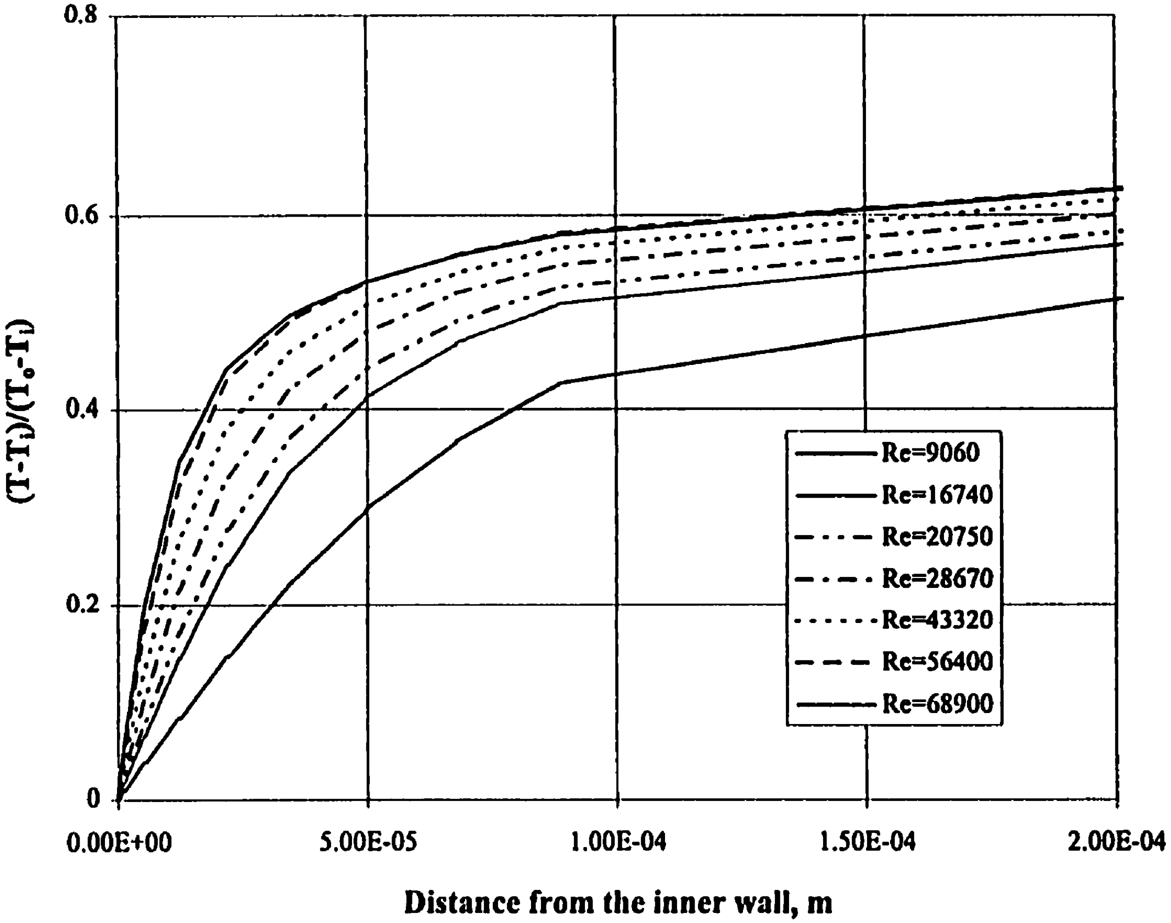


Figure 5.2 Temperature profiles inside the radius of maximum velocity

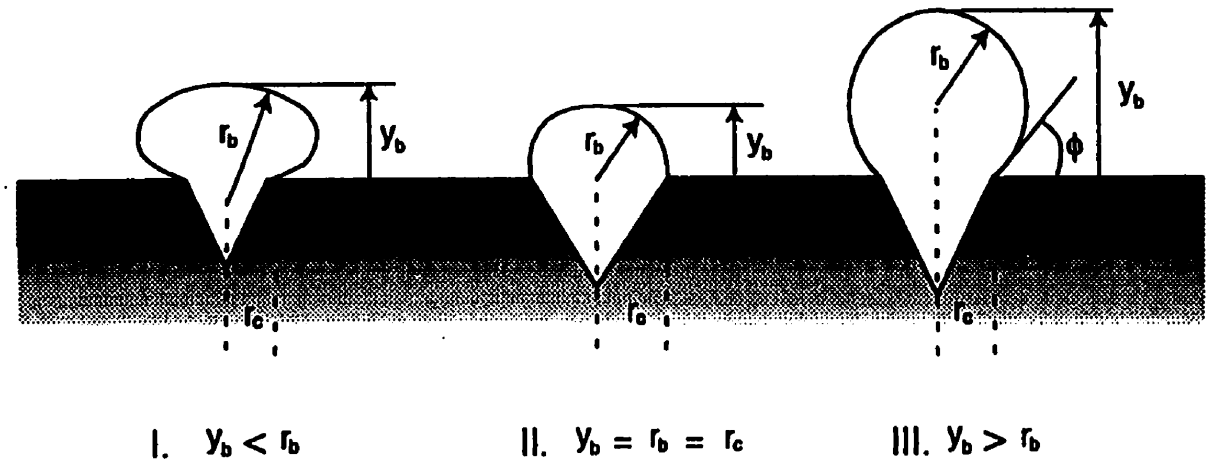


Figure 5.3 Possible bubble models (Reproduced from Reference 36)

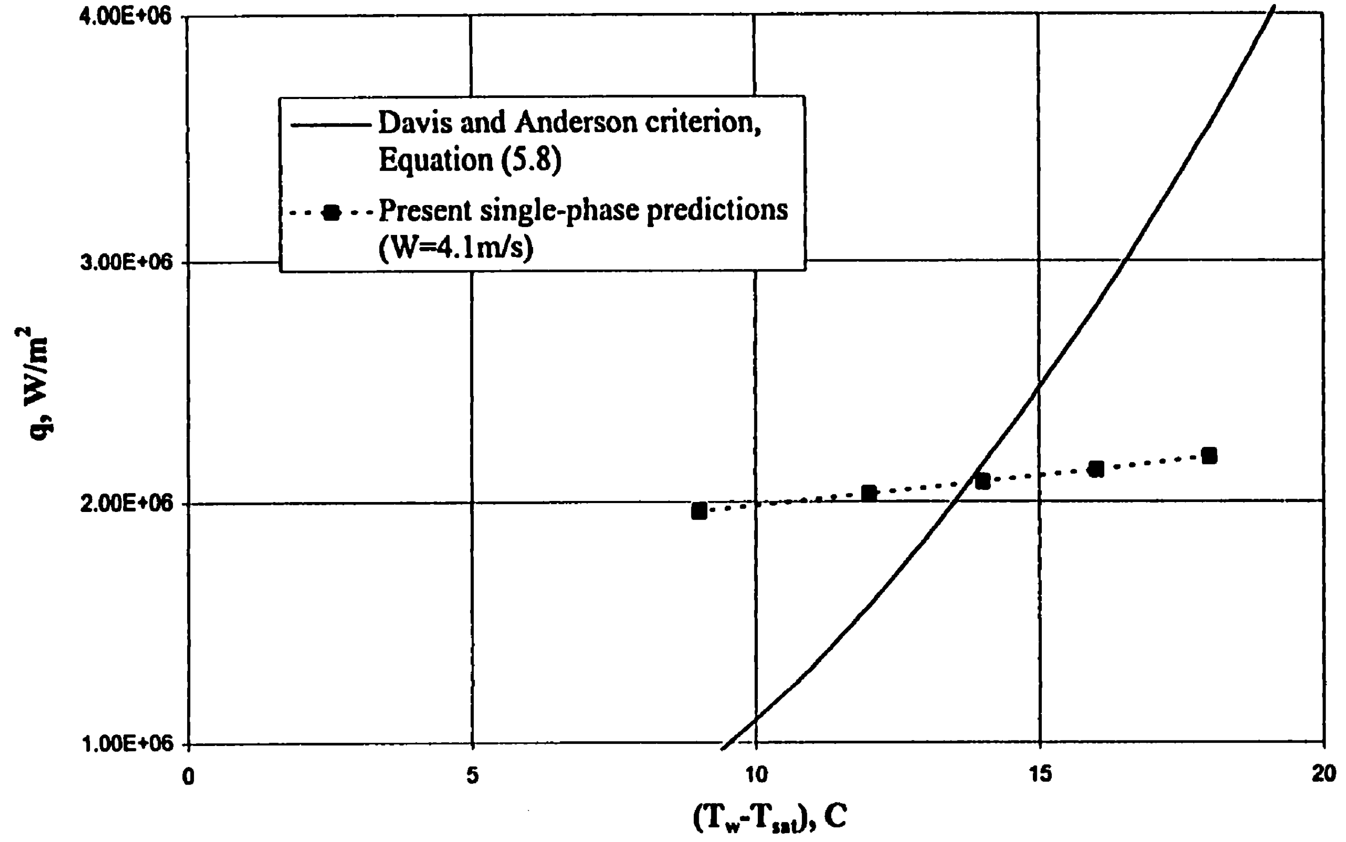


Figure 5.4 Graphical prediction of the ONB heat flux using Davis and Anderson criterion (ONB test number 20)

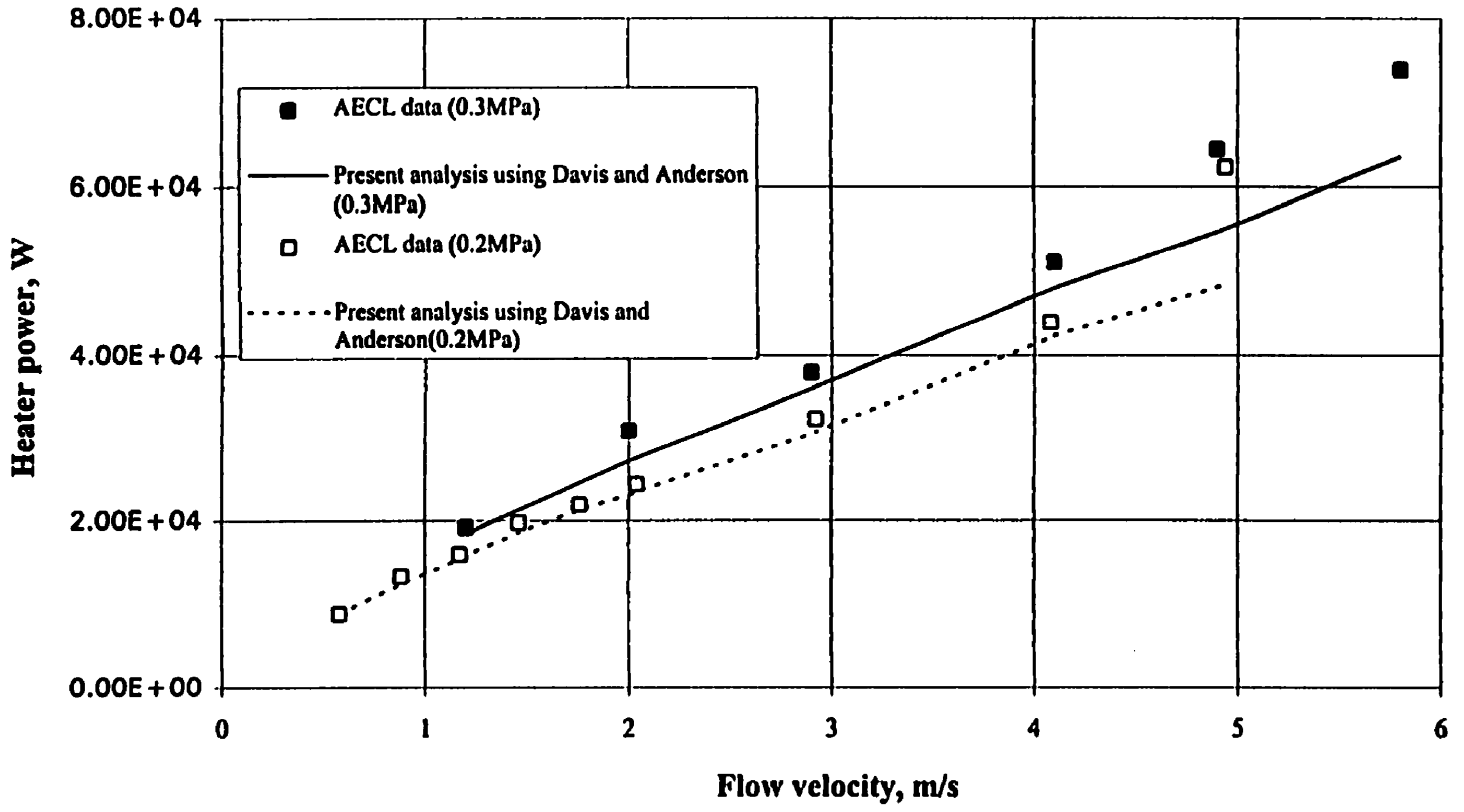


Figure 5.5 Comparison of AECL ONB data with present analysis for various flow velocities

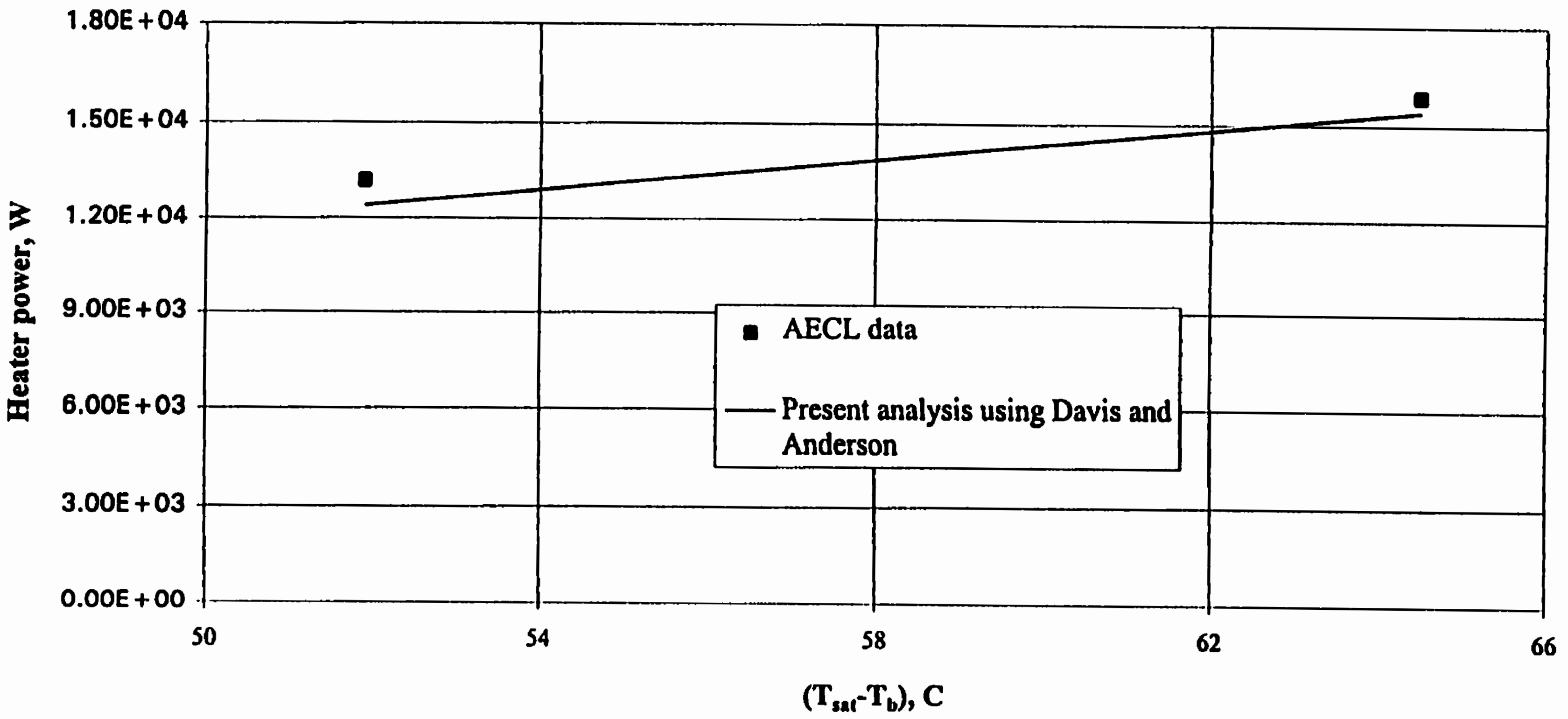


Figure 5.6 Comparison of AECL ONB data with present analysis for various subcoolings

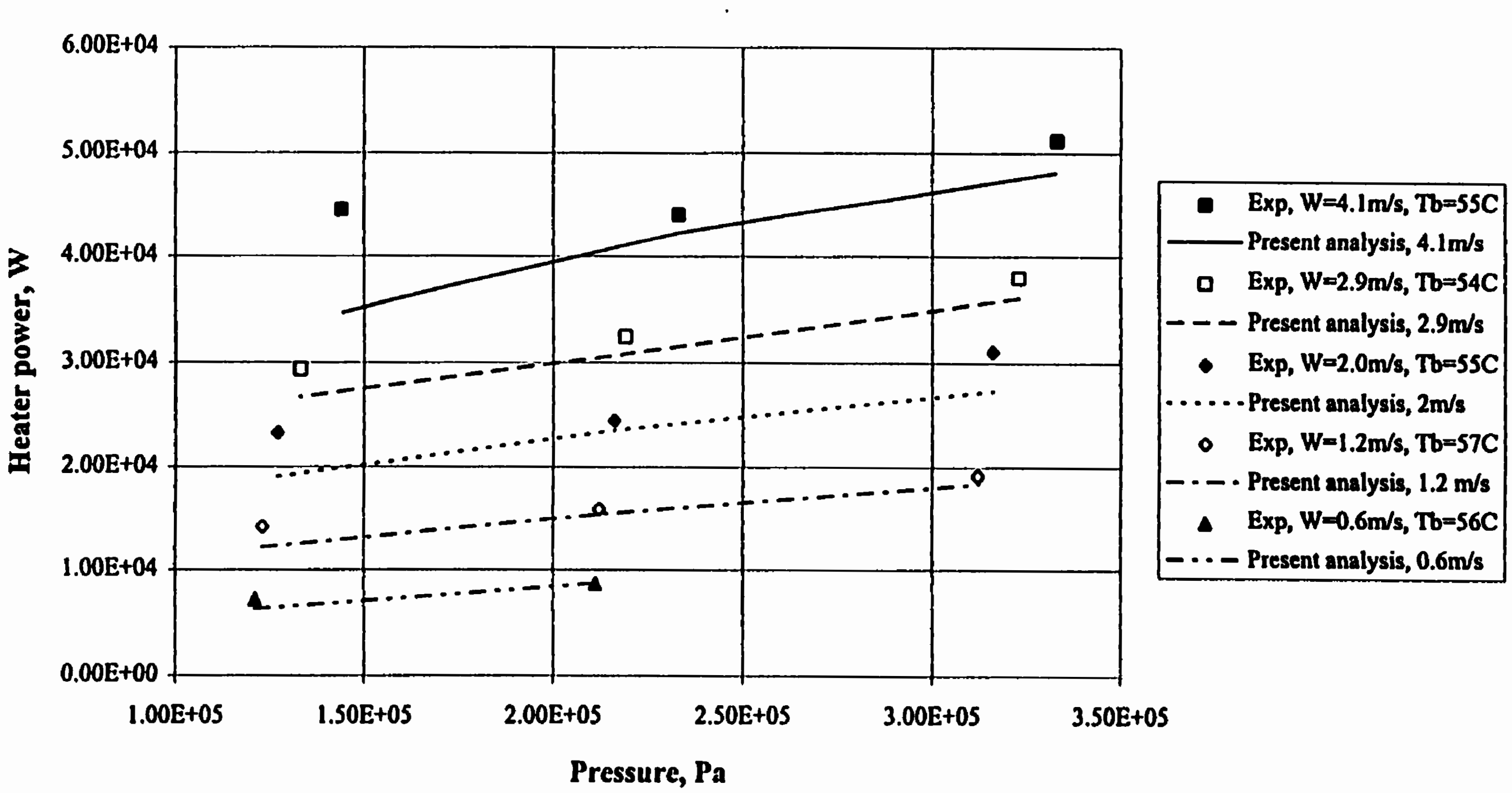


Figure 5.7 Comparison of AECL ONB data with present analysis for various pressures

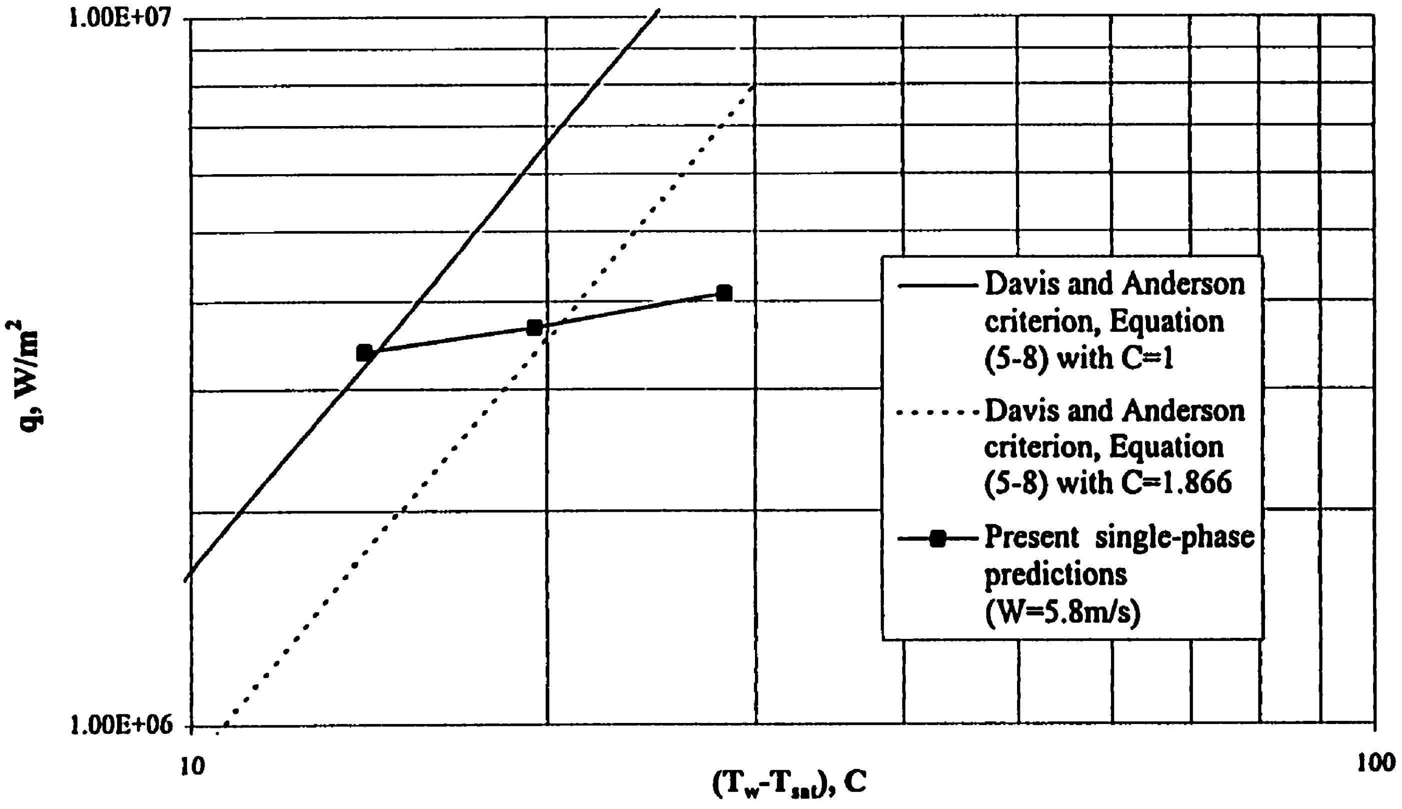


Figure 5.8 Sensitivity of C value in Davis and Anderson criterion (ONB test number 2, p=0.35 MPa, $(T_{sat}-T_b)$=84C)

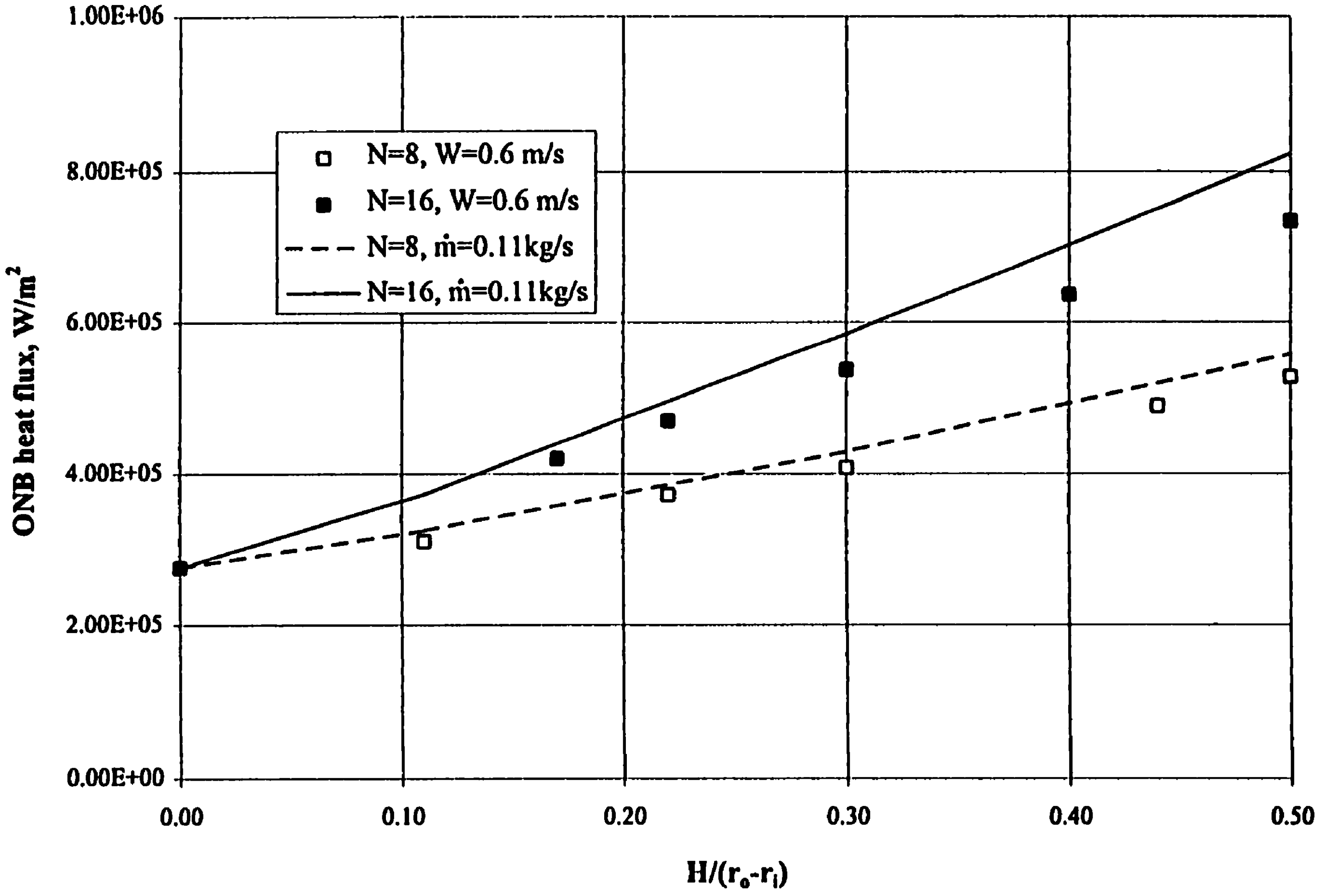


Figure 5.9 Effect of fin geometry on ONB flux for W=0.6 m/s and ṁ=0.11 kg/s

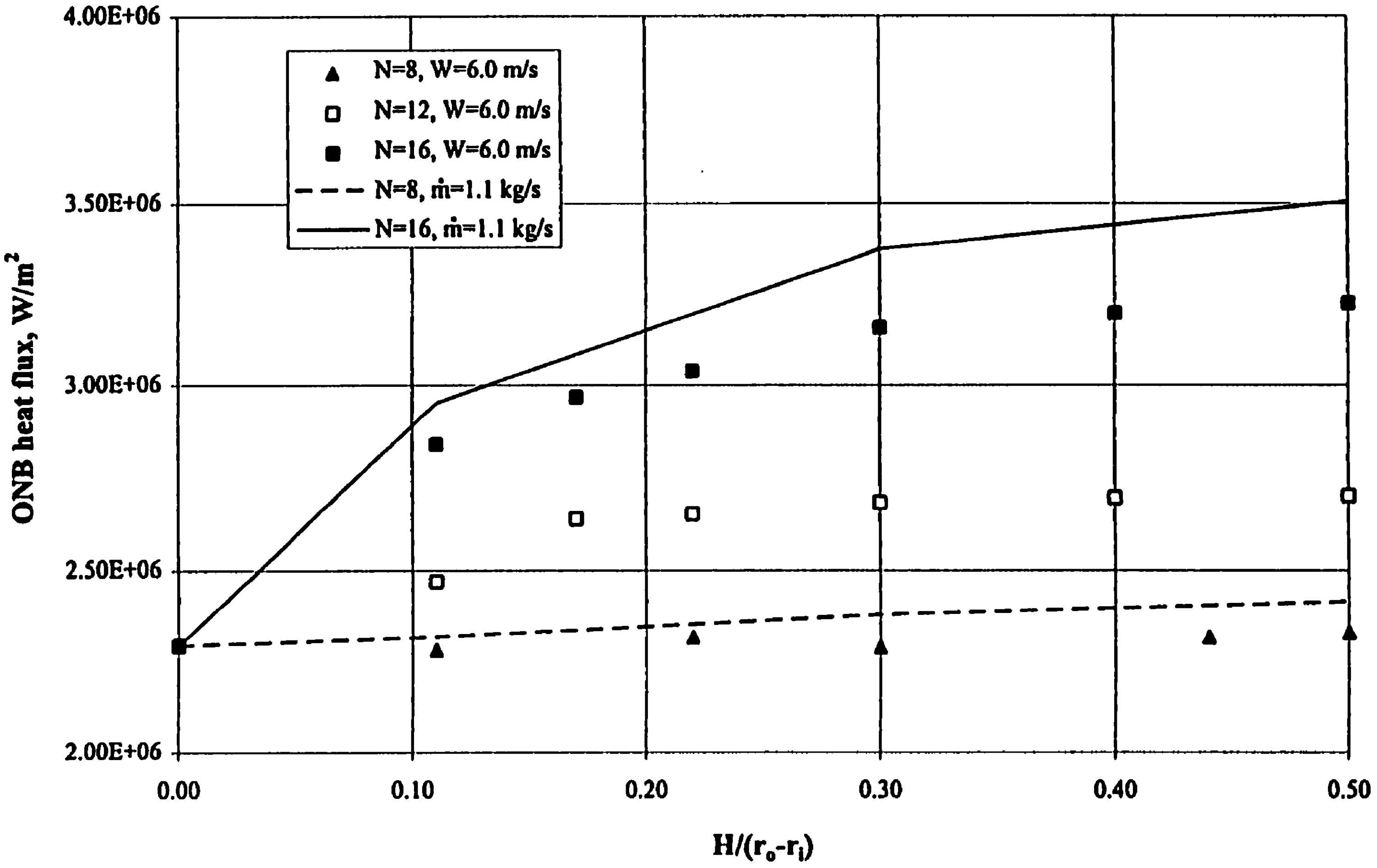


Figure 5.10 Effect of fin geometry on ONB heat flux for W=6.0 m/s and $\dot{m}$=1.1 kg/s

APPENDICES

APPENDIX A

AECL Finned Annuli Data

for Single-Phase and ONB

AECL Single-Phase Data						
Test numbe	Power, W	W, m/s	p, Pa	Tb, C	Tsh, C	Tft, C
1	4664	1.3	122800	46	70.95	67.97
2	4731	1.25	122800	48.27	73.78	70.7
3	4774	1.3	122800	50.79	76.03	72.22
4	4690	1.25	122800	52.09	76.81	73.19
5	4692	1.3	122800	53.93	78.91	74.85
6	4734	1.25	122800	55.13	79.15	75.78
7	6769	1.76	124500	59.18	85.55	80.57
8	6753	1.76	124500	61.37	87.79	80.67
9	6835	1.81	124500	63.3	89.11	84.57
10	6820	1.76	124500	66.42	91.94	87.26
11	6802	1.81	124500	69.22	94.97	89.99
12	9332	2.27	127500	56.75	84.96	79.88
13	9300	2.27	127500	58.48	86.47	80.66
14	9388	2.27	127500	61.34	89.4	83.4
15	9354	2.27	127500	62.36	91.26	85.21
16	9354	2.27	127500	64.49	92.29	86.47
17	9371	2.27	127500	66.55	94.29	88.18
18	9337	2.33	127500	68.52	95.07	89.84
19	9337	2.33	127500	70.46	97.66	91.46
20	9390	2.33	127500	73.15	100.3	94.48
21	9408	2.39	127500	76.95	103.6	98.1
22	9443	2.39	127500	79.71	105.8	100.5
23	11350	2.91	130000	56.38	83.1	74.9
24	11350	2.91	130000	59.58	86.9	77.9
25	11350	2.91	130000	61.68	88.9	79.8
26	11350	2.91	130000	63.18	90.6	81.2
27	11350	2.91	130000	64.28	91.5	82.5
28	11350	2.91	130000	66.18	92.9	84.2
29	11350	2.91	130000	67.57	94.8	85.6
30	11350	2.91	130000	69.17	95.8	87.6
31	11350	2.91	130000	70.57	97.3	88.7
32	11350	2.91	130000	72.47	99	90.15
33	11350	2.91	130000	74.87	101.1	92.1
34	11350	2.91	130000	78.47	106	95.5
35	9562	1.16	211100	66.84	113	108.2
36	11250	1.16	210900	67.68	121.6	115.9
37	9562	1.16	211100	66.84	113	108.2
38	11250	1.16	210900	67.68	121.6	115.9
39	4968	1.45	213800	47.78	67.88	66.63
40	4805	1.45	213900	47.66	68.07	67.09
41	6741	1.45	213600	48.88	76.52	73.46
42	7033	1.45	213300	49.08	76.86	73.73
43	6666	1.45	213600	48.88	76.66	73.73
44	6903	1.45	213300	49	75.93	73.97
45	8614	1.45	213800	50.09	84.75	81.67
46	10400	1.45	213800	51.15	92.14	88.42

47	10230	1.45	213800	51.08	92.29	88.7
48	12600	1.45	213300	52.36	101.5	96.53
49	12170	1.46	213200	52.1	101.3	96.73
50	12460	1.46	213200	52.28	101.3	96.36
51	14140	1.46	213300	53.26	109.4	103.2
52	14280	1.46	213400	53.36	110	104.1
53	16230	1.47	213500	54.25	117.8	111.5
54	17440	1.46	213600	54.97	122.4	115.8
55	17590	1.46	214200	55.19	121.5	115.3
56	17390	1.47	213300	55.08	122.3	116
57	4922	1.74	211700	47.49	64.56	63.08
58	6592	1.74	215100	48.29	71.34	68.95
59	8603	1.75	214700	49.35	79.21	76.15
60	10290	1.75	214600	50.13	86.37	82.24
61	12250	1.76	214600	51.01	93.6	89.26
62	12500	1.75	214900	51.09	93.8	89.99
63	12360	1.75	214900	51.09	93.64	89.27
64	12170	1.75	214900	50.93	93.85	88.09
65	13970	1.75	214900	51.9	99.13	94.62
66	16110	1.75	214600	52.82	106.9	101.1
67	16320	1.75	215400	52.91	108.5	102
68	15970	1.75	214200	52.75	108.1	101.5
69	18020	1.75	214400	53.64	114.3	106.9
70	18870	1.76	214400	53.97	117.4	110.3
71	19020	1.75	214600	54.06	117.1	110
72	19500	1.76	217400	54.25	121.5	112.9
73	20030	1.76	216800	54.5	120.3	113.1
74	19820	1.76	214700	54.38	120.4	113.4
75	19640	1.76	216800	54.35	121.7	112.7
76	20720	1.76	218000	54.92	122.2	115
77	20260	1.76	211700	54.65	122.2	115.9
78	20830	1.76	212400	54.97	122.4	115.3
79	20380	1.76	216800	54.7	122.6	115.2
80	9486	2.03	316800	48.69	77.94	73.96
81	12830	2.04	317500	49.95	89.36	83.74
82	12900	2.03	316900	50.02	89.55	84.3
83	15920	2.04	316200	51.48	99.68	92.89
84	19100	2.04	316000	53.05	109.9	102.3
85	19530	2.04	316200	53.15	110	102.9
86	19190	2.04	315900	53.09	110.2	102.7
87	19440	2.05	316200	53.19	110.5	103.1
88	22890	2.05	317500	53.99	120.9	112
89	24410	2.05	315500	54.58	125	116
90	24300	2.05	314900	54.49	125.1	115.7
91	25670	2.05	318200	54.87	128.8	119
92	27430	2.05	316500	55.49	133.3	123.2
93	27230	2.05	316600	55.39	133.1	123
94	28250	2.06	317500	55.85	135.5	125.2
95	28670	2.05	316700	56.03	135.1	125.8

96	8732	2.04	127400	48.81	75.11	72.04
97	10380	2.04	127100	49.4	80.85	77.13
98	12180	2.04	127200	50.26	87.23	82.96
99	14290	2.04	127700	51.03	94.06	88.59
100	16610	2.05	127500	51.94	101.8	95.15
101	16850	2.05	129700	52.04	102.1	95.26
102	16640	2.04	127800	51.97	101.6	95.68
103	18350	2.05	129700	52.54	105.6	100.5
104	18100	2.04	128800	52.49	106.7	99.69
105	35490	5.86	354400	49.89	97.47	69.03
106	34920	5.86	354700	49.97	96.92	70.07
107	39800	5.83	354000	50.98	103.8	72.27
108	39520	5.82	354100	50.85	103.7	72.07
109	47230	5.84	354100	52.04	112.9	76.71
110	47870	5.83	354100	52.14	114.7	79
111	46810	5.84	354100	51.98	113.8	78.55
112	47660	5.84	354700	52.14	112.5	78.61
113	46980	5.84	354400	52	113.5	78.91
114	53470	5.82	353000	52.81	122.4	82.83
115	60570	5.83	352600	53.85	130.7	86.51
116	66370	5.84	353100	54.68	137.3	89.7
117	67200	5.83	353400	54.78	137.6	87.3
118	68540	5.83	352900	54.89	138.8	88.87
119	67350	5.83	353100	54.64	138.6	87.94
120	68130	5.83	353100	54.74	138.8	89.5
121	66520	5.83	353400	54.53	136.6	87.06
122	42510	5.83	251200	50.53	101.7	85.12
123	45490	5.83	251300	51.29	105.7	87.73
124	48830	5.83	251000	51.81	109.9	90.74
125	51390	5.85	251100	52.13	112.8	92.49
126	53610	5.83	250700	52.32	115.3	94.12
127	56750	5.82	250500	52.48	118.4	96.44
128	57200	5.82	250700	52.62	118.7	96.37
129	60610	5.83	250400	53.37	122.3	99.4
130	63460	5.83	250300	53.81	125.7	101.8
131	66600	5.83	249900	54.23	127.4	103.3
132	42860	5.83	166500	50.59	102.5	85.48
133	45200	5.84	166300	51.11	105	87.72
134	48180	5.83	166200	51.62	107.9	90.28
135	52090	5.84	166300	52.18	112	93.65
136	54290	5.83	166300	52.51	114.4	95.12
137	54100	5.83	166400	52.53	114.5	94.98
138	5689	1.17	312800	48.82	76.93	74.48
139	7406	1.17	312000	50.04	86.4	83.02
140	9041	1.17	311700	51.44	95.59	91.46
141	11280	1.17	312600	52.92	107.2	102.2
142	13240	1.18	312400	54.31	116.9	111
143	14650	1.18	312400	55.34	124.3	118.4
144	16480	1.18	312600	56.69	132.9	126.3

145	4710	1.15	122900	48.07	71.76	69.83
146	6552	1.15	122900	49.53	82.17	79.36
147	7755	1.16	122700	50.73	88.72	85.42
148	9060	1.16	122700	51.57	95.4	91.85
149	10260	1.16	122800	52.37	102.3	98.07
150	3180	1.16	212000	47.51	63.64	62.51
151	4913	1.16	211500	48.71	73.65	71.62
152	6552	1.16	211200	49.92	82.02	79.96
153	7798	1.16	211600	50.79	89.52	85.64
154	9191	1.16	211700	51.63	96.7	92.83
155	10680	1.17	212000	52.8	104.8	100.5
156	12110	1.16	211700	53.77	112	106.8
157	13440	1.17	211400	54.74	118.5	113.5
158	4935	1.45	213800	47.75	67.92	66.72
159	6791	1.45	213500	48.93	76.52	73.68
160	8614	1.45	213800	50.08	84.75	81.67
161	10340	1.45	213800	51.12	92.19	88.51
162	12370	1.46	213200	52.22	101.4	96.54
163	14230	1.46	213400	53.31	109.7	103.7
164	16230	1.47	213500	54.25	117.8	111.5
165	4922	1.74	211700	47.49	64.56	63.08
166	6592	1.74	215100	48.29	71.34	68.95
167	8603	1.75	214700	49.35	79.21	76.15
168	10290	1.75	214600	50.13	86.37	82.24
169	12330	1.75	214900	51.05	93.69	89.19
170	13970	1.75	214900	51.9	99.13	94.62
171	16170	1.75	214800	52.84	107.8	101.5
172	18020	1.75	214400	53.64	114.3	106.9
173	18970	1.76	214500	54.03	117.2	110.1
174	19760	1.76	216100	54.37	120.9	113.1
175	20500	1.76	214800	54.77	122.5	115.3
176	9491	2.03	316800	48.69	77.94	73.96
177	12890	2.03	317000	50.01	89.52	84.21
178	15930	2.04	316200	51.48	99.68	92.89
179	19220	2.04	316000	53.09	110.1	102.5
180	22890	2.05	317500	53.98	120.9	112
181	24350	2.05	315100	54.53	125	115.8
182	25670	2.05	318200	54.87	128.8	119
183	27260	2.05	316600	55.41	133.1	123.1
184	28550	2.05	316900	55.98	135.2	125.6
185	8732	2.04	127400	48.8	75.11	72.04
186	10370	2.04	127100	49.39	80.85	77.13
187	12180	2.04	127200	50.26	87.23	82.96
188	14300	2.04	127700	51.03	94.06	88.59
189	16650	2.04	127900	51.97	101.8	95.32
190	18140	2.05	129000	52.5	106.5	99.81
191	4781	2.04	215600	46.98	63.44	60.91
192	7370	2.04	216300	48.1	71.46	68.86
193	8752	2.04	216500	48.8	76.87	73.16

194	10370	2.04	216500	49.29	82.52	77.22
195	11760	2.05	215900	49.88	87.01	81.88
196	12830	2.05	215600	50.39	91.16	85.3
197	14290	2.04	216000	51.1	96.6	90.24
198	15890	2.04	216300	51.71	101.5	94.02
199	17570	2.05	216400	52.34	107.3	98.91
200	19200	2.05	215800	53.2	112.2	103
201	20360	2.05	215900	53.52	115.9	106.3
202	22050	2.05	216400	54.04	121.7	110.4
203	23050	2.89	321400	51.97	108.6	81.54
204	25610	2.9	322300	52.48	114.3	85.01
205	26910	2.9	322200	52.91	118.4	88.41
206	27760	2.9	322000	53.21	120	89.23
207	29280	2.91	322000	53.65	124.5	91
208	31600	2.91	321400	54.24	130.1	94.3
209	32730	2.91	322200	54.56	132.6	95.9
210	33760	2.91	321800	54.76	134.6	96.67
211	9474	2.89	218900	47.79	72.47	61.68
212	11130	2.9	219400	48.28	76.83	63.97
213	12990	2.9	219500	48.85	81.35	66.98
214	15690	2.91	220400	49.64	88.88	71.05
215	16780	2.91	220100	49.83	91.63	72.36
216	18480	2.92	221400	50.27	96.88	74.93
217	19300	2.91	221800	50.49	96.14	76.71
218	18830	2.91	221700	50.37	98.08	76.01
219	20870	2.92	221500	50.85	102.8	79.53
220	23400	2.92	221800	51.66	109.4	83.24
221	25300	2.93	222100	52.4	114.3	85.92
222	26970	2.92	221100	52.72	118.9	88.25
223	28300	2.92	221500	53.04	122.3	89.67
224	29050	2.92	220200	53.27	121.6	91.55
225	9370	2.91	133000	47.71	69.41	65.99
226	11340	2.91	133000	48.18	73.97	70.29
227	13330	2.91	132800	49.12	79.68	75.25
228	15220	2.91	132800	49.65	84.52	78.44
229	17040	2.91	132800	50.18	88.92	82.61
230	18730	2.91	132700	50.62	92.44	85.84
231	20540	2.91	132700	51.2	97.56	90.26
232	22560	2.91	132500	51.58	101.9	93.6
233	24580	2.9	132500	52.23	106.4	97.6
234	32510	4.05	333600	51.44	111.3	82.81
235	32840	4.06	332600	51.5	112.3	81.88
236	35820	4.06	332300	52.55	118.6	84.47
237	36480	4.07	332800	52.79	119.9	86.07
238	38820	4.07	333300	53.17	124	88.35
239	41790	4.08	333500	53.46	128.4	90.95
240	41290	4.07	333900	53.44	127.1	90.92
241	40250	4.08	333300	53.21	126.6	90.51
242	41970	4.08	332900	53.62	128.8	91.5

243	42580	4.08	333400	53.77	130.5	92.67
244	43890	4.07	332700	53.99	132.9	93.31
245	46730	4.07	332500	54.58	137	96.68
246	46030	4.07	332300	54.45	136.8	95.51
247	46450	4.07	332300	54.57	136.7	95.93
248	15670	4.07	234200	48.03	78.09	63.68
249	19170	4.06	233200	49.22	85.59	67.5
250	21510	4.07	233300	49.54	89.79	70.38
251	24830	4.07	233800	50.47	97.43	74.21
252	27220	4.07	233700	50.89	101.6	77.25
253	30970	4.07	234400	51.54	108.7	80.79
254	30000	4.07	231600	51.36	107.9	78.91
255	32330	4.07	232500	51.8	111.4	82.47
256	33090	4.07	231600	51.96	112.1	82.23
257	33480	4.07	233100	52.02	113.2	83.87
258	35890	4.07	231700	52.55	117.8	85.97
259	38440	4.07	234000	52.94	123.1	88.6
260	17010	4.07	144200	48.24	77.77	71.37
261	21170	4.07	144300	49.3	85.1	77.86
262	26910	4.08	144200	50.54	94.2	86.02
263	32120	4.09	144000	51.45	104.2	92.81
264	23140	4.93	344000	49.06	85.25	66.99
265	22680	4.93	343700	48.93	84.49	65.58
266	28530	4.95	344400	50.01	94.69	70.09
267	29320	4.94	344300	50.2	95.74	71.09
268	34370	4.95	345300	50.86	103	75.39
269	35270	4.95	345400	50.94	105.1	75
270	35780	4.95	345300	51	104.6	76.1
271	42260	4.96	345500	52.04	115	82.04
272	41090	4.96	345700	52.06	114	80.58
273	41930	4.96	345900	52.3	114.7	83.54
274	47180	4.95	342300	53.02	123.7	86.02
275	53870	4.96	342500	54.32	132.2	89.78
276	52770	4.96	342600	54.18	130.8	89.06
277	55870	4.96	342900	54.67	134.8	90.84
278	55600	4.96	343000	54.61	134.8	90.63
279	41880	4.96	239600	51.4	109.2	91.94
280	44700	4.93	239100	52.09	113.7	95.01
281	47300	4.93	238800	52.51	117	97.44
282	50520	4.93	239000	53.12	121.8	100.8
283	53870	4.94	238900	53.75	124.7	104.1
284	54060	4.94	238800	53.89	124.8	104.2
285	39160	4.95	154100	51.48	105.3	90.53
286	43060	4.95	154000	52.5	110.7	95.29

AECL ONB Data

Test numbe	Power, W	W, m/s	p, Pa	Tb, C	Tsh, C	Tft, C
1	15620	0.87	210400	40.07	127.6	123
2	73870	5.83	353400	55.18	143.6	92.13
3	72790	5.82	250000	55.13	130.9	107.5
4	63610	5.84	166200	53.92	120.3	103.1
5	7268	0.59	121000	55.01	114.3	111.2
6	8753	0.58	211200	57.4	127.8	124.3
7	13260	0.88	210500	57.53	127.3	122.6
8	19200	1.18	311900	58.73	140.3	133.8
9	14160	1.16	122800	55.13	114.9	112.5
10	15900	1.17	211800	56.49	126.6	121.3
11	19760	1.46	213400	55.96	127.6	121
12	21970	1.76	213100	55.44	125.9	118.6
13	31100	2.06	316000	56.96	140.1	130.3
14	23270	2.05	127400	54.21	117.3	110.8
15	24460	2.04	215800	55.2	128.5	116.5
16	38020	2.9	322500	55.82	141.3	102.8
17	32520	2.92	219200	54.11	130.5	95.91
18	29380	2.91	132500	53.38	116.2	105.9
19	51190	4.07	333000	55.48	141.5	100.4
20	43970	4.08	233200	54.34	130.9	93.64
21	44450	4.07	143700	53.95	120.1	107.7
22	64410	4.95	343700	55.45	143.2	95.13
23	62310	4.94	238800	55.07	130	110.8
24	57580	4.94	153900	54.2	120.8	107.4
25	13230	1.15	211100	69.12	130.3	124.2

APPENDIX B

Computer Input Description and

Sample Input Data

Set	Variable	Type	Description	Notes
1	FLOWTYPE	CHARACTER	Type of flow simulated	LAMINAR or TURBULENT is specified
2	MIXMODEL	CHARACTER	Mixing length model to be used	M4 was used throughout the present study. This option was provided to test other mixing length models.
3	RMOPT	CHARACTER	Model for radius of maximum velocity to be used	RMCALL that calculates the radius of maximum velocity was used throughout the present study. This option was provided to test other models of radius of maximum velocity.
4	FPROP	CHARACTER	Constant or variable fluid properties to be used	VARIA uses temperature-dependent properties while FIXED uses constant properties.
5	RI	REAL	Inner radius	
	RO	REAL	Outer radius	
	DEN	REAL	Constant fluid density	
	CP	REAL	Constant fluid specific heat	
	AK	REAL	Constant fluid thermal conductivity	
	VIS	REAL	Constant fluid dynamic viscosity	
	DPDZ	REAL	Axial pressure gradient	
	DTDZ	REAL	Axial temperature gradient	
	NGEOMTYP	INTEGER	Type of geometry used	1 for annulus and 21 for finned annulus
	A1	REAL	Coefficient used in Equation (2-52)	
6	IPRT	INTEGER	Model for turbulent Prandtl number to be used	Pr_t=0.9 was used throughout the study. This option was to test other Prt models.
	PRT0	REAL	Turbulent Prandtl number	
7	FNO	INTEGER	Number of fins	
	HFWIDTH	REAL	Half the fin width	
	FHT	REAL	Fin height	
8	THETAN	REAL	Angle of the segment to be simulated	
9		CHARACTER	Comment line	

10	NPTYPE	INTEGER	Type of problem to be solved	NPTYPE=1: Steady state plane problem.
				NTYPE=2: Unsteady plane problem
	NNODE	INTEGER	Number of nodes in the grid	
	NELEM	INTEGER	Number of elements in the grid	
	NOUT	INTEGER	Option for printing generated information to	NOUT= 0 : No generated information is printed.
				NOUT= 1 : Specified generated information is printed.
	NINTO	INTEGER	Option for location of flux values	NINTO=0: Values are printed at Gauss points which are specified in the element data
				NINTO=1: Values are printed at Gauss points of order one less than specified in the element data
	NPRINT1	INTEGER	Used to control the frequency of output for the nodal values of dependent variables with time	NPRINT1=0: No information is printed.
				NPRINT1=n: Generated information is printed every nth time step.
	NPRINT2	INTEGER	Used to control the frequency of output for flux	NPRINT2=0: No information is printed.
				NPRINT2=n: Generated information is printed every nth time step.
	NPRNT3	INTEGER	Used to control the frequency of output for average values in elements of different materials with time	NPRNT3=0: No information is printed.
				NPRNT3=n: Generated information is printed every nth time step.
	NPRNT4	INTEGER	Used to control the frequency of output for the nodal values of dependent variables with iteration for each time step	NPRNT4=0: No information is printed.
				NPRNT4=m: Generated information is printed every mth iteration at each time step.
	NPLOT	INTEGER	Plotting option	NPLOT=0: Data is not saved for subsequent plots.
				NPLOT=1: Data is saved for subsequent plots.

	NPDE	INTEGER	Specifies the number of equations to be solved	NPDE > 0
11	MAXITER	INTEGER	Specifies the maximum number of iterations	MAXITER > 0
	TOLEQ	REAL	Assigns a tolerance value for convergence for each equation and thus must be repeated on the same line NPDE times	Note: Solution is considered converged when absolute (new-old value)/new is less than tolerance specified for each equation.
	RELAX*	REAL	Relaxation factor to expedite convergence. Note the same value is applied to all equations.	New value=RELAX*Old value+(1-RELAX)*New value
12	NREC	INTEGER	Number of nodal point coordinate data records	Note that Set 13 must be repeated NREC times.
13	N1	INTEGER	Node number of first node along the line	
	N2	INTEGER	Node number of last node along the line	N2>N1 or N2=N1
	INC	INTEGER	Increment in node number	INC > 0
	X1	REAL	X-coordinate of first node	
	Y1	REAL	Y-coordinate of first node	
	X2	REAL	X-coordinate of last node	
	Y2	REAL	Y- coordinate of last node	
	GRAD	REAL	Information on the density of the nodal point	GRAD= 0 or 1: Equally spaced nodes are generated.
				GRAD= L_i/L_f: L_i=distance between first two nodes on the line, L_f= distance between last two nodes on line.
14	NREC	INTEGER	Number of material types	Note that Sets 15, 16 and 17 must be repeated for each equation and material type (i.e., NREC*NPDE times) .
15	NCHEK1	INTEGER	Equation number	
	NCHEK2	INTEGER	Material number	
16	IMAT	INTEGER	Flag parameter used to specify whether the material property is a constant, table or	IMAT=0: Property is a constant and is assigned the corresponding value given in PROP.
				IMAT=1: Property is a function. The user must enter the function at the designated subroutines in the source file. The specified value of PROP is not used in this case.

				IMAT=n: Property is a table. The user must enter n number of independent points.
17	PROP	REAL	This set is only used when the value IMAT = 0. When IMAT is not equal to 0 then the corresponding value of PROP is not used within the code.	
18	NREC	INTEGER	Number of element data records	
	N1	INTEGER	Number of first element in sequence	
	N2	INTEGER	Number of last element in sequence	N2>N1 OR N2=N1
	NELINC	INTEGER	Increment in element numbers in sequence	NELINC>0
	NODINC	INTEGER	Increment in nodal numbers from element to	NODINC>0
	NEE	INTEGER	Number of nodes in each element	NEE=4
	NINTE	INTEGER	Order of integration rule per coordinate	NINTE= 1, 2 or 3
	MATE	INTEGER	Material property for each element	
	NODE(I)	INTEGER	Node numbers in the first element in the	
19	NEQN	INTEGER	Equation number	Note that Set 19 must be repeated NEQN times. If NEQN = 0 and
	NREC	INTEGER	Number of point load data records	
20	N	INTEGER	The node number of the node at which the point load is applied	This set should be repeated for all point loads (i.e., NREC times). If NREC = 0 then the next equation is read.
	V	REAL	The value of the point load	
21	NEQN	INTEGER	Equation number	Note that Set 21 must be repeated NEQN times.
	NREC	INTEGER	Number of essential boundary condition data	
22	N1	INTEGER	Node number of first node in the sequence	This set must be repeated for all essential boundary conditions (i.e., NREC times).
	N2	INTEGER	Node number of the last node in the sequence	If N2>N1 or N2=N1
	INC	INTEGER	Increment in nodal numbers	INC>0

	V	REAL	Value of dependent variable in essential boundary condition	
23	NEQN	INTEGER	Equation number	Note that Set 23 must be repeated NEQN times. If NREC= 0 then the next equation is read.
	NREC	INTEGER	Number of natural boundary condition data records	
24	N1	INTEGER	Element number of first element in the sequence	This set should be repeated for all natural boundary conditions (i.e., NREC times).
	N2	INTEGER	Element number of last element in the sequence	N2>N1 or N2=N1
	INC	INTEGER	Increment in element numbers	INC>0
	NS	INTEGER	Side number of element along which natural boundary condition is prescribed	
25	TO	REAL	Initial time	This set is used only for time dependent problems. (i.e. when NPTYPE is not equal to 1)
	TF	REAL	Final time	
	DELTAT	REAL	Time increment to be used	
26	NEQN	INTEGER	Equation number	This set is used as an initial guess for steady-state problems (i.e., NPTYPE= 1) when iteration is required.
	NREC	INTEGER	Number of initial solution data records	
27	N1	INTEGER	Node number of first node in sequence	This set must be repeated NREC*NEQN times.
	N2	INTEGER	Node number of last node in sequence	
	INC	INTEGER	Increment in node numbers	INC>0
	UO	REAL	Initial values of dependent variables	

Sample Input Data (Test Number 277)

```
TURBULENT
M4
RMCALL
VARIA
3.935E-3,8.5E-3,998.,4200.,0.6,1.E-3,-.327E5,26.47,21,0.8
0
0.9
8.,0.38E-3,1.02E-3
22.5
4.96
0.0
Test Number 277
1 1519 1440 1 1 1 1 1 0 1 2
40 0.005 0.005
 0.5 0.5 1.0
 217
   1     2 1  .2286000000E-02  .0000000000E+0   .3175000000E-02  .0000000000E+0 1.
   2     4 1  .3175000000E-02  .0000000000E+0   .3935000000E-02  .0000000000E+0 1.
   4    13 1  .3935000000E-02  .0000000000E+0   .4035000000E-02  .0000000000E+0 0.001
  13    21 1  .4035000000E-02  .0000000000E+0   .4955000000E-02  .0000000000E+0 1000.
  21    30 1  .4955000000E-02  .0000000000E+0   .5055000000E-02  .0000000000E+0 0.001
  30    40 1  .5055000000E-02  .0000000000E+0   .8400000000E-02  .0000000000E+0 1.
  40    49 1  .8400000000E-02  .0000000000E+0   .8500000000E-02  .0000000000E+0 1000.
  50    51 1  .2284835454E-02  .7295852291E-04  .3173382576E-02  .1013312818E-03 1.
  51    53 1  .3173382576E-02  .1013312818E-03  .3932995413E-02  .1255869587E-03 1.
  53    62 1  .3932995413E-02  .1255869587E-03  .4032995413E-02  .1255869587E-03 0.001
  62    70 1  .4032995413E-02  .1255869587E-03  .4955000000E-02  .1255869587E-03 1000.
  70    79 1  .4955000000E-02  .1255869587E-03  .5055000000E-02  .1255869587E-03 0.001
  79    89 1  .5055000000E-02  .1255869587E-03  .8400000000E-02  .1255869587E-03 1.
  89    98 1  .8400000000E-02  .1255869587E-03  .8499072180E-02  .1255869587E-03 1000.
  99   100 1  .2282219205E-02  .1314210877E-03  .3169748895E-02  .1825292885E-03 1.
 100   102 1  .3169748895E-02  .1825292885E-03  .3928491938E-02  .2262213387E-03 1.
 102   111 1  .3928491938E-02  .2262213387E-03  .4028491938E-02  .2262213387E-03 0.001
 111   119 1  .4028491938E-02  .2262213387E-03  .4955000000E-02  .2262213387E-03 1000.
 119   128 1  .4955000000E-02  .2262213387E-03  .5055000000E-02  .2262213387E-03 0.001
 128   138 1  .5055000000E-02  .2262213387E-03  .8400000000E-02  .2262213387E-03 1.
 138   147 1  .8400000000E-02  .2262213387E-03  .8496989108E-02  .2262213387E-03 1000.
 148   149 1  .2279261378E-02  .1753954742E-03  .3165640802E-02  .2436048253E-03 1.
 149   151 1  .3165640802E-02  .2436048253E-03  .3923400490E-02  .3019165315E-03 1.
 151   160 1  .3923400490E-02  .3019165315E-03  .4023400490E-02  .3019165315E-03 0.001
 160   168 1  .4023400490E-02  .3019165315E-03  .4955000000E-02  .3019165315E-03 1000.
 168   177 1  .4955000000E-02  .3019165315E-03  .5055000000E-02  .3019165315E-03 0.001
 177   187 1  .5055000000E-02  .3019165315E-03  .8400000000E-02  .3019165315E-03 1.
 187   196 1  .8400000000E-02  .3019165315E-03  .8494636332E-02  .3019165315E-03 1000.
 197   198 1  .2276796937E-02  .2049187779E-03  .3162217969E-02  .2846094137E-03 1.
 198   200 1  .3162217969E-02  .2846094137E-03  .3919158333E-02  .3527363915E-03 1.
 200   209 1  .3919158333E-02  .3527363915E-03  .4019158333E-02  .3527363915E-03 0.001
 209   217 1  .4019158333E-02  .3527363915E-03  .4955000000E-02  .3527363915E-03 1000.
 217   226 1  .4955000000E-02  .3527363915E-03  .5055000000E-02  .3527363915E-03 0.001
 226   236 1  .5055000000E-02  .3527363915E-03  .8400000000E-02  .3527363915E-03 1.
 236   245 1  .8400000000E-02  .3527363915E-03  .8492677848E-02  .3527363915E-03 1000.
 246   247 1  .2275386200E-02  .2200309963E-03  .3160258612E-02  .3055986060E-03 1.
 247   249 1  .3160258612E-02  .3055986060E-03  .3916729964E-02  .3787497684E-03 1.
 249   258 1  .3916729964E-02  .3787497684E-03  .4016729964E-02  .3787497684E-03 0.001
```

```
258  266 1  .4016729964E-02  .3787497684E-03  .4955000000E-02  .3787497684E-03 1000.
266  275 1  .4955000000E-02  .3787497684E-03  .5055000000E-02  .3787497684E-03 0.001
275  285 1  .5055000000E-02  .3787497684E-03  .8400000000E-02  .3787497684E-03 1.
285  294 1  .8400000000E-02  .3787497684E-03  .8491557490E-02  .3787497684E-03 1000.
295  296 1  .2275315849E-02  .2207573062E-03  .3160160901E-02  .3066073698E-03 1.
296  298 1  .3160160901E-02  .3066073698E-03  .3916608865E-02  .3800000000E-03 1.
298  307 1  .3916608865E-02  .3800000000E-03  .4016608865E-02  .3800000000E-03 0.001
307  315 1  .4016608865E-02  .3800000000E-03  .4955000000E-02  .3800000000E-03 1000.
315  324 1  .4955000000E-02  .3800000000E-03  .5055000000E-02  .3800000000E-03 0.001
324  334 1  .5055000000E-02  .3800000000E-03  .8400000000E-02  .3800000000E-03 1.
334  343 1  .8400000000E-02  .3800000000E-03  .8491501634E-02  .3800000000E-03 1000.
344  345 1  .2275303129E-02  .2208883678E-03  .3160143235E-02  .3067893997E-03 1.
345  347 1  .3160143235E-02  .3067893997E-03  .3916586970E-02  .3802256024E-03 1.
347  356 1  .3916586970E-02  .3802256024E-03  .4016586970E-02  .3802256024E-03 0.001
356  364 1  .4016586970E-02  .3802256024E-03  .4955000000E-02  .3802256024E-03 1000.
364  373 1  .4955000000E-02  .3802256024E-03  .5055000000E-02  .3802256024E-03 0.001
373  383 1  .5055000000E-02  .3802256024E-03  .8400000000E-02  .3802256024E-03 1.
383  392 1  .8400000000E-02  .3802256024E-03  .8491491535E-02  .3802256024E-03 1000.
393  394 1  .2275132277E-02  .2226412450E-03  .3159905940E-02  .3092239514E-03 1.
394  396 1  .3159905940E-02  .3092239514E-03  .3916292873E-02  .3832429130E-03 1.
396  405 1  .3916292873E-02  .3832429130E-03  .4016292873E-02  .3832429130E-03 0.001
405  413 1  .4016292873E-02  .3832429130E-03  .4955000000E-02  .3832429130E-03 1000.
413  422 1  .4955000000E-02  .3832429130E-03  .5055000000E-02  .3832429130E-03 0.001
422  432 1  .5055000000E-02  .3832429130E-03  .8400000000E-02  .3832429130E-03 1.
432  441 1  .8400000000E-02  .3832429130E-03  .8491355891E-02  .3832429130E-03 1000.
442  443 1  .2274799543E-02  .2260155703E-03  .3159443810E-02  .3139105143E-03 1.
443  445 1  .3159443810E-02  .3139105143E-03  .3915720124E-02  .3890512988E-03 1.
445  454 1  .3915720124E-02  .3890512988E-03  .4015720124E-02  .3890512988E-03 0.001
454  462 1  .4015720124E-02  .3890512988E-03  .4955000000E-02  .3890512988E-03 1000.
462  471 1  .4955000000E-02  .3890512988E-03  .5055000000E-02  .3890512988E-03 0.001
471  481 1  .5055000000E-02  .3890512988E-03  .8400000000E-02  .3890512988E-03 1.
481  490 1  .8400000000E-02  .3890512988E-03  .8491091749E-02  .3890512988E-03 1000.
491  492 1  .2274297714E-02  .2310106221E-03  .3158746825E-02  .3208480862E-03 1.
492  494 1  .3158746825E-02  .3208480862E-03  .3914856301E-02  .3976495178E-03 1.
494  503 1  .3914856301E-02  .3976495178E-03  .4014856301E-02  .3976495178E-03 0.001
503  511 1  .4014856301E-02  .3976495178E-03  .4955000000E-02  .3976495178E-03 1000.
511  520 1  .4955000000E-02  .3976495178E-03  .5055000000E-02  .3976495178E-03 0.001
520  530 1  .5055000000E-02  .3976495178E-03  .8400000000E-02  .3976495178E-03 1.
530  539 1  .8400000000E-02  .3976495178E-03  .8490693426E-02  .3976495178E-03 1000.
540  541 1  .2273616110E-02  .2376252992E-03  .3157800153E-02  .3300351378E-03 1.
541  543 1  .3157800153E-02  .3300351378E-03  .3913683025E-02  .4090356748E-03 1.
543  552 1  .3913683025E-02  .4090356748E-03  .4013683025E-02  .4090356748E-03 0.001
552  560 1  .4013683025E-02  .4090356748E-03  .4955000000E-02  .4090356748E-03 1000.
560  569 1  .4955000000E-02  .4090356748E-03  .5055000000E-02  .4090356748E-03 0.001
569  579 1  .5055000000E-02  .4090356748E-03  .8400000000E-02  .4090356748E-03 1.
579  588 1  .8400000000E-02  .4090356748E-03  .8490152520E-02  .4090356748E-03 1000.
589  590 1  .2272740593E-02  .2458580833E-03  .3156584157E-02  .3414695601E-03 1.
590  592 1  .3156584157E-02  .3414695601E-03  .3912175955E-02  .4232071556E-03 1.
592  601 1  .3912175955E-02  .4232071556E-03  .4012175955E-02  .4232071556E-03 0.001
601  609 1  .4012175955E-02  .4232071556E-03  .4955000000E-02  .4232071556E-03 1000.
609  618 1  .4955000000E-02  .4232071556E-03  .5055000000E-02  .4232071556E-03 0.001
618  628 1  .5055000000E-02  .4232071556E-03  .8400000000E-02  .4232071556E-03 1.
628  637 1  .8400000000E-02  .4232071556E-03  .8489457916E-02  .4232071556E-03 1000.
638  639 1  .2271653569E-02  .2557069877E-03  .3155074401E-02  .3551485940E-03 1.
639  641 1  .3155074401E-02  .3551485940E-03  .3910304809E-02  .4401605409E-03 1.
641  650 1  .3910304809E-02  .4401605409E-03  .4010304809E-02  .4401605409E-03 0.001
650  658 1  .4010304809E-02  .4401605409E-03  .4955000000E-02  .4401605409E-03 1000.
658  667 1  .4955000000E-02  .4401605409E-03  .5055000000E-02  .4401605409E-03 0.001
```

```
667  677 1  .5055000000E-02  .4401605409E-03  .8400000000E-02  .4401605409E-03 1.
677  686 1  .8400000000E-02  .4401605409E-03  .8488595803E-02  .4401605409E-03 1000.
687  688 1  .2270333998E-02  .2671694953E-03  .3153241663E-02  .3710687435E-03 1.
688  690 1  .3153241663E-02  .3710687435E-03  .3908033369E-02  .4598914978E-03 1.
690  699 1  .3908033369E-02  .4598914978E-03  .4008033369E-02  .4598914978E-03 0.001
699  707 1  .4008033369E-02  .4598914978E-03  .4955000000E-02  .4598914978E-03 1000.
707  716 1  .4955000000E-02  .4598914978E-03  .5055000000E-02  .4598914978E-03 0.001
716  726 1  .5055000000E-02  .4598914978E-03  .8400000000E-02  .4598914978E-03 1.
726  735 1  .8400000000E-02  .4598914978E-03  .8487549694E-02  .4598914978E-03 1000.
736  737 1  .2268757402E-02  .2802424836E-03  .3151051948E-02  .3892256716E-03 1.
737  739 1  .3151051948E-02  .3892256716E-03  .3905319500E-02  .4823946513E-03 1.
739  748 1  .3905319500E-02  .4823946513E-03  .4005319500E-02  .4823946513E-03 0.001
748  756 1  .4005319500E-02  .4823946513E-03  .4955000000E-02  .4823946513E-03 1000.
756  765 1  .4955000000E-02  .4823946513E-03  .5055000000E-02  .4823946513E-03 0.001
765  775 1  .505500000CE-02  .4823946513E-03  .8400000000E-02  .4823946513E-03 1.
775  784 1  .8400000000E-02  .4823946513E-03  .8486300454E-02  .4823946513E-03 1000.
785  786 1  .2267124164E-02  .2931621136E-03  .3148783561E-02  .4071696023E-03 1.
786  788 1  .3148783561E-02  .4071696023E-03  .3902508130E-02  .5046338220E-03 1.
788  797 1  .3902508130E-02  .5046338220E-03  .4002250130E-02  .5072305145E-03 0.001
797  805 1  .4002250130E-02  .5072305145E-03  .4949925696E-02  .5319447418E-03 1000.
805  814 1  .4949925696E-02  .5319447418E-03  .5049750942E-02  .5345512361E-03 0.001
814  824 1  .5049750942E-02  .5345512361E-03  .8390823403E-02  .6219221965E-03 1.
824  833 1  .8390823403E-02  .6219221965E-03  .8477051423E-02  .6241787952E-03 1000.
834  835 1  .2265417354E-02  .3060722302E-03  .3146412992E-02  .4251003197E-03 1.
835  837 1  .3146412992E-02  .4251003197E-03  .3899570118E-02  .5268566167E-03 1.
837  846 1  .3899570118E-02  .5268566167E-03  .3999026761E-02  .5320468605E-03 0.001
846  854 1  .3999026761E-02  .5320468605E-03  .4944355902E-02  .5814415843E-03 1000.
854  863 1  .4944355902E-02  .5814415843E-03  .5043963756E-02  .5866508563E-03 0.001
863  873 1  .5043963756E-02  .5866508563E-03  .8379329355E-02  .7612779739E-03 1.
873  882 1  .8379329355E-02  .7612779739E-03  .8465433712E-02  .7657885295E-03 1000.
883  884 1  .2263637029E-02  .3189724143E-03  .3143940318E-02  .4430172421E-03 1.
884  886 1  .3143940318E-02  .4430172421E-03  .3896505559E-02  .5490623142E-03 1.
886  895 1  .3896505559E-02  .5490623142E-03  .3995649518E-02  .5568427344E-03 0.001
895  903 1  .3995649518E-02  .5568427344E-03  .4938291176E-02  .6308802244E-03 1000.
903  912 1  .4938291176E-02  .6308802244E-03  .5037639059E-02  .6386879599E-03 0.001
912  922 1  .5037639059E-02  .6386879599E-03  .8365521032E-02  .9004234949E-03 1.
922  931 1  .8365521032E-02  .9004234949E-03  .8451450566E-02  .9071842850E-03 1000.
932  933 1  .2261783245E-02  .3318622473E-03  .3141365619E-02  .4609197879E-03 1.
933  935 1  .3141365619E-02  .4609197879E-03  .3893314554E-02  .5712501938E-03 1.
935  944 1  .3893314554E-02  .5712501938E-03  .3992118530E-02  .5816171821E-03 0.001
944  952 1  .3992118530E-02  .5816171821E-03  .4931732126E-02  .6802557132E-03 1000.
952  961 1  .4931732126E-02  .6802557132E-03  .5030777525E-02  .6906570016E-03 0.001
961  971 1  .5030777525E-02  .6906570016E-03  .8349402245E-02  .1039320329E-02 1.
971  980 1  .8349402245E-02  .1039320329E-02  .8435105894E-02  .1048326553E-02 1000.
981  982 1  .2259856064E-02  .3447413109E-03  .3138688977E-02  .4788073762E-03 1.
982  984 1  .3138688977E-02  .4788073762E-03  .3889997205E-02  .5934195355E-03 1.
984  993 1  .3889997205E-02  .5934195355E-03  .3988433932E-02  .6063692503E-03 0.001
993 1001 1  .3988433932E-02  .6063692503E-03  .4924679408E-02  .7295631081E-03 1000.
1001  1010 1  .4924679408E-02  .7295631081E-03  .5023379885E-02  .7425524433E-03 0.001
1010  1020 1  .5023379885E-02  .7425524433E-03  .8330977448E-02  .1177930115E-02 1.
1020  1029 1  .8330977448E-02  .1177930115E-02  .8416404261E-02  .1189175894E-02 1000.
1030  1031 1  .2257855547E-02  .3576091871E-03  .3135910481E-02  .4966794266E-03 1.
1031  1033 1  .3135910481E-02  .4966794266E-03  .3886553620E-02  .6155696200E-03 1.
1033  1042 1  .3886553620E-02  .6155696200E-03  .3984595868E-02  .6310979866E-03 0.001
1042  1050 1  .3984595868E-02  .6310979866E-03  .4917133728E-02  .7787974735E-03 1000.
1050  1059 1  .4917133728E-02  .7787974735E-03  .5015446927E-02  .7943687547E-03 0.001
1059  1069 1  .5015446927E-02  .7943687547E-03  .8310251729E-02  .1316214569E-02 1.
1069  1078 1  .8310251729E-02  .1316214569E-02  .8395350895E-02  .1329692953E-02 1000.
```

```
1079  1080 1  .2247720719E-02  .4165904113E-03  .3121834332E-02  .5785977935E-03 1.
1080  1082 1  .3121834332E-02  .5785977935E-03  .3869108061E-02  .7170967929E-03 1.
1082  1091 1  .3869108061E-02  .7170967929E-03  .3967433552E-02  .7353203455E-03 0.001
1091  1099 1  .3967433552E-02  .7353203455E-03  .4895062207E-02  .9072461545E-03 1000.
1099  1108 1  .4895062207E-02  .9072461545E-03  .4992934109E-02  .9253856393E-03 0.001
1108  1118 1  .4992934109E-02  .9253856393E-03  .8259341224E-02  .1530778414E-02 1.
1118  1127 1  .8259341224E-02  .1530778414E-02  .8357666714E-02  .1549001967E-02 1000.
1128  1129 1  .2236045415E-02  .4752861253E-03  .3105618632E-02  .6601196184E-03 1.
1129  1131 1  .3105618632E-02  .6601196184E-03  .3849010809E-02  .8181325035E-03 1.
1131  1140 1  .3849010809E-02  .8181325035E-03  .3946825569E-02  .8389236726E-03 0.001
1140  1148 1  .3946825569E-02  .8389236726E-03  .4869635856E-02  .1035073054E-02 1000.
1148  1157 1  .4869635856E-02  .1035073054E-02  .4966999383E-02  .1055768311E-02 0.001
1157  1167 1  .4966999383E-02  .1055768311E-02  .8216439846E-02  .1746458203E-02 1.
1167  1176 1  .8216439846E-02  .1746458203E-02  .8314254606E-02  .1767249372E-02 1000.
1177  1178 1  .2222837638E-02  .5336561018E-03  .3087274497E-02  .7411890303E-03 1.
1178  1180 1  .3087274497E-02  .7411890303E-03  .3826275637E-02  .9186075069E-03 1.
1180  1189 1  .3826275637E-02  .9186075069E-03  .3923512629E-02  .9419520433E-03 0.001
1189  1197 1  .3923512629E-02  .9419520433E-03  .4840872100E-02  .1162190567E-02 1000.
1197  1206 1  .4840872100E-02  .1162190567E-02  .4937660524E-02  .1185427412E-02 0.001
1206  1216 1  .4937660524E-02  .1185427412E-02  .8167907331E-02  .1960941057E-02 1.
1216  1225 1  .8167907331E-02  .1960941057E-02  .8265144323E-02  .1984285593E-02 1000.
1226  1227 1  .2208106439E-02  .5916603372E-03  .3066814498E-02  .8217504683E-03 1.
1227  1229 1  .3066814498E-02  .8217504683E-03  .3800918126E-02  .1018452943E-02 1.
1229  1238 1  .3800918126E-02  .1018452943E-02  .3897510709E-02  .1044334847E-02 0.001
1238  1246 1  .3897510709E-02  .1044334847E-02  .4808790651E-02  .1288511572E-02 1000.
1246  1255 1  .4808790651E-02  .1288511572E-02  .4904937639E-02  .1314274080E-02 0.001
1255  1265 1  .4904937639E-02  .1314274080E-02  .8113776941E-02  .2174079979E-02 1.
1265  1274 1  .8113776941E-02  .2174079979E-02  .8210369523E-02  .2199961884E-02 1000.
1275  1276 1  .2191861914E-02  .6492590781E-03  .3044252658E-02  .9017487195E-03 1.
1276  1278 1  .3044252658E-02  .9017487195E-03  .3772955657E-02  .1117600382E-02 1.
1278  1287 1  .3772955657E-02  .1117600382E-02  .3868837630E-02  .1146001916E-02 0.001
1287  1295 1  .3868837630E-02  .1146001916E-02  .4773413498E-02  .1413949495E-02 1000.
1295  1304 1  .4773413498E-02  .1413949495E-02  .4868853155E-02  .1442220010E-02 0.001
1304  1314 1  .4868853155E-02  .1442220010E-02  .8054085773E-02  .2385728896E-02 1.
1314  1323 1  .8054085773E-02  .2385728896E-02  .8149967746E-02  .2414130430E-02 1000.
1324  1325 1  .2174115196E-02  .7064128492E-03  .3019604439E-02  .9811289573E-03 1.
1325  1327 1  .3019604439E-02  .9811289573E-03  .3742407392E-02  .1215981873E-02 1.
1327  1336 1  .3742407392E-02  .1215981873E-02  .3837513043E-02  .1246883572E-02 0.001
1336  1344 1  .3837513043E-02  .1246883572E-02  .4734764886E-02  .1538418369E-02 1000.
1344  1353 1  .4734764886E-02  .1538418369E-02  .4829431802E-02  .1569177514E-02 0.001
1353  1363 1  .4829431802E-02  .1569177514E-02  .7988874737E-02  .2595742753E-02 1.
1363  1372 1  .7988874737E-02  .2595742753E-02  .8083980388E-02  .2626644453E-02 1000.
1373  1374 1  .2154878449E-02  .7630824803E-03  .2992886734E-02  .1059836778E-02 1.
1374  1376 1  .2992886734E-02  .1059836778E-02  .3709294267E-02  .1313529991E-02 1.
1376  1385 1  .3709294267E-02  .1313529991E-02  .3803558416E-02  .1346910677E-02 0.001
1385  1393 1  .3803558416E-02  .1346910677E-02  .4692871302E-02  .1661832887E-02 1000.
1393  1402 1  .4692871302E-02  .1661832887E-02  .4786700597E-02  .1695059583E-02 0.001
1402  1412 1  .4786700597E-02  .1695059583E-02  .7918188525E-02  .2803977618E-02 1.
1412  1421 1  .7918188525E-02  .2803977618E-02  .8012452674E-02  .2837358304E-02 1000.
1422  1423 1  .2134164855E-02  .8192291328E-03  .2964117854E-02  .1137818240E-02 1.
1423  1425 1  .2964117854E-02  .1137818240E-02  .3673638978E-02  .1410177882E-02 1.
1425  1434 1  .3673638978E-02  .1410177882E-02  .3766997021E-02  .1446014677E-02 0.001
1434  1442 1  .3766997021E-02  .1446014677E-02  .4647761459E-02  .1784108468E-02 1000.
1442  1451 1  .4647761459E-02  .1784108468E-02  .4740688828E-02  .1819779942E-02 0.001
1451  1461 1  .4740688828E-02  .1819779942E-02  .7842075582E-02  .3010290777E-02 1.
1461  1470 1  .7842075582E-02  .3010290777E-02  .7935433625E-02  .3046127572E-02 1000.
1471  1472 1  .2111988611E-02  .8748143265E-03  .2933317516E-02  .1215019898E-02 1.
1472  1474 1  .2933317516E-02  .1215019898E-02  .3635465960E-02  .1505859307E-02 1.
```

```
 1474  1483 1  .3635465960E-02  .1505859307E-02  .3727853914E-02  .1544127650E-02 0.001
 1483  1491 1  .3727853914E-02  .1544127650E-02  .4599466272E-02  .1905161310E-02 1000.
 1491  1500 1  .4599466272E-02  .1905161310E-02  .4691428026E-02  .1943253116E-02 0.001
 1500  1510 1  .4691428026E-02  .1943253116E-02  .7760588073E-02  .3214540832E-02 1.
 1510  1519 1  .7760588073E-02  .3214540832E-02  .7852976026E-02  .3252809176E-02 1000.
4
1 1
0    0   0    0    0   0   0   0
 -1.  0.  0.   0.   0.  0.  0.  0.
1 2
1    1   0    0    0   0   1   0
-1.0E-3 -1.0E-3  0.   0.   0.   0.   0.   0.
1 3
0    0    0    0   0    0  0   0
-1.   0.   0.   0.  0.   0. 0.  0.
1 4
0    0   0   0    0   0   0   0
-1.   0.  0.  0.   0.  0.  0.  0.
2 1
0    0   0   0    0   0   0    0
-152.   -152.    0.    0.    0.  0.    -6.10525223E9    0.
2 2
1    1   0   0    0   0   1   0
-0.6  -0.6 0.   0.   0.  0.  0.   0.
2 3
0    0   0   0    0   0   0   0
-220.  -220 0.  0.   0.  0.  0.   0.
2 4
 0    0   0    0    0   0   0    0
-67.2E-3  -67.2E-3   0.   0.   0.   0.   0.    0.
65
   1 1393  48   49 4 2  1    1   2  51   50
   2 1394  48   49 4 2  3    2   3  52   51
   3 1395  48   49 4 2  3    3   4  53   52
   4 244  48   49 4 2  3    4   5  54   53
 292 1396  48   49 4 2   2   298  299  348  347
   5 245  48   49 4 2  3    5   6  55   54
 293 1397  48   49 4 2   2   299  300  349  348
   6 246  48   49 4 2  3    6   7  56   55
 294 1398  48   49 4 2   2   300  301  350  349
   7 247  48   49 4 2  3    7   8  57   56
 295 1399  48   49 4 2   2   301  302  351  350
   8 248  48   49 4 2  3    8   9  58   57
 296 1400  48   49 4 2   2   302  303  352  351
   9 249  48   49 4 2  3    9  10  59   58
 297 1401  48   49 4 2   2   303  304  353  352
  10 250  48   49 4 2  3   10  11  60   59
 298 1402  48   49 4 2   2   304  305  354  353
  11 251  48   49 4 2  3   11  12  61   60
 299 1403  48   49 4 2   2   305  306  355  354
  12 252  48   49 4 2  3   12  13  62   61
 300 1404  48   49 4 2   2   306  307  356  355
  13 253  48   49 4 2  3   13  14  63   62
 301 1405  48   49 4 2   2   307  308  357  356
  14 254  48   49 4 2  3   14  15  64   63
 302 1406  48   49 4 2   2   308  309  358  357
  15 255  48   49 4 2  3   15  16  65   64
 303 1407  48   49 4 2   2   309  310  359  358
```

```
  16 256  48  49 4 2  3   16  17  66  65
 304 1408  48  49 4 2  2  310 311 360 359
  17 257  48  49 4 2  3   17  18  67  66
 305 1409  48  49 4 2  2  311 312 361 360
  18 258  48  49 4 2  3   18  19  68  67
 306 1410  48  49 4 2  2  312 313 362 361
  19 259  48  49 4 2  3   19  20  69  68
 307 1411  48  49 4 2  2  313 314 363 362
  20 260  48  49 4 2  3   20  21  70  69
 308 1412  48  49 4 2  2  314 315 364 363
  21 1413  48  49 4 2  2   21  22  71  70
  22 1414  48  49 4 2  2   22  23  72  71
  23 1415  48  49 4 2  2   23  24  73  72
  24 1416  48  49 4 2  2   24  25  74  73
  25 1417  48  49 4 2  2   25  26  75  74
  26 1418  48  49 4 2  2   26  27  76  75
  27 1419  48  49 4 2  2   27  28  77  76
  28 1420  48  49 4 2  2   28  29  78  77
  29 1421  48  49 4 2  2   29  30  79  78
  30 1422  48  49 4 2  2   30  31  80  79
  31 1423  48  49 4 2  2   31  32  81  80
  32 1424  48  49 4 2  2   32  33  82  81
  33 1425  48  49 4 2  2   33  34  83  82
  34 1426  48  49 4 2  2   34  35  84  83
  35 1427  48  49 4 2  2   35  36  85  84
  36 1428  48  49 4 2  2   36  37  86  85
  37 1429  48  49 4 2  2   37  38  87  86
  38 1430  48  49 4 2  2   38  39  88  87
  39 1431  48  49 4 2  2   39  40  89  88
  40 1432  48  49 4 2  2   40  41  90  89
  41 1433  48  49 4 2  2   41  42  91  90
  42 1434  48  49 4 2  2   42  43  92  91
  43 1435  48  49 4 2  2   43  44  93  92
  44 1436  48  49 4 2  2   44  45  94  93
  45 1437  48  49 4 2  2   45  46  95  94
  46 1438  48  49 4 2  2   46  47  96  95
  47 1439  48  49 4 2  2   47  48  97  96
  48 1440  48  49 4 2  2   48  49  98  97
0 0
1 4
  4 1474 49 0.
 298  315  1 0.
  21  315 49 0.
  49 1519 49 0.
2 1
1491 1491  1  70.2
0 0
1 1
1 1519 1 0.0
2 1
1 1519 1 20.0
END
```

APPENDIX C

Computer Program

```
C **********************************************************************
C *                                                                    *
C *                                                                    *
C *  PROGRAM FEAT (FINITE ELEMENT ANALYSIS IN TWO DIMENSIONS)          *
C *                                                                    *
C *S.Y. SHIM (NOVEMBER 1996)                                           *
C **********************************************************************
C
C
C.....FEAT IS A FINITE ELEMENT COMPUTER PROGRAM DESIGNED TO SOLVE
C   TWO-DIMENSIONAL STEADY AND UNSTEADY FIELD PROBLEMS.
C.....BOTH PLANE AND AXISYMMETRIC PROBLEMS MAY BE ANALYZED.
C.....BOTH STEADY AND UNSTEADY PROBLEMS MAY BE ANALYZED.
C.....BOTH LINEAR AND NON-LINEAR PROBLEMS MAY BE ANALYSED.
C
C.....FREE-FIELD INPUT OPTION IS EXERCISED.
C.....DOUBLE-PRECISIONING IS USED FOR ALL REAL VARIABLES.
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C
C   CALLS:  PREP, PROS, POST
C
C
   COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
   COMMON/FILENAMES/INFILE,JTITLE
   COMMON /VDIM/ L1,L2
   COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
  .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
   COMMON/CINT/XIQ(9,2,3),WQ(9,3)
   COMMON /BAND/ IB,IB2,ISYM
   COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
   COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
   COMMON/CONST1/ALPHA,BETA,THETA
   COMMON/CONST2/THETD,THETM,THETMD,DT2,ADT,BDT,OM2ADT,
  .HM2BPA,OMADT,HPBMA
C
   INCLUDE 'THVAR.H'
C
C.....THIS PROGRAM IS DIMENSIONED FOR:
C.....2350 ELEMENTS (MAXEL),
C.....2460 NODAL POINTS (MAXNOD),
C.....2460 MAXIMUM HALF-BAND-WIDTH (MAXIB) OR MAXIMUM FULL-BAND-WIDTH,
C.....285 ESSENTIAL BOUNDARY NODES (MAXEBN),
C.....240 NATURAL BOUNDARY SIDES (MAXNBS),
C.....60 POINT LOADS (MAXPTL),
C.....45 MATERIAL PROPERTIES (MAXMAT).
C
C.....THE FOLLOWING SEVEN DIMENSION STATEMENTS MUST BE CHANGED
C   TO RE-DIMENSION THE PROGRAM.
C
C
   DIMENSION NE(2350),MAT(2350),NODES(9,2350),NINT(2350)
   DIMENSION XGM(4,2350),YGM(4,2350),SX(4,2350),SY(4,2350)
   DIMENSION X(2,2460),U(10,2460),UOLD(10,2460),UITER(10,2460)
   DIMENSION PROP(10,10,45)
   DIMENSION NODBC1(10,285),VBC1(10,285),NELBC(10,240),NSIDE(10,240)
   DIMENSION VBC2(10,2,240),NPT(10,60),VPT(10,60)
```

```
      DIMENSION GK(2460,2460),GF(2460),GFBC(2460)
      DIMENSION TVAR(10,10,45,20),VAR(10,10,45,20),IMAT(10,10,45)
      DIMENSION U1(10,2460),UU1(10,2460),U1OLD(10,2460),UU1OLD(10,2460)
      DIMENSION DIFFU(10),UELEM(10,2350),TOLEQ(10),RELAX(10),DIFFMAX(10)
      DIMENSION NBC1(10),NBC2(10)
      DIMENSION WAREA(10,45),WNODES(10),WELEM(10)
      DIMENSION AMATA(10,45),WBAR(10,45)
      DIMENSION SIGMA(2350),UTER2(2460)
C
      DATA EPS/1.E-5/
      CHARACTER*20 INFILE
      CHARACTER*4 LABEL(20)
      LOGICAL FIRST,START2
C
C.....THE FOLLOWING SEVEN PARAMETERS MUST BE CHANGED TO SET THE
C     NEW ARRAY SIZES IN RE-DIMENSIONING THE PROGRAM.
C
      MAXEL  = 2350
      MAXNOD = 2460
      MAXEBN = 285
      MAXNBS = 240
      MAXPTL = 60
      MAXMAT = 45
      MAXIB  = 2460
C
C.....MAXEL  = MAXIMUM NUMBER OF ELEMENTS
C.....MAXNOD = MAXIMUM NUMBER OF NODES
C.....MAXEBN = MAXIMUM NUMBER OF ESSENTIAL BOUNDARY NODES
C.....MAXNBS = MAXIMUM NUMBER OF NATURAL BOUNDARY SIDES
C.....MAXPTL = MAXIMUM NUMBER OF POINT LOADS
C.....MAXMAT = MAXIMUM NUMBER OF DIFFERENT MATERIAL GROUPS
C.....MAXIB  = MAXIMUM HALF-BAND-WIDTH OR MAXIMUM FULL-BAND-WIDTH
C             FOR UNSYMMETRIC PROBLEMS
C
C
      L1 = MAXNOD
      L2 = MAXIB
C
C     SET UNIT NUMBERS FOR I/O AND DISK FILES
C
      NIN   = 51
      NOU   = 52
      NFILE = 53
      NLG   = 2
C
C.....SET FILE NAMES
C
C
      READ 111,INFILE
111   FORMAT(A20)
      CALL TITLE(INFILE,JTITLE)
C
      OPEN (UNIT=NIN,FILE=INFILE(1:JTITLE)//'.inp',STATUS='OLD')
      OPEN (UNIT=NOU,FILE=INFILE(1:JTITLE)//'.lis',STATUS='UNKNOWN')
      OPEN (UNIT=NLG,FILE=INFILE(1:JTITLE)//'.lg',STATUS='UNKNOWN')
      OPEN (UNIT=3,FILE=INFILE(1:JTITLE)//'.da2',STATUS='UNKNOWN')
C
      READ(NIN,112)FLOWTYPE
```

```
112  FORMAT(A9)
     WRITE(NLG,*)'FLOWTYPE ',FLOWTYPE
     PRINT *,'FLOWTYPE ',FLOWTYPE
C
     IF(FLOWTYPE.EQ.'TURBULENT')READ(NIN,113)MIXMODEL
     IF(FLOWTYPE.EQ.'TURBULEKE')READ(NIN,113)KEMODEL
113  FORMAT(A2)
     IF(FLOWTYPE.EQ.'TURBULEKE'.AND.KEMODEL.EQ.'LB'.OR.KEMODEL.EQ.'MY')
     .   READ(NIN,*)EWMAX
C
     READ(NIN,114)RMOPT
114  FORMAT(A6)
     WRITE(NLG,*)'RM OPTION ',RMOPT
     PRINT *,'RM OPTION ',RMOPT
C
     IF(RMOPT.EQ.'RMUSER')READ(NIN,*) RMVALUE
     IF(RMOPT.EQ.'RMUSER')WRITE(NLG,*)'RM VALUE SPECIFIED IN MM ',RMVALUE
     IF(RMOPT.EQ.'RMUSER')PRINT *,'RM VALUE SPECIFIED IN MM ',RMVALUE
C
     READ(NIN,116)FPROP
116  FORMAT(A5)
     WRITE(NLG,*)'FLUID PROP OPTION ',FPROP
     PRINT *,'FLUID PROP OPTION ',FPROP
C
     PRINT*,'RI,RO,DEN,CP,AK,TW,VIS,DPDZ,DTDZ,NGEOMTYPE,A1,TIN,DZ'
     READ(NIN,*)RI,RO,DEN,CP,AK,TW,VIS,DPDZ,DTDZ,NGEOMTYPE,A1,TIN,DZ
C
     IF(FPROP.NE.'FIXED'.AND.(NGEOMTYPE.NE.1.AND.NGEOMTYPE.NE.21.AND.
     .     NGEOMTYPE.NE.22))THEN
      WRITE(NLG,*)'VAR PROP NOT SUPPORTED FOR GEOM OPT # ',NGEOMTYPE
      PRINT *,'VAR PROP NOT SUPPORTED FOR GEOM OPT # ',NGEOMTYPE
      STOP
     ENDIF
C
     READ(NIN,*)IPRT
     IF(IPRT.EQ.0)READ(NIN,*)PRT0
C
           FNO = 0.0
           HFWIDTH = 0.0
           FHT = 0.0
      BANGL= 0.0

     IF (NGEOMTYPE.EQ.0) THEN
      PRINT*,'GEOM TYPE #0: TUBE GEOMETRY MODELLED'
     ELSEIF (NGEOMTYPE.EQ.1) THEN
      PRINT*,'GEOM TYPE #1: ANNULUS GEOMETRY MODELLED'
     ELSEIF (NGEOMTYPE.EQ.11) THEN
      PRINT*,'ENTER NO OF FINS, HALF FIN WIDTH, FIN HEIGHT'
      READ(NIN,*)FNO,HFWIDTH,FHT
      PRINT*,'GEOM TYPE #11: PATANKAR ANNULUS GEOMETRY MODELLED'
     ELSEIF (NGEOMTYPE.EQ.12) THEN
      PRINT*,'GEOM TYPE #12: PATANKAR UNFINNED ANNULUS MODELLED'
     ELSEIF (NGEOMTYPE.EQ.2) THEN
      PRINT*,'ENTER NO OF FINS, HALF ANGLE SUBTENDED BY A FIN'
      READ(NIN,*)FNO,BANGL
      PRINT*,'GEOM TYPE #2: CONCENTRIC FINNED TUBE OF SOLIMAN MODELLED'
     ELSEIF (NGEOMTYPE.EQ.3 .OR. NGEOMTYPE.EQ.21) THEN
      PRINT*,'ENTER NO OF FINS, HALF FIN WIDTH, FIN HEIGHT'
```

```
       READ(NIN,*)FNO,HFWIDTH,FHT
       PRINT*,'GEOM TYPE #3: SQUARE FINNED TUBE #1 OF EDWARDS MODELLED'
      ELSEIF (NGEOMTYPE.EQ.22) THEN
       PRINT*,'GEOM TYPE #22: FA8 GRID BUT ANNULUS'
      ELSE
            PRINT*,'SPECIFY GEOMETRY TYPE,0=TUBE,1=ANNULUS,21=FA8'
        STOP
      ENDIF
C
      PRINT*,'ENTER THETA IN DEGREES'
      READ(NIN,*)THETAN
      PRINT*,'ENTER MAX RE'
      READ(NIN,*)REMAX
      PRINT*,'ENTER DPDZ INCREMENT'
      READ(NIN,*)DPDZINC
C
      FIRST = .TRUE.
      START = .TRUE.
      START2 = .TRUE.
C
C     INITIALIZE
C
      T0 = 0.
      TF = 0.
      DELTAT = 0.
      CDOWN = 0.

      DO 10 II = 1, 10
      DO 10 I = 1, 10
      DO 10 J = 1, MAXMAT
      PROP(II,I,J) = 0.0
      IMAT(II,I,J) = 0
      DO 10 K = 1, 20
      VAR(II,I,J,K) = 0.0
      TVAR(II,I,J,K) = 0.0
   10 CONTINUE
C
    1 CALL PREP (NE,MAT,NODES,NINT,X,PROP,NODBC1,
     .VBC1,NELBC,NSIDE,VBC2,NPT,VPT,U,UOLD,L2,LABEL,IMAT,
     .VAR,TVAR,TOLEQ,MAXITER,NBC1,NBC2,INFILE,JTITLE,
     .RELAX,UELEM,*89)
C
C
C     CHECK INPUT TIME PARAMETERS AND SET SOME CONSTANTS
C
      IF(NPTYPE.EQ.1)  GO TO 80
      IF(DELTAT.LE.0) GO TO 90
      IF(TF.LT.T0) GO TO 200
      IF((TF-T0).LT.DELTAT) GO TO 400
C
      IF(NPTYPE.EQ.3)  GO TO 20
C
      THETD = THETA*DELTAT
      THETM = 1. - THETA
      THETMD = THETM*DELTAT
      GO TO 80
C
   20 CONTINUE
```

```
    DO  25  I = 1, NNODE
    GFBC(I) = 0.0
    DO  24 II = 1, NPDE
    UOLD(II,I) = U(II,I) - UOLD(II,I)*DELTAT
  24 CONTINUE
  25 CONTINUE
C
    DT2 = DELTAT*DELTAT
    ADT = ALPHA*DELTAT
    BDT = BETA*DT2
    OM2ADT = (1. - 2.*ALPHA)*DELTAT
    HM2BPA = (.5 - 2.*BETA + ALPHA)*DT2
    OMADT = (1. - ALPHA)*DELTAT
    HPBMA = (.5 + BETA - ALPHA)*DT2
C
C.....TIME INTEGRATION LOOP STARTS
C
  80 TIME = T0 + DELTAT
    IIP=1
    IIIP=1
    IIIIP=1
    IF(NPTYPE.NE.1) NSTEPT=((TF-T0)/DELTAT)+EPS
    IF(NPTYPE.EQ.1) NSTEPT=1
    NSTEP=1

85   IIIIIP = 0
    DO 12 II = 1,NPDE
    DO 12  I = 1,NNODE
    UITER(II,I) = 0.0
12   CONTINUE

    IF(IIP.EQ.NPRNT1) IIP=0
    IF(IIIP.EQ.NPRNT2) IIIP=0
    IF(IIIIP.EQ.NPRNT3) IIIIP=0
    IF(NSTEP.EQ.NSTEPT) IIP=0
    IF(NSTEP.EQ.NSTEPT) IIIP=0
    IF(NSTEP.EQ.NSTEPT) IIIIP=0
C
    ITER = 0
13   ITER = ITER + 1
    IF (NPRNT4 .GT. 0) IIIIIP = IIIIIP + 1
    IF (NPRNT4 .EQ. IIIIIP) IIIIIP = 0
    IF (ITER .GT. MAXITER) THEN
     WRITE(NOU,600)ITER
     GOTO 26
    ENDIF
C
    NPDE1=1
    NPDECNTL=NPDE
    IF(FLOWTYPE.EQ.'TURBULEKE'.AND.ITER.GT.35)NPDE1=3
    IF(FLOWTYPE.EQ.'TURBULEKE'.AND.ITER.GT.35)NPDECNTL=4
C     IF(ITER.LT.20)NPDE1=1
C     IF(ITER.LT.20)NPDECNTL=2
C     ITERCNTL=ITER/20
C     IF(ITER.GE.20.AND.ITER.NE.(20*ITERCNTL))NPDE1=3
C     IF(ITER.GE.20.AND.ITER.NE.(20*ITERCNTL))NPDECNTL=4
C     IF(ITER.GT.1 .AND. DIFFU(1).LT.TOLEQ(1) .AND. DIFFU(2).LT.
C    . TOLEQ(2))THEN
```

```
C     NPDE1=3
C     NPDECNTL=4
C    ENDIF
   DO 21 IEQ=NPDE1,NPDECNTL
C
   IF (ITER .EQ. 1) THEN
   IF(NOUT .EQ. 1 .AND. NPTYPE.NE.1 .AND. (IIP.EQ.0 .OR. IIIP.EQ.0
  . .OR. IIIIP.EQ.0)) WRITE(NOU,50) IEQ,NSTEP,TIME
 50 FORMAT(///,' GENERATED SOLUTION FOR EQUATION = ',I2,/,
  .1X,'TIME STEP = ',I6,5X,'TIME OF SOLUTION = ',1PE12.4)
   ENDIF
C....
   CALL PROS (NODBC1,VBC1,NELBC,NSIDE,VBC2,NPT,VPT,NE,MAT,NODES,NINT
  ..GK,GF,GFBC,X,U,L1,L2,PROP,TIME,IIP,NSTEP,UOLD,IMAT,
  .VAR,TVAR,ITER,U1,UU1,U1OLD,UU1OLD,IEQ,UELEM,NBC1,NBC2,AMUST,SIGMA)
C....
   IF(NOUT .EQ. 1 .AND. NPRNT4 .GT. 0 .AND. IIIIIP .EQ. 0 .AND.
  .  (IIP .EQ. 0 .OR. IIIP .EQ. 0 .OR. IIIIP .EQ. 0)) THEN
         WRITE(NOU,91)IEQ,ITER,(I,U(IEQ,I),I=1,NNODE)
C        WRITE(NOU,91)IEQ,ITER,(I,AMUST(I),I=1,NELEM)
   ENDIF
 91 FORMAT(1H /,1X,'SOLUTION VECTOR  EQ = ',I3,5X,'ITERATION = ',I5,
  .     /,1X,3('NODE',8X,'U',18X)/,3(I5,5X,1PE11.4,10X))
C
   CALL STRESS(NE,X,NODES,U,IEQ,MAT,PROP,VAR,UELEM,
  .       IMAT,TVAR,SIGMA)
C
   IF((KEMODEL.EQ.'LB'.OR.KEMODEL.EQ.'MY').AND.IEQ.EQ.4)THEN
    WRITE(NOU,160)IEQ
    WRITE(NOU,170)
    DO 35 I=1,NBC1(IEQ)
    WRITE(NOU,180)NODBC1(IEQ,I),VBC1(IEQ,I)
 35 CONTINUE
   ENDIF
160  FORMAT(1X,/,' GENERATED Ew BC  EQ = ',I3)
170  FORMAT(1X,'     NODBC1        VBC1 ')
180  FORMAT(1H ,5X,I5,7X,1PE11.3)
C
21  CONTINUE
C
   CALL CHEKCONV(U,UITER,DIFFU,RELAX,DIFFMAX)
         PRINT 999,TIME,ITER,(DIFFU(IEQ),IEQ=1,NPDE),
  .     (DIFFMAX(IEQ),IEQ=1,NPDE)
         WRITE(NOU,999)TIME,ITER,(DIFFU(IEQ),IEQ=1,NPDE),
  .     (DIFFMAX(IEQ),IEQ=1,NPDE)
         WRITE(NLG,999)TIME,ITER,(DIFFU(IEQ),IEQ=1,NPDE),
  .     (DIFFMAX(IEQ),IEQ=1,NPDE)
999      FORMAT(/,1X,'TIME = ',F10.4,2X,' ITER = ',I5,10(1X,E16.10))
C
   CALL CTBULK(IMAT,NE,MAT,NODES,X,U,TIME,WAREA,WNODES,
  .       WELEM,AMATA,WBAR,UELEM,SIGMA,PROP)
C
   DO 22 IEQ=1,NPDE
   IF (DIFFU(IEQ) .GT. TOLEQ(IEQ)) GOTO 13
22  CONTINUE
C
26  DO 23 IEQ=1,NPDE
   CALL POST (X,NE,MAT,NODES,NINT,U,PROP,IIP,IIIP,IIIIP,TIME
   ..XGM,YGM,SX,SY,LABEL,UELEM,IMAT,VAR,TVAR,IEQ)
   CALL POST2 (X,NE,MAT,NODES,NINT,U,PROP,IIP,IIIP,IIIIP,TIME
   ..XGM,YGM,SX,SY,LABEL,UELEM,IMAT,VAR,TVAR,IEQ,WBAR)
23   CONTINUE
C
   CALL MATAREA(NE,MAT,NODES,X,U,TIME,IIIIP,WAREA,WNODES.
   .  WELEM,AMATA,WBAR,AMUST,SIGMA,UTER2,PRT,YY,VISTT)
C
   TIME = TIME + DELTAT

   IIP=IIP+1
   IIIP=IIIP+1
   IIIIP=IIIIP+1
   NSTEP=NSTEP+1
C
   IF(FPROP.EQ.'FIXED')THEN
    REN = (DEN*(WAREA(1,2)/AMATA(1,2))*DH)/VIS
    IF(NGEOMTYPE.EQ.21.OR.NGEOMTYPE.EQ.22)THEN
         DHNEW=2*(RO-RI)
     REN = (DEN*(WAREA(1,2)/AMATA(1,2))*DHNEW)/VIS
     REO = (DEN*(WAREA(1,2)/AMATA(1,2))*DH)/VIS
    ENDIF
   ELSE
    REN=(DENF(TAVE)*(WAREA(1,2)/AMATA(1,2))*DH)/VISF(TAVE)
    IF(NGEOMTYPE.EQ.21.OR.NGEOMTYPE.EQ.22)THEN
         DHNEW=2*(RO-RI)
     REN=(DENF(TAVE)*(WAREA(1,2)/AMATA(1,2))*DHNEW)/VISF(TAVE)
     REO=(DENF(TAVE)*(WAREA(1,2)/AMATA(1,2))*DH)/VISF(TAVE)
    ENDIF
   ENDIF
         PRINT 11,   ITER,DPDZ,AMATA(1,2),WAREA(1,2)/AMATA(1,2),REN,REO
         WRITE(NLG,11)ITER,DPDZ,AMATA(1,2),WAREA(1,2)/AMATA(1,2),REN,REO
         WRITE(NOU,11)ITER,DPDZ,AMATA(1,2),WAREA(1,2)/AMATA(1,2),REN,REO
11       FORMAT (/,5X,'ITER',6X,'DPDZ',9X,'FLOW AREA',4X,'AVE VEL.',
 .        5X,'REYNOLDS',5X,'RE ORG',/,4X,I5,5(2X,E12.6))
C
    CALL RESULTS(ITER,REN,X,U,START2,SIGMA,UELEM)
C
C---- ITERATION FOR EVERY TIME STEP -----
   ITER = 0
   IF(NSTEP.LE.NSTEPT) GO TO 85
C
C.....TIME INTEGRATION LOOP ENDS
C
   CDOWN=CDOWN+1.
C    IF (WAVE .LT. REMAX .AND. DPDZINC .NE. 0.0) THEN
   IF (REO .LT. REMAX .AND. DPDZINC .NE. 0.0) THEN
         DPDZ = DPDZ - DPDZINC*CDOWN**2.
         GOTO 85
   ENDIF
C
   GO TO 1

  89 STOP

  90 WRITE(NOU,150)
C
 600 FORMAT(///,1X,' MAXIMUM ITERATION HAS BEEN EXCEEDED ',I5)
 150 FORMAT(///,1X,'DELTAT IS LESS THAN OR EQUAL TO ZERO')
   STOP
 200 WRITE(NOU,300)
 300 FORMAT(///,1X,' THE FINAL TIME IS LESS THAN THE INITIAL TIME')
   STOP
 400 WRITE(NOU,500)
 500 FORMAT(///,1X,' DELTAT IS GREATER THAN TF-T0 ')
   STOP
   END
C
   subroutine title(infile,jt)
C
   character*20 infile
   j=1
   do i=1,20
    if(infile(i:i).eq.' ')then
    j=i-1
    go to 100
    endif
   enddo
100  continue
   jt=j
   return
   end

C********************************************************
   SUBROUTINE PREP (NE,MAT,NODES,NINT,X,PROP,NODBC1,
   .VBC1,NELBC,NSIDE,VBC2,NPT,VPT,U,UT,L2,LABEL,IMAT,
   .VAR,TVAR,TOLEQ,MAXITER,NBC1,NBC2,INFILE,JTITLE,
   .RELAX,UELEM,*)
C********************************************************
C
C   PREPROCESSOR ROUTINE: CALL ROUTINES TO READ AND GENERATE DATA
C
C   CALLED BY:  MAIN
C
C   CALLS    : RTHVAR, RCON, RNODE, RELEM, RMAT, RBC, RTIMES,
C          RIC, RCONS, OUTPL1,CALBAN
C
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
   COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
   COMMON /BAND/ IB,IB2,ISYM
   COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,
   .NOUT,NINTO,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
   COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
   COMMON/TIMES/T0,TF,DELTAT
   COMMON/CONST1/ALPHA,BETA,THETA
C
   INCLUDE 'THVAR.H'
C
   DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
   DIMENSION NODBC1(10,1),VBC1(10,1),NELBC(10,1),NSIDE(10,1),
   .VBC2(10,2,1),NPT(10,1),VPT(10,1)
   DIMENSION PROP(10,10,1),X(2,1)
   DIMENSION U(10,1),UT(10,1),UELEM(10,1)
   DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
```

```
      DIMENSION TOLEQ(1),RELAX(1)
      DIMENSION NBC1(1),NBC2(1)
C
      CHARACTER*4 LABEL(20),IEND
      CHARACTER*20 INFILE
C
C
      IEND = 'END'
C
      READ(NIN,100)LABEL
      IF (LABEL(1).EQ.IEND) GO TO 99
C
      WRITE(NOU,150)
      WRITE(NOU,200)LABEL
      WRITE(NOU,155)MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
C.....THERMAL HYDRAULIC VARIABLES
C
          PRINT *,'FLOW TYPE OF THE PROBLEM'
          PRINT *, FLOWTYPE
          WRITE(NOU,*)FLOWTYPE
          WRITE(NLG,*)FLOWTYPE
          PRINT *,'THERMALHYDRAULIC VARIABLES USED'
          PRINT 31
          PRINT 32,RI,RO,DEN,CP,AK
          PRINT 33
          PRINT 34,VIS,DPDZ,TW,DTDZ
          WRITE(NOU,30)
          WRITE(NOU,31)
          WRITE(NOU,32)RI,RO,DEN,CP,AK
          WRITE(NLG,30)
          WRITE(NLG,31)
          WRITE(NLG,32)RI,RO,DEN,CP,AK
          WRITE(NOU,33)
          WRITE(NOU,34)VIS,DPDZ,TW,DTDZ
          WRITE(NLG,33)
          WRITE(NLG,34)VIS,DPDZ,TW,DTDZ
       IF (NGEOMTYPE.EQ.0) THEN
           PRINT 35
       ELSEIF (NGEOMTYPE.EQ.1) THEN
           PRINT 351
       ELSEIF (NGEOMTYPE.EQ.11) THEN
           PRINT 352
           WRITE(NLG,352)
       ELSEIF (NGEOMTYPE.EQ.12) THEN
           PRINT 353
           WRITE(NLG,353)
       ELSEIF (NGEOMTYPE.EQ.2) THEN
        PRINT 36
           PRINT 37,FNO,BANGL
           WRITE(NOU,36)
           WRITE(NOU,37)FNO,BANGL
           WRITE(NLG,36)
           WRITE(NLG,37)FNO,BANGL
       ELSEIF (NGEOMTYPE.EQ.3 .OR. NGEOMTYPE.EQ.21) THEN
        PRINT 38
           PRINT 39,FNO,HFWIDTH,FHT
           WRITE(NOU,38)
           WRITE(NOU,39)FNO,HFWIDTH,FHT
          WRITE(NLG,38)
          WRITE(NLG,39)FNO,HFWIDTH,FHT
          PRINT *,'A1 = ',A1
          WRITE(NOU,*)'A1 = ',A1
          WRITE(NLG,*)'A1 = ',A1
      ELSEIF (NGEOMTYPE.EQ.22) THEN
          PRINT 377
          WRITE(NLG,377)
      ENDIF
C
     IB=0
C
     CALL RTHVAR
C
     CALL RCON(TOLEQ,MAXITER,INFILE,JTITLE,RELAX,RELAXVIST)
C
     CALL RNODE (X)
C
     CALL RMAT (PROP,IMAT,VAR,TVAR)
C
     CALL RELEM (NE,MAT,NODES,NINT,X)
C
     CALL RBC (NODBC1,VBC1,NELBC,NSIDE,VBC2,NPT,VPT,NBC1,NBC2)
C
     IF(NPTYPE.NE.1) CALL RTIMES
C
C     IF(NPTYPE.NE.1) CALL RIC (U,UT,UELEM,NE,NODES)
     CALL RIC (U,UT,UELEM,NE,NODES)
C
     IF(NPTYPE.NE.1) CALL RCONST
C
     CALL OUTPL1 (X,NE,NODES,LABEL)
C
     CALL CALBAN (NODES,NE,L2)
C
C
C
     RETURN
  99 RETURN 1
C
 100 FORMAT(20A4)
 150 FORMAT(//,1X,'FEAT',/,
   .1X,'(Finite Element Analysis in Two dimension)',//,
   .1X,'REVISION BY S.Y. SHIM',
   .1X,'(16 SEP 92)',/,
   .1X,'LAST REVISION BY P.P. REVELIS AND S.Y. SHIM',
   .1X,'(30 JUL 92)',/,
   .1X,'LAST REVISION BY H.U. AKAY, P.G. WILLHITE AND H. DIDANDEH',
   .1X,'(16 APR 87)' )
 155 FORMAT(///,5X,'THIS PROGRAM IS DIMENSIONED FOR:',/,5X,I5,2X,
   .'ELEMENTS,',/,5X,I5,2X,'NODES,',/,5X,I5,2X,'ESSENTIAL BOUNDARY
   . NODES,',/,5X,I5,2X,'NATURAL BOUNDARY SIDES,',/,5X,I5,2X,
   .'POINT LOADS,',/,5X,I5,2X,'MATERIAL PROPERTIES,',/,5X,I5,2X,
   .'MAXIMUM HALF-BAND-WIDTH OR MAXIMUM FULL-BAND-WIDTH.')
 200 FORMAT(///,1X,20A4)
  30 FORMAT(//,'THERMALHYDRAULIC VARIABLES USED')
  31 FORMAT(/,5X,'RI   ',6X,'RO   ',6X,'DEN  ',
   .      6X,'CP   ',6X,'AK   ')
   32 FORMAT(5(1PE10.3,1X))
   33 FORMAT(5X,'VIS  ',6X,'DPDZ ',6X,'TW   ',6X,'DTDZ ')
   34 FORMAT(4(1PE10.3,1X))
   35 FORMAT(/,5X,'TUBE GEOMETRY USED')
  351 FORMAT(/,5X,'ANNULUS GEOMETRY USED')
  352 FORMAT(/,5X,'PATANKAR FINNED ANNULUS USED')
  353 FORMAT(/,5X,'PATANKAR UNFINNED ANNULUS USED')
  377 FORMAT(/,5X,'FA8 UNFINNED ANNULUS USED')
   36 FORMAT(/,5X,'CONC FINNED TUBE OR FINNED ANNULUS GEOMETRY USED')
   37 FORMAT(2X,'NO OF FINS = ', F3.0,2X,'HALF ANGLE SUBTENDED BY
     . A FIN (DEG) = ',F5.1)
   38 FORMAT(/,5X,'SQR FINNED TUBE OR FINNED ANNULUS GEOMETRY USED')
   39 FORMAT(2X,'NO OF FINS = ', F3.0,2X,'HALF WIDTH = ', E10.4,
     . 2X,'FIN HEIGHT = ', E10.4)
      END
C
C
C***********************************************************
      SUBROUTINE RTHVAR
C***********************************************************
C
C     READS AND CALCULATES THERMALHYDRAULICS VARIABLES
C
C     CALLED BY:  PREP
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      INCLUDE 'THVAR.H'
C
      PI = 3.141592654
      IF (NGEOMTYPE.EQ.0 .OR. NGEOMTYPE.EQ.1) THEN
       RAD = ((RO**2.0)-(RI**2.0))/(2.0*RO)
       PWET = 2.0*PI*(RO + RI)
       AFLOW = PI*(RO**2.0-RI**2.0)
       DH = 2.0*(RO-RI)
      ELSEIF (NGEOMTYPE.EQ.11) THEN
       AFLOW = PI*(RO**2. - RI**2.)
       PWET = 2.*PI*(RO +RI) + FNO*2.*FHT
       DH = 4.*AFLOW/PWET
      ELSEIF (NGEOMTYPE.EQ.12.OR.NGEOMTYPE.EQ.22) THEN
       AFLOW = PI*(RO**2. - RI**2.)
       PWET = 2.*PI*(RO +RI)
       DH = 4.*AFLOW/PWET
      ELSEIF (NGEOMTYPE.EQ.2) THEN
       BANGLRAD = BANGL*PI/180.
       AFLOW = PI*RO**2. - FNO*BANGLRAD*(RO**2.-RI**2.)
       PWET = 2.*PI*RO - FNO*2.*BANGLRAD*(RO -RI) +
     .       FNO*2.*(RO-RI)
       DH = 4.*AFLOW/PWET
      ELSEIF (NGEOMTYPE.EQ.3) THEN
       AFLOW = PI*RO**2. - FNO*2.*HFWIDTH*FHT
       PWET = 2.*PI*RO + FNO*2.*FHT
       DH = 4.*AFLOW/PWET
       DHA = 36.84E-3
      ELSEIF (NGEOMTYPE.EQ.4) THEN
      AFLOW = PI*RO**2. - FNO*2.*HFWIDTH*FHT
      PWET = 2.*PI*RO + FNO*2.*FHT
      DH = 4.*AFLOW/PWET
      DHA = 50.66E-3
     ELSEIF (NGEOMTYPE.EQ.5) THEN
      AFLOW = PI*RO**2. - FNO*2.*HFWIDTH*FHT
      PWET = 2.*PI*RO + FNO*2.*FHT
      DH = 4.*AFLOW/PWET
      DHA = 38.30E-3
     ELSEIF (NGEOMTYPE.EQ.21) THEN
      AFLOW = PI*(RO**2. - RI**2.) - FNO*2.*HFWIDTH*FHT
      PWET = 2.*PI*(RO +RI) + FNO*2.*FHT
      DH = 4.*AFLOW/PWET
     ELSE
      PRINT 1
      WRITE(NOU,1)
      WRITE(NLG,1)
1     FORMAT(///,1X,' GEOMETRY TYPE NOT DEFINED ')
      STOP
     ENDIF
C
      PRINT 2,    RI,RO,AFLOW,PWET,DHA,DH,2*RO
      WRITE(NOU,2) RI,RO,AFLOW,PWET,DHA,DH,2*RO
      WRITE(NLG,2) RI,RO,AFLOW,PWET,DHA,DH,2*RO
2     FORMAT(1X,'RI    = ',E10.3,1X,'RO    = ',E10.3,1X,'AFLOW = ',
     .- E10.3,1X,'PWET  = ',E10.3,/,1X,'DH_ED = ',E10.3,1X,
     . 'DH    = ',E10.3,1X,'2*RO  = ',E10.3)
C
     RETURN
     END
C
C
C*************************************************
     SUBROUTINE RCON(TOLEQ,MAXITER,INFILE,JTITLE,RELAX,RELAXVIST)
C*************************************************
C
C   READS CONTROL PARAMETERS
C

C   CALLED BY:  PREP
C
C
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
     COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
     COMMON /AXIS/ IAXIS
     COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
    ..NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
     COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
C
     DIMENSION TOLEQ(1),RELAX(1)
     CHARACTER*20 INFILE
C
     READ(NIN,*)NPTYPE,NNODE,NELEM,NOUT,NINTO,NPRNT1,NPRNT2,NPRNT3,
    .        NPRNT4,NPLOT,NPDE
     READ(NIN,*)MAXITER,(TOLEQ(IEQ),IEQ=1,NPDE)
     READ(NIN,*)(RELAX(IEQ),IEQ=1,NPDE),RELAXVIST
     IF (NPLOT .EQ. 1)
```

```
   .OPEN (UNIT=NFILE,FILE=INFILE(1:JTITLE)//'.plt',STATUS='UNKNOWN')
    IF(MAXITER .LE. 0) THEN
    MAXITER = 50
    WRITE(NOU,12)
    ENDIF
    DO 11 IEQ=1,NPDE
    IF(TOLEQ(IEQ).LE.0.0 .OR. TOLEQ(IEQ).GE.1.0) THEN
    TOLEQ(IEQ)=0.05
    WRITE(NOU,13)IEQ
    ENDIF
11   CONTINUE
    IF(NNODE.GT.MAXNOD) GO TO 1000
    IF(NELEM.GT.MAXEL) GO TO 2000
    IF (NPDE .LT. 1 .OR. NPDE .GT. 10) GO TO 2650
    WRITE(NOU,200)
    WRITE(NOU,300)NPTYPE,NNODE,NELEM,NOUT,NINTO,NPRNT1,NPRNT2,NPRNT3,
   .        NPRNT4,NPLOT,NPDE
    IAXIS = 0
    IF(NPTYPE.LT.0) IAXIS = 1
    NPTYPE = IABS(NPTYPE)
    IF(NPTYPE.EQ.0) GO TO 2600
    IF(NPTYPE.GT.3) GO TO 2600
    IF(IAXIS.EQ.0)  WRITE(NOU,350)
    IF(IAXIS.EQ.1)  WRITE(NOU,360)
 350 FORMAT(//,5X,' NOTE: A PLANE PROBLEM IS SOLVED ####',//)
 360 FORMAT(//,5X,' NOTE: AN AXISYMMETRIC PROBLEM IS SOLVED ####',//)
 100 FORMAT(6I5)
 200 FORMAT(1H ,//,' CONTROL PARAMETERS ')
 300 FORMAT(1X,'NPTYPE =',I6/' NNODE  =',I6/' NELEM  =',I6/
   . ' NOUT   =',I6/' NINTO  =',I6/' NPRNT1 =',I6/' NPRNT2 =',I6/
   . ' NPRNT3 =',I6/' NPRNT4 =',I6/' NPLOT  =',I6/' NPDE   =',I6/)
    RETURN
12   FORMAT(///,3X,'THE MAXIMUM NUMBER OF ITERATIONS (MAXITER) IS',
   . ' OUTSIDE ALLOWABLE LIMITS AND HAS BEEN RESET TO 50')
13   FORMAT(///,3X,'THE TOLERANCE FOR EQUATION ',I2,' IS',
   . ' OUTSIDE ALLOWABLE LIMITS AND HAS BEEN RESET TO 0.05')
1000 WRITE(NOU,1500)NNODE
1500 FORMAT(///,5X,'THE NUMBER OF NODES,',I5,',EXCEEDS THE MAXIMUM
   . NUMBER ALLOWABLE.')
     STOP
2000 WRITE(NOU,2500)NELEM
2500 FORMAT(///,5X,'THE NUMBER OF ELEMENTS,',I5,',EXCEEDS THE
   . MAXIMUM NUMBER ALLOWABLE.')
    STOP
2600 WRITE(NOU,2700)
2700 FORMAT(///,5X,'ERROR IN PROBLEM TYPE: NPTYPE',//)
    STOP
2650 WRITE(NOU,2750)NPDE
2750 FORMAT(///,5X,'ERROR IN PROBLEM TYPE: NPDE = ',I3,//)
    STOP
    END
C
C
C*****************************
    SUBROUTINE RNODE (X)
C*****************************
C
C   READS AND GENERATES NODAL POINT COORDINATES
```

```
C
C    CALLED BY:  PREP
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL*8 SQWTFRAC(100)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE
C
      DIMENSION X(2,1)
C
      DO 15 I=1,NNODE
      X(1,I)=1.E20
   15 X(2,I)=1.E20
      READ(NIN,*)NREC
      WRITE(NOU,400)
      WRITE(NOU,700)
      DO 20 IREC=1,NREC
      READ(NIN,*)N1,N2,INC,X1,Y1,XN,YN,GRAD
      IF(N2.LT.N1) N2 = N1
      IF(INC.LE.0)  INC = 1
      IF(GRAD.LE.0.) GRAD=1.
      INUM = (N2-N1)/INC
      WRITE(NOU,600) N1,N2,INC,X1,Y1,XN,YN,GRAD
      X21=XN-X1
      Y21=YN-Y1
      X(1,N1)=X1
      X(2,N1)=Y1
      ALF=DSQRT(X21*X21+Y21*Y21)
      IF(N2.EQ.N1) ALF=1.
      IF(N2.EQ.N1) GO TO 20
      ALLS=(2.*ALF/INUM)*GRAD/(GRAD+1.)
      ALSS=ALLS/GRAD
      IF(INUM.NE.1)  DEL=(ALLS-ALSS)/(INUM-1)
      IF(INUM.EQ.1)  DEL = 0.
      SUM=0.
      IIP=-1
      DO 180 N=1,INUM
      IIP=IIP+1
      AL1=ALLS-IIP*DEL
      SUM=SUM+AL1
      IN=N1+N*INC
      X(1,IN)=X1+X21*SUM/ALF
      X(2,IN)=Y1+Y21*SUM/ALF
  180 CONTINUE
C
      IGRAD=GRAD
      IF(IGRAD.EQ.99 .OR. IGRAD.EQ.98)THEN
       CALL SQWT(INUM,SQWTFRAC,IGRAD)
       DO 190 N=1,INUM
       IN=N1+N*INC
       X(1,IN)=X1+X21*SQWTFRAC(N)
       X(2,IN)=Y1+Y21*SQWTFRAC(N)
  190  CONTINUE
      ENDIF
C
   20 CONTINUE
```

```
    RETURN
  100 FORMAT(3I5,5X,5F10.0)
  400 FORMAT(1X,' INPUT NODAL POINT DATA ')
  600 FORMAT(3I5,5(1PE11.3))
  700 FORMAT(/' N1  N2  INC    X1       Y1       XN        YN
   .     GRAD')
  800 FORMAT(I5)
    END
C
C
C*************************************
     SUBROUTINE SQWT(N,SQWTFRAC,IN)
C*************************************
C
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     REAL*8 SQWTFRAC(100)
C
C
C...SQUARE-WEIGH THE GRID ABOVE THE FIN TIP
C
     SUM=0.0
     DO 111 I=1,N
      SUM=SUM+I*I
111   CONTINUE
     SUM1=0.0
    IF (IN.EQ.99)THEN
     DO 112 I=1,N
          SUM1 = I*I/SUM + SUM1
          SQWTFRAC(I) = SUM1
112   CONTINUE
C
    ELSEIF (IN.EQ.98) THEN
     DO 113 I=1,N
      J = N-I+1
      SUM1 = SUM1 + J*J
          SQWTFRAC(I) = SUM1/SUM
113   CONTINUE
    ENDIF
C
     RETURN
     END

C
C
C*****************************************
    SUBROUTINE RMAT (PROP,IMAT,VAR,TVAR)
C*****************************************
C
C   READS MATERIAL PROPERTY DATA
C
C   CALLED BY:  PREP
C
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
    COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
    COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT
   ..NINTO,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
```

```
      COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT
      COMMON /BAND/ IB,IB2,ISYM
C
      DIMENSION PROP(10,10,1)
      DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
C
      L = NPTYPE + 7
      READ(NIN,*) NREC
      IF(NREC.GT.MAXMAT) GO TO 600
      NMAT=NREC
      IF(NREC.EQ.0) GO TO 500
      DO  4  II = 1, NPDE
      WRITE(NOU,400)II
      DO  5  J = 1, NREC
      READ(NIN,*) NCHECK1,NCHECK2
      IF (NCHECK1 .NE. II .OR. NCHECK2 .NE. J) GO TO 1000
      READ(NIN,*)  (IMAT(II,I,J),I=1,L)
      READ(NIN,*)  (PROP(II,I,J),I=1,L)
      DO 6 I=1,L
            IF (IMAT(II,I,J) .GT. 1) THEN
            READ(NIN,*)(VAR(II,I,J,K),K=1,IMAT(II,I,J)),
     .   (TVAR(II,I,J,K),K=1,IMAT(II,I,J))
            ENDIF
6     CONTINUE
      IF(NPTYPE.EQ.1 .AND. J .EQ. 1) WRITE(NOU,300)
      IF(NPTYPE.EQ.2 .AND. J .EQ. 1) WRITE(NOU,310)
      IF(NPTYPE.EQ.3 .AND. J .EQ. 1) WRITE(NOU,320)
      WRITE(NOU,340)  J,(IMAT(II,I,J),I=1,L)
      WRITE(NOU,330)  J,(PROP(II,I,J),I=1,L)
5     CONTINUE
4     CONTINUE
C
      ISYM = 1
      DO  11 II = 1, NPDE
      DO  10  I = 1, NREC
      IF (DABS(PROP(II,4,I)) .GT. 0.0)  ISYM = 2
      IF (DABS(PROP(II,5,I)) .GT. 0.0)  ISYM = 2
   10 CONTINUE
   11 CONTINUE
C
  100 FORMAT(3F10.0)
  300 FORMAT('  MAT NO',4X,'K11',8X,'K22',8X,'K12',8X,'M1',9X,'M2',
     .9X,'B',10X,'F',10X,'MU')
  310 FORMAT('  MAT NO',4X,'K11',8X,'K22',8X,'K12',8X,'M1',9X,'M2',
     .9X,'B',10X,'F',10X,'MU',9X,'RHO1')
  320 FORMAT('  MAT NO',4X,'K11',8X,'K22',8X,'K12',8X,'M1',9X,'M2',
     .9X,'B',10X,'F',10X,'MU',9X,'RHO1',7X,'RHO2')
  330 FORMAT(I5,2X,10(1X,E10.3))
  340 FORMAT(I5,2X,10(6X,I5))
  400 FORMAT(/,2X,'INPUT MATERIAL PROPERTIES FOR EQUATION = ',I3)
      RETURN
  500 WRITE(NOU,510)
  510 FORMAT(///,2X,'THE NUMBER OF MATERIALS IS EQUAL TO ZERO')
      STOP
  600 WRITE(NOU,610) NREC
  610 FORMAT(///,5X,'THE NUMBER OF MATERIALS,',I5,',EXCEEDS THE
     . MAXIMUM NUMBER ALLOWABLE.')
      STOP
```

```
 1000 WRITE(NOU,1100)
 1100 FORMAT(///,2X,'CHECK MATERIAL OR EQUATION NUM (NOT CONSISTENT)')
      STOP
      END
C
C
C***********************************************
      SUBROUTINE RELEM (NE,MAT,NODES,NINT,X)
C***********************************************
C
C     READS AND GENERATES ELEMENT DATA
C
C     CALLED BY:  PREP
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT
C
      DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
      DIMENSION X(2,1)
      DIMENSION NODE(9)
C
C.....READ ELEMENT DATA
C
      READ(NIN,*)NREC
      WRITE(NOU,600)
      WRITE(NOU,700)
      DO 20 IREC=1,NREC
      READ(NIN,*)N1,N2,IELINC,NODINC,NEE,NINTE,MATE,(NODE(I),I=1,NEE)
      IF(IELINC.LE.0) IELINC=1
      IF(NODINC.LE.0) NODINC=1
      IF(N2.LE.N1) N2=N1
      IF(N2.GT.NELEM)  GO TO 99
      WRITE(NOU,350)N1,N2,IELINC,NODINC,NEE,NINTE,MATE,(NODE(I),I=1,NEE)
      IF(NEE.EQ.3) NODE(4)=NODE(1)
      IF(NEE.EQ.3) NEE=4
      IF(NEE.NE.6)  GO TO 8
      N4 = NODE(4)
      N5 = NODE(5)
      N6 = NODE(6)
      NODE(4) = NODE(1)
      NODE(5) = N4
      NODE(6) = N5
      NODE(7) = N6
      NODE(8) = NODE(1)
      NEE = 8
    8 CONTINUE
      NINC=-1
      DO 25 N=N1,N2,IELINC
      NINC=NINC+1
      DO 10 M=1,NEE
   10 NODES(M,N)=NODE(M)+NINC*NODINC
      NE(N)=NEE
      NINT(N)=NINTE
   25 MAT(N)=MATE
   20 CONTINUE
```

```
C
C
      DO 280 N=1,NELEM
      SUMX=0.
      SUMY=0.
      NEN=NE(N)
      IF(NEN.EQ.4) GO TO 280
      DO 275 M=5,NEN
      MM=NODES(M,N)
      IF (M.EQ.9) GO TO 15
      M4=NODES(M-4,N)
      M3=NODES(M-3,N)
      IF(M.EQ.8) M3=NODES(1,N)
      IF(X(1,MM).EQ.1.E20) X(1,MM)=0.5*(X(1,M4)+X(1,M3))
      IF(X(2,MM).EQ.1.E20) X(2,MM)=0.5*(X(2,M4)+X(2,M3))
      IF(NEN.EQ.8) GO TO 275
      SUMX=SUMX+X(1,M4)
      SUMY=SUMY+X(2,M4)
      IF(M.NE.9)  GO TO 275
   15 IF(X(1,MM).EQ.1.E20)  X(1,MM)=.25*SUMX
      IF(X(2,MM).EQ.1.E20)  X(2,MM)=.25*SUMY
  275 CONTINUE
  280 CONTINUE
C
C.....PRINT NODAL POINT COORDINATES
C
      IF(NOUT .NE. 1)  GO TO 32
      WRITE(NOU,520)
      WRITE(NOU,220)
      DO 30 N=1,NNODE
      WRITE(NOU,320)N,X(1,N),X(2,N)
   30 CONTINUE
C
C.....PRINT ELEMENT DATA
C
      WRITE(NOU,500)
      WRITE(NOU,200)
      DO 31 N=1,NELEM
      NEN=NE(N)
      WRITE(NOU,800)N,NE(N),NINT(N),MAT(N),
     .  (NODES(I,N),I=1,NEN)
   31 CONTINUE
   32 CONTINUE
      RETURN
C
   99 WRITE(NOU,400)
C
  100 FORMAT(16I5)
  200 FORMAT(28H ELEM NO   NEE   NINTE   MAT,20X,12HNODE NUMBERS )
  220 FORMAT(1X,' NODE NO.      X-COORDINATE       Y-COORDINATE ')
  320 FORMAT(I7,10X,1PE14.6,10X,1PE14.6)
  350 FORMAT(1X,I5,2(2X,I5),4(1X,I5),7X,9I5)
  400 FORMAT(37H0ELEMENT NUMBER EXCEEDS MAXIMUM VALUE  )
  500 FORMAT(1H ,/,' GENERATED ELEMENT DATA ')
  520 FORMAT(/,1X,' GENERATED COORDINATES ')
  600 FORMAT(/,1X,' INPUT ELEMENT DATA')
  700 FORMAT(1X,'  N1    N2   EINC  NINC   NEE   NINTE  MAT
     .     NODES')
```

```
 800 FORMAT(1X ,I5,5X,2I5,I8,10X,9I5)
 900 FORMAT(I5)
C
      STOP
C
      END
C
C
C*****************************************************************
      SUBROUTINE RBC (NODBC1,VBC1,NELBC,NSIDE,VBC2,NPT,VPT,NBC1,NBC2)
C*****************************************************************
C
C   CALLED BY:  PREP
C
C
C   READS POINT LOAD AND BOUNDARY CONDITION DATA
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT
     ..NINTO,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
C
      DIMENSION NODBC1(10,1),VBC1(10,1),NELBC(10,1),NSIDE(10,1),
     .   VBC2(10,2,1),NPT(10,1),VPT(10,1)
      DIMENSION NBC1(1),NBC2(1)
C
C   READ POINT LOADS
C
      DO 15 II=1,NPDE
      READ(NIN,*) NCHECK,NREC
      IF (NCHECK .EQ. 0 .AND. NREC .EQ. 0) GOTO 20
      IF(NREC.GT.MAXPTL) GO TO 500
      NPOINT=NREC
      IF(NPOINT.EQ.0) GO TO 15
      WRITE(NOU,100)
      DO 10 I = 1,NREC
      IF (NCHECK .NE. II) GOTO 3000
      READ(NIN,*)N,V
      WRITE(NOU,120) N,V
      NPT(II,I)=N
  10 VPT(II,I)=V
  15 CONTINUE
C
C   READ ESSENTIAL BOUNDARY CONDITION DATA
C
  20 CONTINUE
      DO 25 II=1,NPDE
      READ(NIN,*) NCHECK,NREC
      IF(NREC .EQ. 0 .AND. NCHECK .EQ. 0) GO TO 40
      NBC1(II)=0
      IF(NREC.EQ.0) GO TO 25
      WRITE(NOU,130)II
      WRITE(NOU,140)
      DO 30 J= 1,NREC
      READ(NIN,*)N1,N2,INC,V
      IF(INC.LE.0) INC = 1
      IF(N2.LT.N1)  N2 = N1
      NUM = (N2-N1)/INC + 1
      WRITE(NOU,150)N1,N2,INC,V
      DO 30 I = 1,NUM
      NBC1(II)=NBC1(II)+1
      N=N1+(I-1)*INC
      NODBC1(II,NBC1(II))=N
      VBC1(II,NBC1(II))=V
   30 CONTINUE
      IF(NBC1(II).GT.MAXEBN) GO TO 1000
C
      IF(NOUT.NE.1)  GO TO 36
      WRITE(NOU,160)II
      WRITE(NOU,170)
      DO 35 I=1,NBC1(II)
      WRITE(NOU,180)NODBC1(II,I),VBC1(II,I)
   35 CONTINUE
   36 CONTINUE
   25 CONTINUE
C
C   READ NATURAL BOUNDARY CONDITION DATA
C
   40 CONTINUE
      DO 45 II=1,NPDE
      READ(NIN,*) NCHECK,NREC
      IF(NREC .EQ. 0 .AND. NCHECK .EQ. 0) GO TO 60
      NBC2(II)=0
      IF(NREC.EQ.0) GO TO 45
      WRITE(NOU,190)II
      WRITE(NOU,200)
      DO 50 J = 1,NREC
      READ(NIN,*)N1,N2,INC,NS,P,V
      IF(INC.LE.0)  INC = 1
      IF(N2.LE.N1)  N2 = N1
      NUM = (N2-N1)/INC + 1
      WRITE(NOU,210)N1,N2,INC,NS,P,V
      DO 50 I = 1,NUM
      NBC2(II)=NBC2(II)+1
      N=N1+(I-1)*INC
      NELBC(II,NBC2(II))=N
      NSIDE(II,NBC2(II))=NS
      VBC2(II,1,NBC2(II))= P
   50 VBC2(II,2,NBC2(II))= V
      IF(NBC2(II).GT.MAXNBS) GO TO 2000
      IF(NOUT.NE.1)  GO TO 72
      WRITE(NOU,220)II
      WRITE(NOU,270)
      DO 70 I=1,NBC2(II)
      WRITE(NOU,230)NELBC(II,I),NSIDE(II,I),VBC2(II,1,I),VBC2(II,2,I)
   70 CONTINUE
   72 CONTINUE
   45 CONTINUE
   60 CONTINUE
  100 FORMAT(/,1X,'INPUT POINT LOAD DATA:'  /
     .     '  NPT      VPT '  /)
  110 FORMAT(I5,5X,F10.0)
  120 FORMAT(1X,I5,1PE11.3)
  130 FORMAT(/,1X,' INPUT ESSENTIAL BOUNDARY CONDITION DATA  EQ = ',I3)
```

```
 140 FORMAT(1X,' N1    N2    INC      V ')
 150 FORMAT(I5,2X,I5,2X,I5,3X,1PE11.3)
 160 FORMAT(1X,/,' GENERATED ESSENTIAL BOUNDARY CONDITION  EQ = ',I3)
 170 FORMAT(1X,'      NODBC1         VBC1 ')
 180 FORMAT(1H ,5X,I5,7X,1PE11.3)
 190 FORMAT(/,1X,' INPUT NATURAL BOUNDARY CONDITION DATA  EQ = ',I3)
 200 FORMAT(1X,' N1   N2   INC   NS        P         GAMA')
 210 FORMAT(4I5,5X,2(1PE11.3))
 220 FORMAT(/,1X,' GENERATED NATURAL BOUNDARY CONDITION   EQ = ',I3)
 230 FORMAT(1H ,I5,3X,I5,5X,2(1PE11.3))
 240 FORMAT(I5)
 250 FORMAT(4I5,2F10.0)
 260 FORMAT(3I5,5X,F10.0)
 270 FORMAT(1X,' NELBC    NSIDE      P         GAMA ')
     RETURN
 500 WRITE(NOU,550) NREC
 550 FORMAT(///,5X,'THE NUMBER OF POINT LOADS.',I5,'.EXCEEDS THE
    . MAXIMUM NUMBER ALLOWABLE.')
1000 WRITE(NOU,1500)NBC1(II)
1500 FORMAT(///,5X,'THE NUMBER OF ESSENTIAL BOUNDARY NODES.',I5,
    .',EXCEEDS THE MAXIMUM NUMBER ALLOWABLE.')
     STOP
2000 WRITE(NOU,2500)NBC2(II)
2500 FORMAT(///,5X,'THE NUMBER OF NATURAL BOUNDARY SIDES.',I5,
    .',EXCEEDS THE MAXIMUM NUMBER ALLOWABLE.')
     STOP
3000 WRITE(NOU,3100)
3100 FORMAT(//,2X,'CHECK POINT LOAD OR EQUATION NUM (NOT CONSISTENT)')
     STOP
     END
C
C
C**************************
     SUBROUTINE RTIMES
C**************************
C
C    READS INITIAL AND FINAL TIMES, AND THE TIME INCREMENT
C
C    CALLED BY:  PREP
C
C
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
     COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
     COMMON/TIMES/T0,TF,DELTAT
C
     READ(NIN,*)T0,TF,DELTAT
     IF(TF.NE.T0) WRITE(NOU,10)T0,TF,DELTAT
  10 FORMAT(/,1X,'THE INITIAL TIME   =',1PE12.4,/,1X,'THE FINAL',
    .' TIME     =',1PE12.4,/,1X,'THE TIME INCREMENT =',1PE12.4)
     RETURN
     END
C
C
C****************************************
     SUBROUTINE RIC (U,UT,UELEM,NE,NODES)
C****************************************
C
```

```
C    READS AND GENERATES INITIAL CONDITIONS
C
C    CALLED BY:  PREP
C
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
    COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
    COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT
   ..NINTO,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
C
    DIMENSION NE(1),NODES(9,1)
    DIMENSION U(10,1),UT(10,1),UELEM(10,1)
C
    DO 11 II=1,NPDE
    READ(NIN,*) NCHECK,NREC
    WRITE(NOU,20)II
    WRITE(NOU,30)
    DO 10 I=1,NREC
    IF (NCHECK .NE. II) GOTO 210
    READ(NIN,*)N1,N2,INC,U0
    IF(INC.LE.0) INC=1
    IF(N2.LT.N1) N2=N1
    NUM=(N2-N1)/INC + 1
    WRITE(NOU,40)N1,N2,INC,U0
    DO 10 J=1,NUM
    N=N1+(J-1)*INC
    U(II,N)=U0
  10 CONTINUE
    DO 16 NEL=1,NELEM
    NN=NE(NEL)
    UELEM(II,NEL) = 0.0
    DO 17 I = 1,NN
    UELEM(II,NEL) = UELEM(II,NEL) + U(II,NODES(I,NEL))
17   CONTINUE
    UELEM(II,NEL) = UELEM(II,NEL)/DBLE(NN)
16   CONTINUE
  11 CONTINUE
C
          IF (NPTYPE .EQ. 1) RETURN
C
    IF(NOUT.NE.1)  GO TO 12
    DO 13 II=1,NPDE
    WRITE(NOU,50)II
    WRITE(NOU,60)
    WRITE(NOU,70)(I,U(II,I),I=1,NNODE)
  13 CONTINUE
  12 CONTINUE
    IF(NPTYPE.EQ.2) RETURN
    DO 14 II=1,NPDE
    READ(NIN,*) NCHECK,NREC
    WRITE(NOU,80)II
    WRITE(NOU,90)
    DO 200 I=1,NREC
    IF (NCHECK .NE. II) GOTO 210
    READ(NIN,*) N1,N2,INC,UT0
    IF(INC.LE.0) INC=1
    IF(N2.LT.N1) N2=N1
```

```
      NUM=(N2-N1)/INC + 1
      WRITE(NOU,100) N1,N2,INC,UT0
      DO 200 J=1,NUM
      N=N1+(J-1)*INC
      UT(II,N)=UT0
  200 CONTINUE
   14 CONTINUE
      IF(NOUT.NE.1)  RETURN
      DO 15 II=1,NPDE
      WRITE(NOU,110)II
      WRITE(NOU,120)
      WRITE(NOU,130) (I,UT(II,I),I=1,NNODE)
   15 CONTINUE
      RETURN
C
   20 FORMAT(//,1X,'INPUT INITIAL SOLUTION DATA FOR EQUATION = ',I3)
   30 FORMAT(1X,' N1    N2    INC       U0')
   40 FORMAT(I5,2X,I5,2X,I5,3X,1PE11.3)
   50 FORMAT(/,1X,'GENERATED INITIAL SOLUTION FOR EQUATION = ',I3)
   60 FORMAT(1X,3('  NODE NO.      U0    '))
   70 FORMAT(3(3X,I5,5X,1PE12.4))
   80 FORMAT(//,1X,'INPUT INITIAL DERIVATIVE DATA FOR EQUATION = ',I3)
   90 FORMAT(1X,' N1    N2    INC       UT0')
  100 FORMAT(I5,2X,I5,2X,I5,3X,1PE11.3)
  110 FORMAT(/,1X,' GENERATED INITIAL DERIVATIVE FOR EQUATION = ',I3)
  120 FORMAT(1X,3('  NODE NO.      UT0    '))
  130 FORMAT(3(3X,I5,5X,1PE12.4))
  210 WRITE(NOU,220)
  220 FORMAT(//,1X,'ERROR EQUA. NUM. DOES NOT MATCH IN INIT.COND INPUT')
      STOP
      END
C
C
C*************************
      SUBROUTINE RCONST
C*************************
C
C   READS CONSTANTS FOR TIME APPROXIMATIONS
C
C
C   CALLED BY:  PREP
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CONST1/ALPHA,BETA,THETA
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,
     .NINTO,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON /BAND/ IB,IB2,ISYM
C
      IF (NPTYPE .EQ. 3)  GO TO 100
      READ (NIN,*)  THETA
      WRITE (NOU,10)  THETA
      IF (THETA .EQ. 0.0)  ISYM = 1
      RETURN
  100 READ (NIN,*)  ALPHA,BETA
      WRITE (NOU,20)  ALPHA,BETA
      ISYM = 1
      RETURN
C
  10 FORMAT(//,1X,'THE VALUE OF THETA =',1PE11.3)
  20 FORMAT(//,1X,'THE VALUE OF ALPHA =',1PE11.3,/,1X,'THE VALUE OF
    . BETA  =',1PE11.3)
      END
C
C
C**************************************
      SUBROUTINE OUTPL1 (X,NE,NODES,LABEL)
C**************************************
C
C   SAVES GRID INFORMATION ON NFILE FOR A SUBSEQUENT GRID PLOT
C
C   CALLED BY:  PREP
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT
     .,NINTO,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
C
      DIMENSION NE(1),NODES(9,1)
      DIMENSION X(2,1)
C
      CHARACTER*4 LABEL(20)
C
C
      IF (NPLOT .EQ. 1) THEN
      REWIND NFILE
      WRITE(NFILE,50) LABEL
      WRITE(NFILE,100)NNODE,NELEM,NPDE
      ENDIF
      DO 20 N=1,NNODE
      IF (NPLOT .EQ. 1) WRITE(NFILE,300)X(1,N),X(2,N)
  20 CONTINUE
      DO 10 N=1,NELEM
      NEN=NE(N)
      IF (NPLOT.EQ.1) WRITE(NFILE,200)NEN,(NODES(I,N),I=1,NEN)
  10 CONTINUE
      RETURN
  50 FORMAT(20A4)
 100 FORMAT(3I5)
 300 FORMAT(2(2X,E12.6))
 200 FORMAT(16I5)
      END
C
C
C****************************************
      SUBROUTINE CALBAN ( NODES,NE,L2)
C****************************************
C
C.... CALCULATES HALF-BAND OR FULL-BAND WIDTH
C
C   CALLED BY :  PREP
C
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
     .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON /BAND/ IB,IB2,ISYM
C
      DIMENSION NODES(9,1),NE(1)
C
      IB = 0
      DO 200 NEL = 1, NELEM
      MAX = 0
      MIN = 100000
      N = NE(NEL)
C
      DO 100 I = 1, N
      IF (NODES(I,NEL) .GT. MAX) MAX = NODES(I,NEL)
  100 IF (NODES(I,NEL) .LT. MIN) MIN = NODES(I,NEL)
      NDIF = MAX - MIN
      IF (NDIF .GT. IB) IB = NDIF
  200 CONTINUE
      IB = IB + 1
      IF (ISYM .EQ. 2) IB2 = 2.*IB - 1
      IF (ISYM.EQ.1 .AND. IB.GT.L2) GO TO 310
      IF (ISYM.EQ.2 .AND. IB2.GT.L2) GO TO 320
      IF (ISYM.EQ.1) WRITE(NOU,5010) IB
      IF (ISYM.EQ.2) WRITE(NOU,5020) IB2
      RETURN
  310 WRITE(NOU,5030) IB
      STOP
  320 WRITE(NOU,5040) IB2
      STOP
 5010 FORMAT(//,' THE MAXIMUM HALF-BAND-WIDTH: IB ',I3)
 5020 FORMAT(//,' THE MAXIMUM FULL-BAND-WIDTH: IB2 =',I4)
 5030 FORMAT(//,' THE HALF-BAND-WIDTH =',I5,
     .' EXCEEDS THE MAXIMUM ALLOWABLE',//)
 5040 FORMAT(//,' THE FULL-BAND-WIDTH =',I5,
     .' EXCEEDS THE MAXIMUM ALLOWABLE',//)
C
      END
C
C
C**********************************************************************
      SUBROUTINE PROS (NODBC1,VBC1,NELBC,NSIDE,VBC2,NPT,VPT,NE,MAT,NODES
     .,NINT,GK,GF,GFBC,X,U,L1,L2,PROP,TIME,IIP,NSTEP,UOLD,IMAT,
     .VAR,TVAR,ITER,U1,UU1,U1OLD,UU1OLD,IEQ,UELEM,NBC1,NBC2,AMUST,SIGMA)
C**********************************************************************
C
C     PROCESSOR ROUTINE:  FORMS AND SOLVES FINITE ELEMENT EQUATIONS
C
C     CALLED BY:  MAIN
C
C     CALLS    : SETINT, FORMKF, APLYBC, SOLVE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
```

```
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
      COMMON/TIMES/T0,TF,DELTAT
C
      DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
      DIMENSION GK(L1,1),GF(1),GFBC(1),X(2,1),U(10,1),UOLD(10,1)
      DIMENSION NODBC1(10,1),VBC1(10,1),NELBC(10,1),NSIDE(10,1),
     .VBC2(10,2,1),NPT(10,1),VPT(10,1)
      DIMENSION PROP(10,10,1)
      DIMENSION U1(10,1),UU1(10,1),U1OLD(10,1),UU1OLD(10,1)
      DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
      DIMENSION UELEM(10,1)
      DIMENSION NBC1(1),NBC2(1)
      DIMENSION AMUST(1)
      DIMENSION SIGMA(1)
C
      CALL SETINT
C
      CALL FORMKF (X,NE,MAT,NODES,NINT,GK,GF,L1,L2,PROP,U,IIP,UOLD,GFBC,
     .IMAT,VAR,TVAR,ITER,UU1,UU1OLD,IEQ,UELEM,SIGMA)
C
      CALL APLYBC (NODBC1,VBC1,NELBC,NSIDE,VBC2,NPT,VPT,NE,MAT,NODES,
     .NINT,GK,GF,GFBC,X,U,UOLD,L1,ITER,U1,U1OLD,IEQ,NBC1,NBC2)
C
      CALL SOLVE (GK,GF,X,U,L1,L2,TIME,IIP,UOLD,ITER,IEQ)
C
C----- INITIALIZE THE RESULT FOR EACH ELEMENT ------
            DO 20 IEE=1,NELEM
             WNODE = 0.0
             DO 30 INN=1,NE(IEE)
              WNODE = WNODE + U(IEQ,NODES(INN,IEE))
30           CONTINUE
             UELEM(IEQ,IEE) = WNODE/NE(IEE)
20          CONTINUE

      RETURN
      END
C
C
C*************************
      SUBROUTINE SETINT
C*************************
C
C   SETS UP QUADRATURE RULES OF ORDERS 1, 2, AND 3.
C
C
C   CALLED BY:  PROS
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CINT/XIQ(9,2,3),WQ(9,3)
C
C  THREE-POINT QUADRATURE
C
      XIQ(1,1,3)=-0.7745966692
      XIQ(2,1,3)=0.0
      XIQ(3,1,3)=-XIQ(1,1,3)
      XIQ(4,1,3)=-XIQ(3,1,3)
      XIQ(5,1,3)=0.0
      XIQ(6,1,3)=-XIQ(4,1,3)
      XIQ(7,1,3)=-XIQ(6,1,3)
      XIQ(8,1,3)=XIQ(5,1,3)
      XIQ(9,1,3)=-XIQ(7,1,3)
      XIQ(1,2,3)=-0.7745966692
      XIQ(2,2,3)=XIQ(1,2,3)
      XIQ(3,2,3)=XIQ(2,2,3)
      XIQ(4,2,3)=0.0
      XIQ(5,2,3)=XIQ(4,2,3)
      XIQ(6,2,3)=XIQ(5,2,3)
      XIQ(7,2,3)=0.7745966692
      XIQ(8,2,3)=XIQ(7,2,3)
      XIQ(9,2,3)=XIQ(8,2,3)
      WQ(1,3)=0.3086419753
      WQ(2,3)=0.4938271605
      WQ(3,3)=WQ(1,3)
      WQ(4,3)=WQ(2,3)
      WQ(5,3)=0.7901234567
      WQ(6,3)=WQ(4,3)
      WQ(7,3)=WQ(3,3)
      WQ(8,3)=WQ(6,3)
      WQ(9,3)=WQ(7,3)
C
C   TWO-POINT QUADRATURE
C
      XIQ(1,1,2)=-0.5773502692
      XIQ(2,1,2)=-XIQ(1,1,2)
      XIQ(3,1,2)=XIQ(1,1,2)
      XIQ(4,1,2)=XIQ(2,1,2)
      XIQ(1,2,2)=-0.5773502692
      XIQ(2,2,2)=XIQ(1,2,2)
      XIQ(3,2,2)=-XIQ(1,2,2)
      XIQ(4,2,2)=-XIQ(1,2,2)
      WQ(1,2)=1.0
      WQ(2,2)=1.0
      WQ(3,2)=1.0
      WQ(4,2)=1.0
C
C   ONE-POINT QUADRATURE
C
      XIQ(1,1,1)=0.
      XIQ(1,2,1)=0.
      WQ(1,1)=4.0
C
      RETURN
      END
C
C
C**********************************************************************
      SUBROUTINE FORMKF (X,NE,MAT,NODES,NINT,GK,GF,L1,L2,PROP,U,IIP,UOLD
     .,GFBC,IMAT,VAR,TVAR,ITER,UU1,UU1OLD,IEQ,UELEM,SIGMA)
C**********************************************************************
C
C   SETS UP GLOBAL MATRIX K AND GLOBAL VECTOR F
C
```

```
C
C   CALLED BY:  PROS
C
C   CALLS    : ELEM, USET1, USET2, ASSMB
C
C
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
     COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
     COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
    ..NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
     COMMON/CINT/XIQ(9,2,3),WQ(9,3)
     COMMON /BAND/ IB,IB2,ISYM
     COMMON/MAX/MAXEL,MAXNOD,MAXEBN,MAXNBS,MAXPTL,MAXMAT,MAXIB
     COMMON/TIMES/T0,TF,DELTAT,NSTEP
C
     INCLUDE 'THVAR.H'
C
     DIMENSION X(2,1),U(10,1),UOLD(10,1),UEOLD(9)
     DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
     DIMENSION GK(L1,1),GF(1),GFBC(1)
     DIMENSION EK(9,9),EF(9),XX(2,9)
     DIMENSION PROP(10,10,1),UU1(10,1),UU1OLD(10,1)
     DIMENSION EC(9,9),AE(9,9),FE(9),UE(9),EM(9,9)
     DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
     DIMENSION UELEM(10,1)
     DIMENSION SIGMA(1)
C
C   INITIALIZE THE ARRAYS
C
     DO 12 I = 1, NNODE
     GF(I) = 0.0
C     IF (NSTEP .GT. 1)  GO TO 12
     GFBC(I) = 0.0
     DO 10 J = 1, L2
     GK(I,J) = 0.0
   10 CONTINUE
   12 CONTINUE
     NLT=7
     DO 50 NEL=1,NELEM
     N=NE(NEL)
     NL1=NINT(NEL)
     IF(NL1.EQ.1) NL=1
     IF(NL1.EQ.2) NL=4
     IF(NL1.EQ.3) NL=9
     DO 15 I=1,N
     XX(1,I)=X(1,NODES(I,NEL))
   15 XX(2,I)=X(2,NODES(I,NEL))
     UELEM(IEQ,NEL) = 0.0
     DO 17 I = 1,N
     UELEM(IEQ,NEL) = UELEM(IEQ,NEL) + U(IEQ,NODES(I,NEL))
17   CONTINUE
     UELEM(IEQ,NEL) = UELEM(IEQ,NEL)/DBLE(N)
     NELE = NEL
C
     IF(FLOWTYPE .EQ. 'TURBULENT' .AND. IEQ .EQ. 1
    .     .AND. NGEOMTYPE .EQ. 1)THEN
      CALL VISC1 (NELE,N,XX,NODES,U,IEQ,ITER,MAT(NEL),SIGMA,X,UELEM)
      ELSEIF(FLOWTYPE .EQ. 'TURBULENT' .AND. IEQ .EQ. 1
     .    .AND. NGEOMTYPE .EQ. 21)THEN
       CALL VISC2122 (NELE,N,XX,NODES,U,IEQ,ITER,MAT(NEL),SIGMA,X,UELEM)
      ELSEIF(FLOWTYPE .EQ. 'TURBULENT' .AND. IEQ .EQ. 1
     .    .AND. NGEOMTYPE .EQ. 22)THEN
       CALL VISC2122 (NELE,N,XX,NODES,U,IEQ,ITER,MAT(NEL),SIGMA,X,UELEM)
      ELSEIF(FLOWTYPE .EQ. 'TURBULENT' .AND. IEQ .EQ. 1)THEN
       CALL VISC (NELE,N,XX,NODES,U,IEQ,ITER,MAT(NEL),SIGMA,X,UELEM)
      ELSEIF(FLOWTYPE .EQ. 'TURBULEKE' .AND. IEQ .EQ. 1)THEN
       CALL VISC (NELE,N,XX,NODES,U,IEQ,ITER,MAT(NEL),SIGMA,X,UELEM)
      ELSEIF(FLOWTYPE .EQ. 'TURBULEKE' .AND. IEQ .EQ. 3)THEN
       CALL VISCKE (NELE,N,XX,NODES,U,IEQ,ITER,MAT(NEL),SIGMA,X,UELEM)
      ENDIF
C
      CALL ELEM (XX,N,EK,EF,NL,XIQ,WQ,MAT(NEL),NELE,X,PROP,EC,EM,
     .       NODES,UELEM,IMAT,VAR,TVAR,IEQ)
C
   30 IF(NPTYPE.EQ.1) GO TO 45

      IF (ITER .EQ. 1) THEN
        DO 61 K=1,NNODE
        UU1(IEQ,K) = U(IEQ,K)
        UU1OLD(IEQ,K) = UOLD(IEQ,K)
61          CONTINUE
      ENDIF

        DO 40 I=1,N
        IF(NPTYPE.EQ.3)  UEOLD(I)=UU1OLD(IEQ,NODES(I,NEL))
   40   UE(I)=UU1(IEQ,NODES(I,NEL))
C
      IF(NPTYPE.EQ.2) CALL USET1 (EK,EF,AE,FE,UE,EC,N)
C
      IF(NPTYPE.EQ.3) CALL USET2 (EK,EF,AE,FE,UE,UEOLD,EC,EM,N)
C
   45 IF(NPTYPE.EQ.1) CALL ASSMB (EK,EF,N,NODES(1,NEL),GK,GF,9,L1)
C
      IF(NPTYPE.NE.1) CALL ASSMB (AE,FE,N,NODES(1,NEL),GK,GF,9,L1)
C
   50 CONTINUE
      IF(IB.GT.MAXIB) GO TO 1000
      RETURN
 1000 WRITE(NOU,1500)IB
 1500 FORMAT(///,5X,'THE HALF-BANDWIDTH IB,',I5,',EXCEEDS THE
     . MAXIMUM ALLOWABLE.')
      STOP
      END
C
C
C*********************************************************************
      SUBROUTINE ELEM (XX,N,EK,EF,NL,XI,W,MAT,NEL,X,PROP,EC,EM,NODES,
     >           UELEM,IMAT,VAR,TVAR,IEQ)
C*********************************************************************
C
C...ELEMENT EQUATIONS FOR QUADRILATERAL ELEMENTS OF FOUR, EIGHT
C  OR NINE NODES.
C
C
C   CALLED BY:  FORMKF
```

```
C    CALLS   : GETMAT, SHAPE4, SHAPE8, SHAPE9
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON /AXIS/ IAXIS
      COMMON /BAND/ IB,IB2,ISYM
C
      DIMENSION PROP(10,10,1),NODES(9,1)
      DIMENSION EK(9,1),EF(1),XI(9,2,3),W(9,3)
      DIMENSION DPSIX(9),DPSIY(9),DXDS(2,2),DSDX(2,2)
      DIMENSION PSI(9),DPSI(9,2),XX(2,9)
      DIMENSION X(2,1)
      DIMENSION EC(9,1),EM(9,1)
      DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
      DIMENSION UELEM(10,1)
C
C.....INITIALIZE ELEMENT ARRAYS
C
      DO 10 I=1,N
      EF(I)=0.0
      DO 10 J=1,N
      EC(I,J)=0.0
      EM(I,J)=0.0
   10 EK(I,J)=0.0
C
      CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT,PROP,
     >          UELEM,IMAT,VAR,TVAR,NEL,IEQ)
C
C    CALCULATE ARTIFICIAL DISSIPATION COEFFICIENTS IF NEEDED
C
      IF(RMU .EQ. 0.0)  GO TO 12
      IF(XM.EQ.0.0 .AND. YM.EQ.0.0)  GO TO 12
C
      XMAX = DMAX1(XX(1,1),XX(1,2),XX(1,3),XX(1,4))
      YMAX = DMAX1(XX(2,1),XX(2,2),XX(2,3),XX(2,4))
      XMIN = DMIN1(XX(1,1),XX(1,2),XX(1,3),XX(1,4))
      YMIN = DMIN1(XX(2,1),XX(2,2),XX(2,3),XX(2,4))
      DXMAX = XMAX - XMIN
      DYMAX = YMAX - YMIN
      SM2 = XM*XM + YM*YM
      DSMS = DABS(DXMAX*XM) + DABS(DYMAX*YM)
      DISSIP = 0.5*RMU*DSMS/SM2
      XK = XK + DISSIP*XM*XM
      YK = YK + DISSIP*YM*YM
      XYK = XYK + DISSIP*XM*YM
   12 CONTINUE
C
C.....BEGIN INTEGRATION POINT LOOP
C
      DO 50 L=1,NL
      IF(NL.EQ.1) NN=1
      IF(NL.EQ.4) NN=2
      IF(NL.EQ.9) NN=3
      IF(N.EQ.4) GO TO 15
      IF(N.EQ.8) GO TO 25
      IF(N.EQ.9) GO TO 35
```

```
C
  15 CALL SHAPE4 (XI(L,1,NN),XI(L,2,NN),N,PSI,DPSI)
C
    GO TO 66
C
  25 CALL SHAPE8 (XI(L,1,NN),XI(L,2,NN),N,PSI,DPSI,NODES(1,NEL))
C
    GO TO 66
C
  35 CALL SHAPE9 (XI(L,1,NN),XI(L,2,NN),N,PSI,DPSI)
C
C.....CALCULATE DXDS
C
  66 CONTINUE
    DO 20 I=1,2
    DO 20 J=1,2
    DXDS(I,J)=0.0
    DO 20 K=1,N
  20 DXDS(I,J)=DXDS(I,J)+DPSI(K,J)*XX(I,K)
C
C.....CALCULATE DSDX
C
    DETJ=DXDS(1,1)*DXDS(2,2)-DXDS(1,2)*DXDS(2,1)
    IF(DETJ.LE.0.0) GO TO 99
    DSDX(1,1)=DXDS(2,2)/DETJ
    DSDX(2,2)=DXDS(1,1)/DETJ
    DSDX(1,2)=-DXDS(1,2)/DETJ
    DSDX(2,1)=-DXDS(2,1)/DETJ
C
C.....CALCULATE D(PSI)/DX
C
    YGA = 0.
    DO 30 I=1,N
    YGA = YGA + PSI(I)*XX(2,I)
    DPSIX(I)=DPSI(I,1)*DSDX(1,1)+DPSI(I,2)*DSDX(2,1)
  30 DPSIY(I)=DPSI(I,1)*DSDX(1,2)+DPSI(I,2)*DSDX(2,2)
    IF(IAXIS.EQ.0) YGA = 1.
C
C.....ACCUMULATE INTEGRATION POINT VALUE OF INTEGRALS.
C
    FAC = DETJ*W(L,NN)*YGA
    DO 40 I = 1, N
    EF(I) = EF(I)+XF*PSI(I)*FAC
    JJ = I
    IF(ISYM.EQ.2)  JJ = 1
    DO 40 J = JJ, N
    EC(I,J) = EC(I,J)+(FAC*XRHO1*PSI(I)*PSI(J))
    EM(I,J) = EM(I,J)+(FAC*XRHO2*PSI(I)*PSI(J))
    EK(I,J) = EK(I,J)+FAC*(XK*DPSIX(I)*DPSIX(J)+YK*DPSIY(I)*DPSIY(J)
   .+XYK*DPSIX(I)*DPSIY(J)+XYK*DPSIY(I)*DPSIX(J)
   .+XM*PSI(I)*DPSIX(J)+YM*PSI(I)*DPSIY(J)+XB*PSI(I)*PSI(J))
  40 CONTINUE
  50 CONTINUE
    IF(ISYM.EQ.2)  RETURN
C
C.....CALCULATE LOWER SYMMETRIC PART OF EK, EC AND EM
C
    DO  60  I = 1, N
```

```
      DO 60 J = 1, I
      EC(I,J) = EC(J,I)
      EM(I,J) = EM(J,I)
   60 EK(I,J) = EK(J,I)
C
      RETURN
   99 WRITE(NOU,100) DETJ,NEL,X
  100 FORMAT(13H BAD JACOBIAN,E10.3,3X,12H ELEMENT NO.,I5,/,1P9E11.3
     ,/,1P9E11.3)
      STOP
      END
C
C
C*******************************************
      SUBROUTINE SHAPE4 (XI,YI,N,PSI,DPSI)
C*******************************************
C
C     CALCULATES SHAPE FUNCTIONS AND THEIR DERIVATIVES
C     FOR FOUR-NODED ELEMENTS
C
C
C     CALLED BY:  ELEM, BCINT, EVAL
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
C
      DIMENSION PSI(9),DPSI(9,2)
C
      IF(N.LT.4.OR.N.GT.4) GO TO 99
C
      PSI(1)=0.25*(1.-XI)*(1.-YI)
      PSI(2)=0.25*(1.+XI)*(1.-YI)
      PSI(3)=0.25*(1.+XI)*(1.+YI)
      PSI(4)=0.25*(1.-XI)*(1.+YI)
C
C     CALCULATES DERIVATIVES OF SHAPE FUNCTIONS
C
      DPSI(1,1)=0.25*(YI-1.)
      DPSI(1,2)=0.25*(XI-1.)
      DPSI(2,1)=0.25*(1.-YI)
      DPSI(2,2)=0.25*(-1.-XI)
      DPSI(3,1)=0.25*(1.+YI)
      DPSI(3,2)=0.25*(1.+XI)
      DPSI(4,1)=0.25*(-1.-YI)
      DPSI(4,2)=0.25*(1.-XI)
      RETURN
   99 WRITE(NOU,100)N,XI,YI
  100 FORMAT (/,'  ERROR IN CALL TO SHAPE4  N= ',I3,1X,2E13.5)
      STOP
      END
C
C
C*************************************************
      SUBROUTINE SHAPE8 (XI,YI,N,PSI,DPSI,NODES)
C*************************************************
C
```

```
C.....CALCULATES SHAPE FUNCTIONS AND THEIR DERIVATIVES
C     FOR BIQUADRATIC EIGHT-NODED ELEMENTS.
C
C     CALLED BY:  ELEM, BCINT, EVAL
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
C
      DIMENSION PSI(9),DPSI(9,2),NODES(8)
C
      IF (N.LT.8.OR.N.GT.8)  GO TO 99
C
      PSI(1)=0.25*(1.-XI)*(1.-YI)*(-1.-XI-YI)
      PSI(2)=0.25*(1.+XI)*(1.-YI)*(-1.+XI-YI)
      PSI(3)=0.25*(1.+XI)*(1.+YI)*(-1.+XI+YI)
      PSI(4)=0.25*(1.-XI)*(1.+YI)*(-1.-XI+YI)
      PSI(5)=0.5*(1.-XI**2)*(1.-YI)
      PSI(6)=0.5*(1.+XI)*(1.-YI**2)
      PSI(7)=0.5*(1.-XI**2)*(1.+YI)
      PSI(8)=0.5*(1.-XI)*(1.-YI**2)
C
C     CALCULATES DERIVATIVES OF SHAPE FUNCTIONS
C
      DPSI(1,1)=0.25*(2.*XI+YI-2.*XI*YI-YI**2)
      DPSI(1,2)=0.25*(2.*YI+XI-2.*XI*YI-XI**2)
      DPSI(2,1)=0.25*(2.*XI-YI-2.*XI*YI+YI**2)
      DPSI(2,2)=0.25*(2.*YI-XI+2.*XI*YI-XI**2)
      DPSI(3,1)=0.25*(2.*XI+YI+2.*XI*YI+YI**2)
      DPSI(3,2)=0.25*(2.*YI+XI+2.*XI*YI+XI**2)
      DPSI(4,1)=0.25*(2.*XI-YI+2.*XI*YI-YI**2)
      DPSI(4,2)=0.25*(2.*YI-XI-2.*XI*YI+XI**2)
      DPSI(5,1)=0.5*(2.*XI*YI-2.*XI)
      DPSI(5,2)=0.5*(XI**2-1.)
      DPSI(6,1)=0.5*(1.-YI**2)
      DPSI(6,2)=0.5*(-2.*YI-2.*XI*YI)
      DPSI(7,1)=0.5*(-2.*XI-2.*XI*YI)
      DPSI(7,2)=0.5*(1.-XI**2)
      DPSI(8,1)=0.5*(YI**2-1.)
      DPSI(8,2)=0.5*(2.*XI*YI-2.*YI)
C
C.....MODIFICATIONS FOR TRIANGULAR ELEMENTS
C
      IF(NODES(1) .NE. NODES(8))  RETURN
      DELH = 0.125*(1.-XI*XI)*(1.-YI*YI)
      DELHX = -0.25*XI*(1.-YI*YI)
      DELHY = -0.25*YI*(1.-XI*XI)
      PSI(6) = PSI(6) - 2.*DELH
      PSI(2) = PSI(2) + DELH
      PSI(3) = PSI(3) + DELH
      DPSI(6,1) = DPSI(6,1) - 2.*DELHX
      DPSI(6,2) = DPSI(6,2) - 2.*DELHY
      DPSI(2,1) = DPSI(2,1) + DELHX
      DPSI(2,2) = DPSI(2,2) + DELHY
      DPSI(3,1) = DPSI(3,1) + DELHX
      DPSI(3,2) = DPSI(3,2) + DELHY
      RETURN
```

```
  99 WRITE(NOU,100)N
 100 FORMAT (/,'   ERROR IN CALL TO SHAPE8  N= ',I3)
     STOP
     END
C
C
C*********************************************
     SUBROUTINE SHAPE9 (XI,YI,N,PSI,DPSI)
C*********************************************
C.....
C   CALCULATES SHAPE FUNCTIONS AND THEIR DERIVATIVES
C   FOR BIQUADRATIC NINE-NODED ELEMENTS
C
C   CALLED BY:  ELEM, BCINT, EVAL
C
C
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
     DIMENSION PSI(9),DPSI(9,2)
C
     COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
C
     IF(N.LT.9.OR.N.GT.9) GO TO 99
C.....
     PSI(1) =0.25*(XI**2-XI)*(YI**2-YI)
     PSI(2) =0.25*(XI**2+XI)*(YI**2-YI)
     PSI(3) =0.25*(XI**2+XI)*(YI**2+YI)
     PSI(4) =0.25*(XI**2-XI)*(YI**2+YI)
     PSI(5) =0.5*(1.-XI**2)*(YI**2-YI)
     PSI(6) =0.5*(XI**2+XI)*(1.-YI**2)
     PSI(7) =0.5*(1.-XI**2)*(YI**2+YI)
     PSI(8) =0.5*(XI**2-XI)*(1.-YI**2)
     PSI(9) =(1.-XI**2)*(1.-YI**2)
C.....
C.....CALCULATES DERIVATIVES OF SHAPE FUNCTIONS
C.....
     DPSI(1,1)=0.25*(2.*XI*YI**2-2.*XI*YI-YI**2+YI)
     DPSI(1,2)=0.25*(2.*YI*XI**2-2.*XI*YI-XI**2+XI)
     DPSI(2,1)=0.25*(2.*XI*YI**2-2.*XI*YI+YI**2-YI)
     DPSI(2,2)=0.25*(2.*YI*XI**2+2.*XI*YI-XI**2-XI)
     DPSI(3,1)=0.25*(2.*XI*YI**2+2.*XI*YI+YI**2+YI)
     DPSI(3,2)=0.25*(2.*YI*XI**2+2.*XI*YI+XI**2+XI)
     DPSI(4,1)=0.25*(2.*XI*YI**2+2.*XI*YI-YI**2-YI)
     DPSI(4,2)=0.25*(2.*YI*XI**2+XI**2-2.*XI*YI-XI)
     DPSI(5,1)=0.5*(2.*XI*YI-2.*XI*YI**2)
     DPSI(5,2)=0.5*(2.*YI-1.-2.*YI*XI**2+XI**2)
     DPSI(6,1)=0.5*(2.*XI-2.*XI*YI**2+1.-YI**2)
     DPSI(6,2)=0.5*(-2.*YI*XI**2-2.*XI*YI)
     DPSI(7,1)=0.5*(-2.*XI*YI**2-2.*XI*YI)
     DPSI(7,2)=0.5*(2.*YI+1.-2.*YI*XI**2-XI**2)
     DPSI(8,1)=0.5*(2.*XI-2.*XI*YI**2-1.+YI**2)
     DPSI(8,2)=0.5*(-2.*YI*XI**2+2.*XI*YI)
     DPSI(9,1)=(-2.*XI+2.*XI*YI**2)
     DPSI(9,2)=(-2.*YI+2.*YI*XI**2)
     RETURN
  99 WRITE(NOU,100)N
 100 FORMAT (/,'   ERROR IN CALL TO SHAPE9  N= ',I3)
     STOP
```

```
      END
C
C
C*********************************************************************
      SUBROUTINE GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT,PROP,
     >              UELEM,IMAT,VAR,TVAR,NEL,IEQ)
C*********************************************************************
C
C   CALCULATES MATERIAL PROPERTIES
C
C   CALLED BY:  ELEM, EVAL
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
C
      DIMENSION PROP(10,10,1)
      DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
      DIMENSION UELEM(10,1)
C
      XK=FUNC(1,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      YK=FUNC(2,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      XYK=FUNC(3,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      XM=FUNC(4,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      YM=FUNC(5,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      XB=FUNC(6,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      XF=FUNC(7,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      RMU=FUNC(8,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      XRHO1=FUNC(9,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
      XRHO2=FUNC(10,IMAT,MAT,UELEM,PROP,VAR,TVAR,NEL,IEQ)
C
      RETURN
      END
C***********************************************
      SUBROUTINE USET1 (EK,EF,AE,FE,UE,EC,N)
C***********************************************
C
C   FORMS NEW ELEMENT MATRICES FOR 1ST ORDER TIME INTEGRATIONS
C
C   CALLED BY:  FORMKF
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/TIMES/T0,TF,DELTAT
      COMMON/CONST1/ALPHA,BETA,THETA
      COMMON/CONST2/THETD,THETM,THETMD,DT2,ADT,BDT,OM2ADT,
     .HM2BPA,OMADT,HPBMA
C
      DIMENSION EK(9,1),EF(1)
      DIMENSION AE(9,1),BE(9,9),FE(1),FE2(9),UE(1)
      DIMENSION EC(9,1)
C
C   INITIALIZE ELEMENT ARRAYS
C
      DO 5 I=1,N
      FE(I)=0.0
      FE2(I)=0.0
      DO 5 J=1,N
      AE(I,J)=0.0
    5 BE(I,J)=0.0
C
C   SET-UP AE
C
      DO 10 I=1,N
      DO 10 J=1,N
      AE(I,J) = AE(I,J) + EC(I,J) + THETD*EK(I,J)
   10 CONTINUE
C
C    SET-UP BE
C
      DO 20 I=1,N
      DO 20 J=1,N
      BE(I,J) = BE(I,J) + EC(I,J) - THETMD*EK(I,J)
   20 CONTINUE
C
C   SET-UP FE
C
      DO 30 I=1,N
      FE(I) = DELTAT*EF(I)
   30 CONTINUE
C
C   MULTIPLY BE AND UE AND ADD RESULT TO FE
C
C   MULTIPLY BE AND UE
C
      DO 40 I=1,N
      SUM=0.0
      DO 50 K=1,N
      SUM=SUM+BE(I,K)*UE(K)
   50 CONTINUE
      FE2(I)=SUM
   40 CONTINUE
C
C   ADD RESULT TO FE
C
      DO 60 I=1,N
      FE(I)=FE(I)+FE2(I)
   60 CONTINUE
C
      RETURN
      END
C
C
C********************************************************
      SUBROUTINE USET2 (EK,EF,AE,FE,UE,UEOLD,EC,EM,N)
C********************************************************
C
C   FORMS NEW ELEMENT MATRICES FOR 2ND ORDER TIME INTEGRATIONS
C
C   CALLED BY:  FORMKF
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
```

```
      COMMON/TIMES/T0,TF,DELTAT
      COMMON/CONST1/ALPHA,BETA
      COMMON/CONST2/THETD,THETM,THETMD,DT2,ADT,BDT,OM2ADT,
     .HM2BPA,OMADT,HPBMA
C
      DIMENSION EK(9,1),EF(1)
      DIMENSION AE(9,1),BE(9,9),CE(9,9),FE(1),FE2(9),FE3(9)
      DIMENSION UE(1),UEOLD(1)
      DIMENSION EC(9,1),EM(9,1)
C
C   INITIALIZE ELEMENT ARRAYS
C
      DO 10 I=1,N
      FE(I)=0.0
      FE2(I)=0.0
      FE3(I)=0.0
      DO 10 J=1,N
      AE(I,J)=0.0
      BE(I,J)=0.0
      CE(I,J)=0.0
   10 CONTINUE
C
C   SET-UP AE
C
      DO 20 I=1,N
      DO 20 J=1,N
      AE(I,J) = EM(I,J) + ADT*EC(I,J) + BDT*EK(I,J)
   20 CONTINUE
C
C   SET-UP BE
C
      DO 30 I=1,N
      DO 30 J=1,N
      BE(I,J) = -2.*EM(I,J) + OM2ADT*EC(I,J) + HM2BPA*EK(I,J)
   30 CONTINUE
C
C   SET-UP CE
C
      DO 40 I=1,N
      DO 40 J=1,N
      CE(I,J) = EM(I,J) - OMADT*EC(I,J) + HPBMA*EK(I,J)
   40 CONTINUE
C
C   SET-UP FE
C
      DO 50 I=1,N
      FE(I) = DT2*EF(I)
   50 CONTINUE
C
C   MULTIPLY BE AND UE; STORE IN FE2
C
      DO 60 I=1,N
      SUM=0.0
      DO 70 J=1,N
      SUM=SUM+BE(I,J)*UE(J)
   70 CONTINUE
      FE2(I)=SUM
   60 CONTINUE
```

```
C
C   MULTIPLY CE AND UEOLD; STORE RESULT IN FE3
C
      DO 80 I=1,N
      SUM=0.0
      DO 90 J=1,N
      SUM=SUM+CE(I,J)*UEOLD(J)
   90 CONTINUE
      FE3(I)=SUM
   80 CONTINUE
C
C   SUM RIGHT-HAND SIDE VECTORS
C
      DO 100 I=1,N
      FE(I)=FE(I)-FE2(I)-FE3(I)
  100 CONTINUE
C
      RETURN
      END
C
C
C*****************************************************
      SUBROUTINE ASSMB (EK,EF,N,NODE,GK,GF,NN,L1)
C*****************************************************
C
C   ASSEMLAGE OF ELEMENT EQUATIONS
C   ADDS EK AND EF TO GK AND GF, RESPECTIVELY.
C
C   CALLED BY:  FORMKF
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON /BAND/ IB,IB2,ISYM
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
C
      DIMENSION EK(NN,1),EF(1),NODE(1),GK(L1,1),GF(1)
C
      DO 20 I = 1, N
      IG = NODE(I)
      GF(IG) = GF(IG) + EF(I)
C       IF (NSTEP .GT. 1)  GO TO 20
      DO 10 J = 1, N
      JG = NODE(J) - IG + 1
      IF (ISYM.EQ.2)  JG = JG + IB - 1
      IF (JG.LE.0) GO TO 10
      GK(IG,JG) = GK(IG,JG) + EK(I,J)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
C
C
C************************************************************************
      SUBROUTINE APLYBC (NODBC1,VBC1,NELBC,NSIDE,VBC2,NPT,VPT,NE,MAT,
     .NODES,NINT,GK,GF,GFBC,X,U,UOLD,L1,ITER,U1,U1OLD,IEQ,NBC1,NBC2)
C************************************************************************
```

```
C
C.....MODIFIES K AND F TO ACCOUNT FOR BOUNDARY CONDITIONS.
C
C     CALLED BY:  PROS
C
C     CALLS    :  BCINT, USETB1, USETB2, ASSMB, DRCHL
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
     ..NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
      COMMON/CONST2/THETD,THETM,THETMD,DT2,ADT,BDT,OM2ADT,
     .HM2BPA,OMADT,HPBMA
      COMMON /BAND/ IB,IB2,ISYM
C
      INCLUDE 'THVAR.H'
C
      DIMENSION NODBC1(10,1),VBC1(10,1),NELBC(10,1),NSIDE(10,1),
     .VBC2(10,2,1),NPT(10,1),VPT(10,1)
      DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
      DIMENSION GK(L1,1),GF(1),GFBC(1),U(10,1),UOLD(10,1)
      DIMENSION X(2,1),U1(10,1),U1OLD(10,1)
      DIMENSION NOD(3), PE(3,3),GAMA(3),XX(2,9),NODA(3)
      DIMENSION NBC1(1),NBC2(1)
C
C.....APPLY POINT LOADS
C
      IF (NPOINT.EQ.0) GO TO 20
      GO TO (10,14,16), NPTYPE
   10 CONTINUE
      DO  11  I = 1, NPOINT
      N = NPT(IEQ,I)
  11  GF(N) = GF(N) + VPT(IEQ,I)
      GO TO 20
   14 CONTINUE
      DO  15  I = 1, NPOINT
      N = NPT(IEQ,I)
  15  GF(N) = GF(N) + DELTAT*VPT(IEQ,I)
      GO TO 20
   16 CONTINUE
      DO  17  I = 1, NPOINT
      N = NPT(IEQ,I)
      GF(N) = GF(N) + DT2*VPT(IEQ,I)
   17 CONTINUE
C
C.....APPLY NATURAL BOUNDARY CONDITIONS
C
   20 IF (NBC2(IEQ).EQ.0) GO TO 70
      DO 60 I=1,NBC2(IEQ)
C
C.....PICK OUT NODES ON SIDE OF ELEMENT
C
      NEL=NELBC(IEQ,I)
      NS=NSIDE(IEQ,I)
      NC=4
   IF(NE(NEL).EQ.6) NC=3
   NOD(1)=NS
   IF(NE(NEL).EQ.4) GO TO 45
   NOD(2)=NS+NC
   NOD(3)=NS+1
   IF(NS.EQ.NC) NOD(3)=1
C
C.....PICK OUT NODAL COORDINATES (8-9 NODE ELEMENTS)
C
   DO 50 J=1,3
   NJ=NOD(J)
 50 NODA(J)=NODES(NJ,NEL)
   GO TO 54
 45 NOD(2)=NS+1
   IF(NS.EQ.NC) NOD(2)=1
C
C.....PICK OUT NODAL COORDINATES (4-NODE ELEMENTS)
C
   DO 53 J=1,2
   NJ=NOD(J)
 53 NODA(J)=NODES(NJ,NEL)
 54 N=NE(NEL)
   DO 55 L=1,N
   XX(1,L)=X(1,NODES(L,NEL))
 55 XX(2,L)=X(2,NODES(L,NEL))
C
C.....CALL BCINT TO CALCULATE BOUNDARY INTEGRALS PE AND GAMA
C
C
   CALL BCINT (VBC2(IEQ,1,I),VBC2(IEQ,2,I),PE,GAMA,NOD,NEL,XX,
   .NS,NE,VBC2,MAT,NODES,NINT,NODBC1,VBC1,NELBC,NSIDE,NPT,VPT)
C
C
C
C.....CALL ASSMB TO ADD PE TO GK AND GAMA TO GF
C
 59 CONTINUE
   NSNO = 3
   IF(N.EQ.4)  NSNO = 2
C
   IF (ITER .EQ. 1) THEN
    DO 61 K=1,NNODE
    U1(IEQ,K) = U(IEQ,K)
    U1OLD(IEQ,K) = UOLD(IEQ,K)
61        CONTINUE
   ENDIF
C
   IF(NPTYPE.EQ.2) CALL USETB1 (PE,GAMA,NSNO,NODA,U1,IEQ)
   IF(NPTYPE.EQ.3) CALL USETB2 (PE,GAMA,NSNO,NODA,U1,U1OLD,IEQ)
   CALL ASSMB (PE,GAMA,NSNO,NODA,GK,GF,3,L1)
C
 60 CONTINUE
C
C.....APPLY ESSENTIAL BOUNDARY CONDITIONS
C
 70 CONTINUE
   IF (NBC1(IEQ) .EQ. 0) RETURN
C    IF (NSTEP .GT. 1)  GO TO 80
```

```
C
      DO 75 I=1,NBC1(IEQ)
      N=NODBC1(IEQ,I)
C
      IF(IEQ.EQ.4.AND.KEMODEL.EQ.'LB')CALL EW_LB (U,X,N,VBC1(IEQ,I))
      IF(IEQ.EQ.4.AND.KEMODEL.EQ.'MY')CALL EW_LB (U,X,N,VBC1(IEQ,I))
      IF(IEQ.EQ.4.AND.KEMODEL.EQ.'HE')CALL EW_HE (U,X,N,VBC1(IEQ,I))
C
      IF (ISYM.EQ.1)  CALL DRCHLS (GK,GF,GFBC,N,VBC1(IEQ,I),L1)
      IF (ISYM.EQ.2)  CALL DRCHLU (GK,GF,N,VBC1(IEQ,I),L1)
C
   75 CONTINUE
   80 CONTINUE
C
C
      DO  90  I = 1, NNODE
      IF(ISYM.EQ.1)  GF(I) = GF(I) - GFBC(I)
   90 CONTINUE
      DO  95  I = 1, NBC1(IEQ)
      N = NODBC1(IEQ,I)
C
      IF(IEQ.EQ.4.AND.KEMODEL.EQ.'LB')CALL EW_LB (U,X,N,VBC1(IEQ,I))
      IF(IEQ.EQ.4.AND.KEMODEL.EQ.'MY')CALL EW_LB (U,X,N,VBC1(IEQ,I))
      IF(IEQ.EQ.4.AND.KEMODEL.EQ.'HE')CALL EW_HE (U,X,N,VBC1(IEQ,I))
C
      VALUE = VBC1(IEQ,I)
      GF(N) = VALUE
   95 CONTINUE
      END
C
C************************************
      SUBROUTINE EW_LB (U,X,I,EWBC)
C************************************
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      INCLUDE 'THVAR.H'
C
      DIMENSION U(10,1),X(2,1)
C
      DS2=DSQRT((X(1,I)-X(1,I-1))**2.+(X(2,I)-X(2,I-1))**2.)
      DS1=DSQRT((X(1,I-1)-X(1,I-2))**2.+(X(2,I-1)-X(2,I-2))**2.)
      DS=(DS2+DS1)/2.
C
      EWBC=VIS/DEN*((U(3,I)-U(3,I-1))/DS2-(U(3,I-1)-U(3,I-2))/DS1)/DS
      EWBC=DMIN1(DABS(EWBC),EWMAX)
C
      RETURN
      END
C
C************************************
      SUBROUTINE EW_HE (U,X,I,EWBC)
C************************************
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
```

```
      INCLUDE 'THVAR.H'
C
      DIMENSION U(10,1),X(2,1)
C
      EWBC=DABS(U(4,I-1))
C
      RETURN
      END
C
C************************************************
      SUBROUTINE USETB1 (PE,GAMA,NSNO,NODA,U1,IEQ)
C************************************************
C
C.....SETS UP CONTRIBUTIONS FROM NATURAL BOUNDARY CONDITIONS TO
C     STEP-BY-STEP INTEGRATION OF FIRST ORDER UNSTEADY PROBLEMS.
C
C     CALLED BY:  APLYBC
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION PE(3,1),GAMA(1),U1(10,1),NODA(1),PE1(3,3)
C
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
      COMMON/CONST1/ALPHA,BETA,THETA
      COMMON/CONST2/THETD,THETM,THETMD,DT2,ADT,BDT,OM2ADT,
     .HM2BPA,OMADT,HPBMA
C
      DO  100  I = 1, NSNO
      GAMA(I) = GAMA(I)*DELTAT
      DO  100  J = 1, NSNO
      PE1(I,J) = THETMD*PE(I,J)
  100 PE(I,J) = THETD*PE(I,J)
C
      DO  250  I = 1, NSNO
      SUM = 0.
      DO  200  J = 1, NSNO
      JJ = NODA(J)
  200 SUM = SUM + PE1(I,J)*U1(IEQ,JJ)
  250 GAMA(I) = GAMA(I) - SUM
C
      RETURN
      END
C
C*****************************************************
      SUBROUTINE USETB2 (PE,GAMA,NSNO,NODA,U1,U1OLD,IEQ)
C*****************************************************
C
C.....SETS UP CONTRIBUTIONS FROM NATURAL BOUNDARY CONDITIONS TO
C     STEP-BY-STEP INTEGRATION OF SECOND ORDER UNSTEADY PROBLEMS.
C
C     CALLED BY:  APLYBC
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION PE(3,1),GAMA(1),U1(10,1),U1OLD(10,1),NODA(1),
     .      PE1(3,3),PE2(3,3)
C
```

```
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
      COMMON/CONST1/ALPHA,BETA,THETA
      COMMON/CONST2/THETD,THETM,THETMD,DT2,ADT,BDT,OM2ADT,
     .HM2BPA,OMADT,HPBMA
C
      DO 100 I = 1, NSNO
      GAMA(I) = DT2*GAMA(I)
      DO 100 J = 1, NSNO
      PE1(I,J) = HM2BPA*PE(I,J)
      PE2(I,J) = HPBMA*PE(I,J)
  100 PE(I,J) = BDT*PE(I,J)
C
      DO 250 I = 1, NSNO
      SUM1 = 0.
      SUM2 = 0.
      DO 200 J = 1, NSNO
      JJ = NODA(J)
      SUM1 = SUM1 + PE1(I,J)*U1(IEQ,JJ)
  200 SUM2 = SUM2 + PE2(I,J)*U1OLD(IEQ,JJ)
  250 GAMA(I) = GAMA(I) - SUM1 - SUM2
C
      RETURN
      END
C
C*************************************************
      SUBROUTINE DRCHLS (GK,GF,GFBC,NEQ,VALUE,L1)
C*************************************************
C
C.....THIS SUBROUTINE MODIFIES THE STIFFNESS MATRIX GK AND LOAD VECTOR
C     GF FORESSENTIAL BOUNDARY CONDITIONS.
C.....MATRIX GK IS SYMMETRIC.
C
C     CALLED BY:  APLYBC
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON /BAND/ IB,IB2,ISYM
      COMMON/CCON/NNODE
      DIMENSION GK(L1,1),GF(1),GFBC(1)
C
      GK(NEQ,1)=1.0
      DO 200 N=2,IB
      NEQN=NEQ-N+1
      IF(NEQN.LT.1) GO TO 150
      GFBC(NEQN) = GFBC(NEQN) + GK(NEQN,N)*VALUE
      GK(NEQN,N) = 0.
  150 CONTINUE
      NEQNN=NEQ+N-1
      IF(NEQNN.GT.NNODE) GO TO 200
      GFBC(NEQNN) = GFBC(NEQNN) + GK(NEQ,N)*VALUE
      GK(NEQ,N) = 0.
  200 CONTINUE
      RETURN
      END
C
C
C*************************************************
```

```
      SUBROUTINE DRCHLU (GK,GF,NEQ,VALUE,L1)
C**************************************************
C
C.....THIS SUBROUTINE MODIFIES THE STIFFNESS MATRIX GK AND LOAD VECTOR
C     GF FOR ESSENTIAL BOUNDARY CONDITIONS.
C.....MATRIX GK IS UNSYMMETRIC.
C
C     CALLED BY:  APLYBC
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON /BAND/ IB,IB2,ISYM
      DIMENSION GK(L1,1),GF(1)
      DO  200  JJ = 1, IB2
      GK(NEQ,JJ) = 0.0
  200 CONTINUE
      GK(NEQ,IB) = 1.0
      GF(NEQ) = VALUE
      RETURN
      END
C.....
C.....
C.....
C************************************************************
      SUBROUTINE SOLVE (GK,GF,X,U,L1,L2,TIME,IIP,UOLD,ITER,IEQ)
C************************************************************
C.....
C.....
C.....SOLVES THE LINEAR EQUATIONS: GK*U = GF FOR NODAL POINT VALUES
C.....
C     CALLED BY:  PROS
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON /BAND/ IB,IB2,ISYM
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
     .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
C
      DIMENSION GK(L1,1),GF(1)
      DIMENSION X(2,1),U(10,1),UOLD(10,1)
C
C     SAVE OLD SOLUTION VECTOR FOR 2ND ORDER TIME PROBLEMS
C
      IF (ITER .EQ. 1) THEN
       DO 10 I=1,NNODE
   10  UOLD(IEQ,I)=U(IEQ,I)
      ENDIF
C
C      IF (ISYM.EQ.1 .AND. NSTEP.EQ.1)  CALL TRIBS (GK,NNODE,IB,L1)
      IF (ISYM.EQ.1)  CALL TRIBS (GK,NNODE,IB,L1)
C
      IF (ISYM.EQ.1)  CALL RHSBS (GK,U,GF,NNODE,IB,L1,IEQ)
C
C      IF (ISYM.EQ.2 .AND. NSTEP.EQ.1)  CALL TRIBU (GK,NNODE,IB,IB2,L1)
      IF (ISYM.EQ.2)  CALL TRIBU (GK,NNODE,IB,IB2,L1)
```

```
C
      IF (ISYM.EQ.2)  CALL RHSBU (GK,U,GF,NNODE,IB,IB2,L1,IEQ)
C
      RETURN
      END
C
C
C****************************************
      SUBROUTINE TRIBS (GK,NEQS,IB,L1)
C****************************************
C
C.....THIS SUBROUTINE TRIANGULARIZES A BANDED AND SYMMETRIC MATRIX GK.
C.....ONLY THE UPPER HALF-BAND OF THE MATRIX IS STORED.
C.....STORAGE IS IN THE FORM OF A RECTANGULAR ARRAY L1 X L2.
C.....THE HALF-BAND WIDTH IS IB.
C.....THE NUMBER OF EQUATIONS IS NEQS.
C
C   CALLED BY:  SOLVE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
C
      DIMENSION GK(L1,1)
C
      DO 120 I=2,NEQS
      M1=MIN0(IB-1,NEQS-I+1)
      DO 120 J=1,M1
      SUM=0.0
      K1=MIN0(I-1,IB-J)
      DO 100 K=1,K1
  100 SUM=SUM+GK(I-K,K+1)*GK(I-K,J+K)/GK(I-K,1)
  120 GK(I,J)=GK(I,J)-SUM
      RETURN
      END
C.....
C.....
C**********************************************
      SUBROUTINE RHSBS (GK,U,GF,NEQS,IB,L1,IEQ)
C**********************************************
C
C.....FOR THE LINEAR SYSTEM GK*U=GF WITH THE MATRIX GK TRIANGULARIZED
C    BY ROUTINE TRIBS, THIS ROUTINE PERFORMS FORWARD SUBSTITUTION
C    INTO GF AND THE BACK SUBSTITUTION  INTO U.
C.....THE HALF-BAND WIDTH OF A IS IB.
C.....THE NUMBER OF EQUATIONS IS NEQS.
C
C   CALLED BY:  SOLVE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
C
      DIMENSION GK(L1,1),U(10,1),GF(1)
      NPI=NEQS+1
      DO 110 I=2,NEQS
      SUM=0.
      K1=MIN0(IB-1,I-1)
      DO 100 K=1,K1
  100 SUM=SUM+GK(I-K,K+1)/GK(I-K,1)*GF(I-K)
  110 GF(I)=GF(I)-SUM
C
C.....BEGIN BACK-SUBSTITUTION
C
      U(IEQ,NEQS)=GF(NEQS)/GK(NEQS,1)
      DO 130 K=2,NEQS
      I=NPI-K
      J1=I+1
      J2=MIN0(NEQS,I+IB-1)
      SUM=0.0
      DO 120 J=J1,J2
      MM=J-J1+2
  120 SUM = SUM + GK(I,MM)*U(IEQ,J)
  130 U(IEQ,I)=(GF(I)-SUM)/GK(I,1)
      RETURN
      END
C
C
C********************************************
      SUBROUTINE TRIBU (GK,NEQS,IB,IB2,L1)
C********************************************
C
C.....REDUCES MATRIX GK BY GAUSS ELIMINATION WHERE GK IS UNSYMMETRIC.
C
C     CALLED BY:  SOLVE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      DIMENSION GK(L1,1)
C
      KMIN = IB + 1
      DO  50  N = 1, NEQS
      IF (GK(N,IB) .EQ. 0.0)  GO TO 60
      IF (GK(N,IB) .EQ. 1.0)  GO TO 20
      C = 1./GK(N,IB)
      DO  10  K = KMIN, IB2
      IF (GK(N,K) .EQ. 0.0)  GO TO 10
      GK(N,K) = C*GK(N,K)
   10 CONTINUE
   20 CONTINUE
      DO  40  L = 2, IB
      JJ = IB - L + 1
      I = N + L - 1
      IF (I .GT. NEQS)  GO TO 40
      IF (GK(I,JJ) .EQ. 0.0)  GO TO 40
      KI = IB + 2 - L
      KF = IB2 + 1 - L
      J = IB
      DO  30  K = KI, KF
      J = J + 1
      IF (GK(N,J) .EQ. 0.0)  GO TO 30
      GK(I,K) = GK(I,K) - GK(I,JJ)*GK(N,J)
   30 CONTINUE
   40 CONTINUE
   50 CONTINUE
      RETURN
   60 WRITE (NOU,5010) N, GK(N,IB)
      STOP
 5010 FORMAT(//,' SET OF EQUATIONS MAY BE SINGULAR ...'/,
     .' DIAGONAL TERM OF EQUATION ',I4,' AT TRIBU IS EQUAL TO ',
     .1PE15.8)
      END
C
C
C*************************************************
      SUBROUTINE RHSBU (GK,U,GF,NEQS,IB,IB2,L1,IEQ)
C*************************************************
C
C.....FOR THE LINEAR SYSTEM GK*U=GF WITH THE MATRIX GK TRIANGULARIZED
C     BY ROUTINE TRIBU, THIS ROUTINE PERFORMS FORWARD SUBUTITUTION
C     INTO GF AND THE BACK SUBSTITUTION  INTO U.
C.....THE HALF-BAND-WIDTH IS IB, FULL-BAND-WIDTH IS IB2.
C.....THE NUMBER OF EQUATIONS IS NEQS.
C
C    CALLED BY:  SOLVE
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      DIMENSION GK(L1,1),U(10,1),GF(1)
C
C.....REDUCE THE LOAD VECTOR GF
C
      DO  30  N = 1, NEQS
      IF (GK(N,IB) .EQ. 0.0)  GO TO 60
      IF (GK(N,IB) .EQ. 1.0)  GO TO 10
      GF(N) = GF(N)/GK(N,IB)
   10 CONTINUE
      DO  20  L = 2, IB
      JJ = IB - L + 1
      I = N + L - 1
      IF (I.GT.NEQS)  GO TO 20
      IF (GK(I,JJ) .EQ. 0.0)  GO TO 20
      GF(I) = GF(I) - GK(I,JJ)*GF(N)
   20 CONTINUE
   30 CONTINUE
C
C     BACK-SUBSTITUTION
C
      LL = IB + 1
      U(IEQ,NEQS) = GF(NEQS)
      DO  50  M = 2, NEQS
      N = NEQS + 1 - M
      SUM = 0.0
      DO  40  L = LL, IB2
      IF(GK(N,L) .EQ. 0.0)  GO TO 40
      K = N + L - IB
      SUM = SUM + GK(N,L)*U(IEQ,K)
   40 CONTINUE
```

```
      U(IEQ,N) = GF(N) - SUM
   50 CONTINUE
      RETURN
   60 WRITE(NOU,5010) N,GK(N,IB)
      STOP
 5010 FORMAT (//,' SET OF EQUATIONS ARE SINGULAR',/,
     .' DIAGONAL TERM OF EQUATION ',I4,' AT RHSBU IS EQUAL TO ',
     .1PE15.8,/)
      END
C.....
C.....
C*******************************************************************
      SUBROUTINE BCINT (PVAL,GAMVAL,PE,GAMA,NOD,NEL,XX,NS,NE,VBC2,MAT,
     .NODES,NINT,NODBC1,VBC1,NELBC,NSIDE,NPT,VPT)
C*******************************************************************
C.....
C.....ACCUMULATES BOUNDARY INTEGRALS FOR QUADRILATERAL ELEMENTS.
C.....
C   CALLED BY:  APLYBC
C
C   CALLS   : SHAPE4, SHAPE8, SHAPE9
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON /AXIS/ IAXIS
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT
C
      DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
      DIMENSION VBC2(10,2,1),NODBC1(10,1),VBC1(10,1),NELBC(10,1),
     .NSIDE(10,1),NPT(10,1),VPT(10,1)
      DIMENSION PE(3,1),GAMA(1)
      DIMENSION XI(3),YI(3),W(3),XX(2,9),NOD(1)
      DIMENSION DPSIX(9),DPSIY(9),DSDX(2,2)
      DIMENSION PSI(9),DPSI(9,2),DXDS(2,2)
C
      N=NE(NEL)
C.....
C.....INITIALIZE ELEMENT ARRAYS
C
      DO 10 I=1,3
      GAMA(I)=0.0
      DO 10 J=1,3
   10 PE(I,J)=0.0
      GO TO (20,30,40,50), NS
   20 YI(1)=-1.0
      YI(2)=YI(1)
      YI(3)=YI(2)
      XI(1)=-0.7745966694
      XI(2)=0.0
      XI(3)=-XI(1)
      W(1)=0.55555555555
      W(2)=0.88888888888
      W(3)=W(1)
      GO TO 60
   30 XI(1)=1.0
      XI(2)=XI(1)
      XI(3)=XI(2)
      YI(1)=-0.7745966694
      YI(2)=0.0
      YI(3)=-YI(1)
      W(1)=0.5555555555
      W(2)=0.8888888888
      W(3)=W(1)
      GO TO 60
   40 YI(1)=1.0
      YI(2)=YI(1)
      YI(3)=YI(2)
      XI(1)=-0.7745966694
      XI(2)=0.0
      XI(3)=-XI(1)
      W(1)=0.5555555555
      W(2)=0.8888888888
      W(3)=W(1)
      GO TO 60
   50 XI(1)=-1.0
      XI(2)=XI(1)
      XI(3)=XI(2)
      YI(1)=-0.7745966694
      YI(2)=0.0
      YI(3)=-YI(1)
      W(1)=0.5555555555
      W(2)=0.8888888888
      W(3)=W(1)
C     BEGIN INTEGRATION POINT LOOP
   60 DO 90 L=1,3
      IF(N.EQ.4) GO TO 15
      IF(N.EQ.8) GO TO 25
      IF(N.EQ.9) GO TO 35
C
   15 CALL SHAPE4 (XI(L),YI(L),N,PSI,DPSI)
C
      GO TO 66
C
   25 CALL SHAPE8 (XI(L),YI(L),N,PSI,DPSI,NODES(1,NEL))
C
      GO TO 66
C
   35 CALL SHAPE9 (XI(L),YI(L),N,PSI,DPSI)
C
C.....CALCULATE DXDS
C
   66 CONTINUE
      KK1 = 3
      IF(N.EQ.4)  KK1 = 2
      DO 70 I=1,2
      DXDS(I,1)=0.0
      DXDS(I,2)=0.
      DO 70 KK=1,KK1
      K=NOD(KK)
      IF(NS.EQ.1.OR.NS.EQ.3) DXDS(I,1)=DXDS(I,1)+DPSI(K,1)*XX(I,K)
   70 IF(NS.EQ.2.OR.NS.EQ.4) DXDS(I,2)=DXDS(I,2)+DPSI(K,2)*XX(I,K)
      NS1 = NS + 1
      IF(NS.EQ.4) NS1 = 1
      IF(N.EQ.4) YGA=XX(2,NS)*PSI(NS)+XX(2,NS1)*PSI(NS1)
      IF(N.NE.4) YGA=XX(2,NS)*PSI(NS)+XX(2,NS+4)*PSI(NS+4)
     .+XX(2,NS1)*PSI(NS1)
C
C.....CALCULATE JACOBIAN DS
C
      IF(NS.EQ.1.OR.NS.EQ.3) DS=DSQRT((DXDS(1,1))**2+(DXDS(2,1))**2)
      IF(NS.EQ.2.OR.NS.EQ.4) DS=DSQRT((DXDS(1,2))**2+(DXDS(2,2))**2)
      IF(IAXIS.EQ.0)  YGA = 1.
      IF(DS.LE.0.) GO TO 99
C
C.....ACCUMULATE INTEGRATION POINT VALUE OF INTEGRALS
C
      FAC=DS*W(L)*YGA
      IF(N.EQ.4) GO TO 85
      DO 80 I=1,3
      IIP=NOD(I)
      GAMA(I)=GAMA(I)-PSI(IIP)*FAC*GAMVAL
      DO 80 J=1,3
      JJ=NOD(J)
   80 PE(I,J)=PE(I,J)+PSI(IIP)*PSI(JJ)*FAC*PVAL
      GO TO 90
   85 DO 87 I=1,2
      IIP=NOD(I)
      GAMA(I)=GAMA(I)-PSI(IIP)*FAC*GAMVAL
      DO 87 J=1,2
      JJ=NOD(J)
   87 PE(I,J)=PE(I,J)+PSI(IIP)*PSI(JJ)*FAC*PVAL
   90 CONTINUE
C     WRITE(NOU,3000) NEL,NS
 3000 FORMAT('  CONVECTION CHECK: NEL,NS:',I5)
C     WRITE(NOU,3010)  (NOD(I),I=1,3)
 3010 FORMAT(' NOD(I)',3I5)
C     WRITE(NOU,3020) ((PE(I,J),J=1,2),I=1,2)
 3020 FORMAT(' PE(I,J)',(1P2E12.3))
      RETURN
   99 WRITE(NOU,110)DS,NEL,X
  110 FORMAT(17H BAD JACOBIAN(DS),1PE10.3,3X,12H ELEMENT NO.,
     .    I5/1P9E10.3/1P9E10.3/)
      STOP
      END
C******************************************************************
      SUBROUTINE POST (X,NE,MAT,NODES,NINT,U,PROP,IIP,IIIP,IIIIP,TIME
     .,XGM,YGM,SX,SY,LABEL,UELEM,IMAT,VAR,TVAR,IEQ)
C******************************************************************
C.....
C.....POSTPROCESSING ROUTINE: EVALUATES AND PRINTS FINITE ELEMENT SOLUTIONS
C.....
C     CALLED BY:  MAIN
C
C     CALLS   :  EVAL
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
     .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/CINT/XIQ(9,2,3),WQ(9,3)
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
C
      DIMENSION PROP(10,10,1)
      DIMENSION X(2,1),U(10,1),XGM(4,1),YGM(4,1),SX(4,1),SY(4,1)
      DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
      DIMENSION XX(2,9)
      DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
      CHARACTER*4 LABEL(20)
      DIMENSION UELEM(10,1)
C
C
C---- PRINT SOLUTION AT EACH TIME STEP ----
      IF(NSTEP.EQ.NSTEPT .AND. NPLOT.NE.0)
     . WRITE(NFILE,110) (U(IEQ,I),I=1,NNODE)
      IF(NOUT .EQ. 1 .AND. NPTYPE.GT.1 .AND. NSTEP.EQ.NSTEPT)
     . WRITE(NOU,80)
C     IF(NOUT .EQ. 1 .AND. NPTYPE.NE.1 .AND. (IIP.EQ.0 .OR. IIIP.EQ.0
C    . .OR. IIIIP.EQ.0)) WRITE(NOU,50) IEQ,NSTEP,TIME
      IF(NOUT .EQ. 1 .AND. NPTYPE.EQ.1) WRITE(NOU,75)
      IF (NPRNT1 .NE. 0) THEN
      IF(NOUT .EQ. 1 .AND. (IIP.EQ.0 .OR. NPTYPE.EQ.1))WRITE(NOU,90)IEQ,
     .                  (I,U(IEQ,I),I=1,NNODE)
      ENDIF
C  50 FORMAT(///,' GENERATED SOLUTION FOR EQUATION = ',I2,/,
C    .1X,'TIME STEP = ',I6,5X,'TIME OF SOLUTION = ',1PE12.4)
   75 FORMAT(//,1X,'THE STEADY-STATE SOLUTION:')
   80 FORMAT(//,1X,'THE SOLUTION AT THE FINAL TIME STEP:')
   90 FORMAT(1H /,1X,'SOLUTION VECTOR  EQ = ',I3,/,1X,3('NODE',8X,
     . 'U',18X)/,3(I5,5X,1PE13.6,10X))
  110 FORMAT(1P8E11.4)
C
      IF (NPRNT2 .NE. 0) THEN
      IF(NOUT .EQ. 1 .AND. (IIIP.EQ.0 .AND. NPTYPE.NE.1))
     . WRITE(NOU,100)
      IF(NOUT .EQ. 1 .AND. NPTYPE.EQ.1) WRITE(NOU,150) IEQ
      ENDIF
      NLT=7
      DO 10 NEL=1,NELEM
      NEL1 = NEL
      N=NE(NEL)
      NL1=NINT(NEL)
      IF(NINTO.EQ.1) NL1=NINT(NEL)-1
      IF(NL1.EQ.0 .OR. NL1.EQ.1) NL=1
      IF(NL1.EQ.2) NL=4
      IF(NL1.EQ.3) NL=9
      IF (NPRNT2 .NE. 0) THEN
C     IF(NOUT .EQ. 1 .AND. (IIIP.EQ.0 .OR. NPTYPE.EQ.1))
C    .    WRITE(NOU,200) NEL
      ENDIF
      DO 20 I=1,N
      XX(1,I)=X(1,NODES(I,NEL))
   20 XX(2,I)=X(2,NODES(I,NEL))
C
      CALL EVAL (NEL1,XIQ,XX,N,NL,MAT,NODES,U,PROP,IIIP,
     .XGM,YGM,SX,SY,UELEM,IMAT,VAR,TVAR,IEQ)
C
C
   10 CONTINUE
```

```
C
 100 FORMAT(//,1X,5HGAUSS,2X,7HX-COORD,4X,7HY-COORD,
   .  7X,3H U ,5X,6H  QX  ,6X,6H  QY  ,5X,6H   Q  ,
   .  6X,6H ANGLE,6X,9H JACOBIAN/
   .,6H POINT)
 150 FORMAT(//,' STEADY-STATE FLUX SOLUTION FOR EQUATION = '
   .,I2,/1X,4H NEL,1X,5HGAUSS,2X,7HX-COORD,4X,7HY-COORD,7X,
   .3H U ,7X,6H  QX  ,
   .6X,6H  QY  ,5X,6H   Q  ,6X,6H ANGLE,6X,9H JACOBIAN/,
   .6H POINT)
C  200 FORMAT(/12H ELEMENT NO.,I3)
    END
C************************************************************
    SUBROUTINE EVAL (NEL,XI,XX,N,NL,MAT,NODES,U,PROP,IIIP
   .,XGM,YGM,SX,SY,UELEM,IMAT,VAR,TVAR,IEQ)
C************************************************************
C.....
C.....CALCULATES U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C.  FOR QUADRILATERAL ELEMENTS
C.....
C   CALLED BY:  POST
C
C   CALLS    :  GETMAT, SHAPE4, SHAPE8, SHAPE9, OUTPL2
C
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
    COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
    COMMON /PLTOUT/ XG,YG,SIGHX,SIGHY
    COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
   .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
    COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
C
    INCLUDE 'THVAR.H'
C
    DIMENSION XGM(4,1),YGM(4,1),SX(4,1),SY(4,1)
    DIMENSION PROP(10,10,1)
    DIMENSION MAT(1),NODES(9,1)
    DIMENSION U(10,1)
    DIMENSION PSI(9),DPSI(9,2),XX(2,9)
    DIMENSION DPSIX(9),DPSIY(9),DXDS(2,2),DSDX(2,2)
    DIMENSION XI(9,2,3),W(1,3)
    DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
    DIMENSION UELEM(10,1)
    DATA PI,PI2 /3.141592654,1.570796327/
C
C.....CALCULATE U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C
    CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),PROP,
   >           UELEM,IMAT,VAR,TVAR,NEL,IEQ)
C
C.....BEGIN INTEGRATION POINT LOOP
C
    DO 50 L=1,NL
    IF(NL.EQ.1) NN=1
    IF(NL.EQ.4) NN=2
    IF(NL.EQ.9) NN=3
    IF(N.EQ.4) GO TO 15
```

```
      IF(N.EQ.8) GO TO 25
      IF(N.EQ.9) GO TO 35
C
   15 CALL SHAPE4 (XI(L,1,NN),XI(L,2,NN),N,PSI,DPSI)
C
      GO TO 66
C
   25 CALL SHAPE8 (XI(L,1,NN),XI(L,2,NN),N,PSI,DPSI,NODES(1,NEL))
C
      GO TO 66
C
   35 CALL SHAPE9 (XI(L,1,NN),XI(L,2,NN),N,PSI,DPSI)
C
C.....CALCULATE DXDS
C
   66 DO 20 I=1,2
      DO 20 J=1,2
      DXDS(I,J)=0.0
      DO 20 K=1,N
   20 DXDS(I,J)=DXDS(I,J)+DPSI(K,J)*XX(I,K)
C
C.....CALCULATE DSDX
C
      DETJ=DXDS(1,1)*DXDS(2,2)-DXDS(1,2)*DXDS(2,1)
      DSDX(1,1)=DXDS(2,2)/DETJ
      DSDX(2,2)=DXDS(1,1)/DETJ
      DSDX(1,2)=-DXDS(1,2)/DETJ
      DSDX(2,1)=-DXDS(2,1)/DETJ
C
C.....CALCULATE D(PSI)/DX
C
      DO 30 I=1,N
      DPSIX(I)=DPSI(I,1)*DSDX(1,1)+DPSI(I,2)*DSDX(2,1)
   30 DPSIY(I)=DPSI(I,1)*DSDX(1,2)+DPSI(I,2)*DSDX(2,2)
      UH=0.
      DUHDX=0.
      DUHDY=0.
      XG=0.
      YG=0.
      DO 10 I=1,N
      XG=XG+PSI(I)*XX(1,I)
      YG=YG+PSI(I)*XX(2,I)
      UH=UH+PSI(I)*U(IEQ,NODES(I,NEL))
      DUHDX=DUHDX+DPSIX(I)*U(IEQ,NODES(I,NEL))
   10 DUHDY=DUHDY+DPSIY(I)*U(IEQ,NODES(I,NEL))
      SIGHX=-XK*DUHDX
      SIGHY=-YK*DUHDY
      LM=L
      NLM=NL
      NELM=NEL
C
      IF(NSTEP.EQ.NSTEPT .AND. NPLOT.NE.0) CALL OUTPL2 (LM,NLM.NELM,
     .XGM.YGM,SX,SY)
C
      SIGMA=DSQRT(SIGHX*SIGHX+SIGHY*SIGHY)
C
C     DETERMINE ANGLE
C
```

```
    IF (DABS(SIGHX) .GT. 1.E-13)  GO TO 75
    ALFA = PI2
    IF (SIGHY .LT. 0.)  ALFA = ALFA + PI
    IF (DABS(SIGHY) .LT. 1.E-13)  ALFA = 0.0
    GO TO 80
  75 ALFA = DATAN(SIGHY/SIGHX)
    IF (SIGHX .LT. 0.)  ALFA = ALFA + PI
  80 CONTINUE
    ALFA=ALFA*57.2958
C
C   PRINT FLUX RELATED SOLUTION
C
    IF (NPRNT2 .NE. 0) THEN
    IF(NOUT .EQ. 1 .AND. (IIIP.EQ.0 .OR. NPTYPE.EQ.1))
   . WRITE(NOU,100)NEL,L,XG,YG,UH,SIGHX,SIGHY,SIGMA,ALFA,DETJ
    ENDIF
C
 100 FORMAT(1X,I4,1X,I1,3X,1PE10.3,2X,1PE10.3,2X,1PE10.3,
   . 4(2X,1PE10.3),2X,1PE10.3)
  50 CONTINUE
    RETURN
    END
C*****************************************************************
    SUBROUTINE POST2 (X,NE,MAT,NODES,NINT,U,PROP,IIP,IIIP,IIIIP,TIME
   .,XGM,YGM,SX,SY,LABEL,UELEM,IMAT,VAR,TVAR,IEQ,WBAR)
C*****************************************************************
C.....
C.....POSTPROCESSING ROUTINE: EVALUATES AND PRINTS FINITE ELEMENT SOLUTIONS
C.....
C   CALLED BY:  MAIN
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
    COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
    COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
   .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
    COMMON/CINT/XIQ(9,2,3),WQ(9,3)
    COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
C
    INCLUDE 'THVAR.H'
C
    DIMENSION PROP(10,10,1)
    DIMENSION X(2,1),U(10,1),XGM(4,1),YGM(4,1),SX(4,1),SY(4,1)
    DIMENSION NE(1),MAT(1),NODES(9,1),NINT(1)
    DIMENSION XX(2,9)
    DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
    CHARACTER*4 LABEL(20)
    DIMENSION UELEM(10,1),WBAR(10,45)
    DATA PI,PI2  /3.141592654,1.570796327/
C
C---- PRINT SOLUTION AT EACH TIME STEP ----
C
    IF(NOUT .EQ. 1 .AND. NPTYPE.EQ.1) WRITE(NLG,75)
    IF (NPRNT1 .NE. 0 .AND. NGEOMTYPE .GE. 2) THEN
     IF(IEQ.EQ.1.AND.NOUT.EQ.1.AND.(IIP.EQ.0.OR.NPTYPE.EQ.1))
   .  WRITE(NLG,89)IEQ,(I,U(IEQ,I)/(-1./VIS*RO**2.*DPDZ),I=1,NNODE)
     IF(IEQ.EQ.1.AND.NOUT.EQ.1.AND.(IIP.EQ.0.OR.NPTYPE.EQ.1))
   .  WRITE(NLG,90)IEQ,(I,U(IEQ,I)/WBAR(1,2),I=1,NNODE)
```

```
    IF(IEQ.EQ.2.AND.NOUT.EQ.1.AND.(IIP.EQ.0.OR.NPTYPE.EQ.1))
   . WRITE(NLG,91)IEQ,(I,(U(IEQ,I)-TW)*2.*PI*AK/QLN,I=1,NNODE)
    IF(IEQ.EQ.2.AND.NOUT.EQ.1.AND.(IIP.EQ.0.OR.NPTYPE.EQ.1))
   . WRITE(NLG,92)IEQ,(I,(TW-U(IEQ,I))/(TW-TBULK(2)),I=1,NNODE)
    ENDIF
 75 FORMAT(//,1X,'THE STEADY-STATE SOLUTION:')
 89 FORMAT(1H /,1X,'NORMALIZED VEL U/(-1./VIS*RO**2.*DPDZ) EQ = ',
   . I3,/,1X,4('NODE',8X,'U',18X)/,4(I5,5X,1PE11.4,10X))
 90 FORMAT(1H /,1X,'NORMALIZED VEL (U/UB) EQ = ',I3,/,1X,4('NODE',8X,
   . 'U',18X)/,4(I5,5X,1PE11.4,10X))
 91 FORMAT(1H /,1X,'NORMALIZED TEMP (T-TW)*2.*PI*AK/QLN EQ = ',I3,/,
   . 1X,4('NODE',8X,'T',18X)/,4(I5,5X,1PE11.4,10X))
 92 FORMAT(1H /,1X,'NORMALIZED TEMP (TW-T)/(TW-TB) EQ = ',I3,/,
   . 1X,4('NODE',8X,'T',18X)/,4(I5,5X,1PE11.4,10X))
    RETURN
    END
C**************************************************
    SUBROUTINE OUTPL2 (L,NL,NEL,XGM,YGM,SX,SY)
C**************************************************
C
C.....SAVES FLUX INFORMATION ON NFILE FOR A SUBSEQUENT CONTOUR
C   AND VECTOR PLOTS
C
C   CALLED BY:  EVAL
C
C
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
    COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
    COMMON /PLTOUT/ XG,YG,SIGX,SIGY
C
    DIMENSION XGM(4,1),YGM(4,1),SX(4,1),SY(4,1)
C
    IF(L.GT.4)  RETURN
    XGM(L,NEL)=XG
    YGM(L,NEL)=YG
    SX(L,NEL)=SIGX
    SY(L,NEL)=SIGY
    IF(L.EQ.NL .AND. NPLOT .NE. 0) WRITE(NFILE,100) NL,(XGM(I,NEL),
   . YGM(I,NEL),SX(I,NEL),SY(I,NEL),I=1,NL)
100 FORMAT(I5/(1P4E11.3))
    RETURN
    END
C
C
C*****************************************************************
C          SUBROUTINE CHEKCONV
C          ---------------
c
C     Checks for convergence and applies relaxation
C
C*****************************************************************
          SUBROUTINE CHEKCONV(U,UITER,DIFFU,RELAX,DIFFMAX)

     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
          COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
   . ,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
```

```
          DIMENSION U(10,1),UITER(10,1),DIFFU(1),RELAX(1),DIFFMAX(1)

C------ DETERMINE AVERAGE TEMPERATURE DIFFERENCE OVER ENTIRE MESH
      DO 15 IEQ=1,NPDE
          DIFFU(IEQ) = 0.0D0
          DIFFMAX(IEQ) = 0.0D0
      UMAX = 0.0D0
          DO 10 J=1,NNODE
           DIFFU(IEQ) = DIFFU(IEQ) + DABS(U(IEQ,J)-UITER(IEQ,J))
           DIFFMAX(IEQ) = DABS(U(IEQ,J)-UITER(IEQ,J))/DABS(UITER(IEQ,J))
       IF (DIFFMAX(IEQ) .GT. UMAX) THEN
        UMAX = DIFFMAX(IEQ)
       ENDIF
           U(IEQ,J) = (1.0-RELAX(IEQ))*UITER(IEQ,J) + RELAX(IEQ)*U(IEQ,J)
           UITER(IEQ,J) = U(IEQ,J)
10        CONTINUE
          DIFFU(IEQ) = DIFFU(IEQ) / NNODE
      DIFFMAX(IEQ) = UMAX
15        CONTINUE

          RETURN
          END

C*********************************************************************
C
C           FUNCTION FUNC
C           ------------
C
C*********************************************************************
      REAL*8 FUNCTION FUNC(MATNUM,IMAT,MAT,UELEM,PROP,VAR,
     .          TVAR,NEL,IEQ)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
      DIMENSION PROP(10,10,1)
          DIMENSION UELEM(10,1)

          IF (MATNUM .EQ. 1) THEN
C------- THIS VARIABLE IS  K11
       IF (IMAT(IEQ,1,MAT) .GT. 1) THEN
        FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,1,MAT) .EQ. 1) THEN
            FUNC = FK11(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,1,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 2) THEN
C------- THIS VARIABLE IS  K22
       IF (IMAT(IEQ,2,MAT) .GT. 1) THEN
        FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,2,MAT) .EQ. 1) THEN
            FUNC = FK22(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,2,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 3) THEN
C------- THIS VARIABLE IS  K12
       IF (IMAT(IEQ,3,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,3,MAT) .EQ. 1) THEN
            FUNC = FK12(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,3,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 4) THEN
C------- THIS VARIABLE IS  M1
      IF (IMAT(IEQ,4,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,4,MAT) .EQ. 1) THEN
            FUNC = FM1(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,4,MAT)
            ENDIF
          ELSEIF (MATNUM .EQ. 5) THEN
C------- THIS VARIABLE IS  M2
      IF (IMAT(IEQ,5,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,5,MAT) .EQ. 1) THEN
            FUNC = FM2(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,5,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 6) THEN
C------- THIS VARIABLE IS  B
      IF (IMAT(IEQ,6,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,6,MAT) .EQ. 1) THEN
            FUNC = FBB(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,6,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 7) THEN
C------- THIS VARIABLE IS  F
      IF (IMAT(IEQ,7,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,7,MAT) .EQ. 1) THEN
            FUNC = FFF(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,7,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 8) THEN
C------- THIS VARIABLE IS  MU
      IF (IMAT(IEQ,8,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,8,MAT) .EQ. 1) THEN
            FUNC = FMU(IEQ,UELEM,NEL,MAT)
           ELSE
            FUNC = PROP(IEQ,8,MAT)
           ENDIF
          ELSEIF (MATNUM .EQ. 9) THEN
C------- THIS VARIABLE IS  RHO1
      IF (IMAT(IEQ,9,MAT) .GT. 1) THEN
       FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
           ELSEIF (IMAT(IEQ,9,MAT) .EQ. 1) THEN
            FUNC = FRHO1(IEQ,UELEM,NEL,MAT)
           ELSE
```

```
              FUNC = PROP(IEQ,9,MAT)
             ENDIF
            ELSEIF (MATNUM .EQ. 10) THEN
C------ THIS VARIABLE IS  RHO2
        IF (IMAT(IEQ,10,MAT) .GT. 1) THEN
        FUNC = PROPRTES(MATNUM,IEQ,MAT,UELEM(IEQ,NEL),IMAT,VAR,TVAR)
             ELSEIF (IMAT(IEQ,10,MAT) .EQ. 1) THEN
              FUNC = FRHO2(IEQ,UELEM,NEL,MAT)
             ELSE
              FUNC = PROP(IEQ,10,MAT)
             ENDIF
            ENDIF
C
            RETURN
            END

C*********************************************************************
C
C             FUNCTION FK11
C             ----------
C
C*********************************************************************
      REAL*8 FUNCTION FK11(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
C
      INCLUDE 'THVAR.H'
C
      DIMENSION UELEM(10,1)

            IF (IEQ .EQ. 1) THEN
             IF (FPROP.EQ.'FIXED')THEN
        FK11 = -1.0*AMUST(NEL) - VIS
             ELSEIF (FPROP.EQ.'FIXTB')THEN
        FK11 = -1.0*AMUST(NEL) - VISF(TAVE)
       ELSE
        FK11 = -1.0*AMUST(NEL) - VISF(UELEM(2,NEL))
       ENDIF
C... fa8.inp, turbulence test model
C            FK11 = -0.01*DEN*UELEM(1,NEL)*DH/2. - VIS
C            FK11 = 0.0
            ELSEIF (IEQ .EQ. 2) THEN
C... e1sfg6.inp, fa8.inp
             IF (FPROP.EQ.'FIXED')THEN
        PR = VIS*CP/AK
        FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
             ELSEIF (FPROP.EQ.'FIXTB')THEN
        PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
        FK11 = -1.0*AKF(TAVE)*AMUST(NEL)/VISF(TAVE)*
     .     PR/PRT(NEL) - AKF(TAVE)
             ELSE
        PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
        FK11 = -1.0*AKF(UELEM(2,NEL))*AMUST(NEL)/VISF(UELEM(2,NEL))*
     .     PR/PRT(NEL) - AKF(UELEM(2,NEL))
             ENDIF
C            FK11 = 0.0
C... fa8.inp, turbulence test model
C         VIST = 0.01*DEN*UELEM(1,NEL)*DH/2. + VIS
```

```
C      FK11 = -1.0*AK*VIST/VIS*PR/PRT(NEL) - AK
C
              ELSEIF (IEQ .EQ. 3) THEN
        SIGMAK = 1.0
               FK11 = - VIS - VISTT(NEL)/SIGMAK
C
              ELSEIF (IEQ .EQ. 4) THEN
               SIGMAE = 1.3
               FK11 = - VIS - VISTT(NEL)/SIGMAE
C
              ELSEIF (IEQ .EQ. 5) THEN
              FK11 = 0.0
              ELSEIF (IEQ .EQ. 6) THEN
              FK11 = 0.0
              ELSEIF (IEQ .EQ. 7) THEN
              FK11 = 0.0
              ELSEIF (IEQ .EQ. 8) THEN
              FK11 = 0.0
              ELSEIF (IEQ .EQ. 9) THEN
              FK11 = 0.0
              ELSEIF (IEQ .EQ. 10) THEN
              FK11 = 0.0
              ENDIF
              RETURN
              END
C*****************************************************************
C
C           FUNCTION FK22
C          -------------
C
C*****************************************************************
      REAL*8 FUNCTION FK22(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
C
      INCLUDE 'THVAR.H'
C
      DIMENSION UELEM(10,1)

             IF (IEQ .EQ. 1) THEN
              IF(FPROP.EQ.'FIXED')THEN
        FK22 = -1.0*AMUST(NEL) - VIS
              ELSEIF (FPROP.EQ.'FIXTB')THEN
        FK22 = -1.0*AMUST(NEL) - VISF(TAVE)
              ELSE
        FK22 = -1.0*AMUST(NEL) - VISF(UELEM(2,NEL))
              ENDIF
C... fa8.inp, turbulence test model
C             FK22 = -0.01*DEN*UELEM(1,NEL)*DH/2. - VIS
C             FK22 = 0.0
             ELSEIF (IEQ .EQ. 2) THEN
C... e1sfg6.inp,fa8.inp
              IF(FPROP.EQ.'FIXED')THEN
        PR = VIS*CP/AK
        FK22 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
              ELSEIF (FPROP.EQ.'FIXTB')THEN
        PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
      FK22 = -1.0*AKF(TAVE)*AMUST(NEL)/VISF(TAVE)*
     .       PR/PRT(NEL) -AKF(TAVE)
             ELSE
      PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
      FK22 = -1.0*AKF(UELEM(2,NEL))*AMUST(NEL)/VISF(UELEM(2,NEL))*
     .       PR/PRT(NEL) -AKF(UELEM(2,NEL))
             ENDIF
C            FK22 = 0.0
C... fa8.inp, turbulence test model
C        VIST = 0.01*DEN*UELEM(1,NEL)*DH/2. + VIS
C        FK22 = -1.0*AK*VIST/VIS*PR/PRT(NEL) - AK
C
            ELSEIF (IEQ .EQ. 3) THEN
      SIGMAK = 1.0
             FK22 = - VIS - VISTT(NEL)/SIGMAK
C
            ELSEIF (IEQ .EQ. 4) THEN
             SIGMAE = 1.3
             FK22 = - VIS - VISTT(NEL)/SIGMAE
C
            ELSEIF (IEQ .EQ. 5) THEN
            FK22 = 0.0
            ELSEIF (IEQ .EQ. 6) THEN
            FK22 = 0.0
            ELSEIF (IEQ .EQ. 7) THEN
            FK22 = 0.0
            ELSEIF (IEQ .EQ. 8) THEN
            FK22 = 0.0
            ELSEIF (IEQ .EQ. 9) THEN
            FK22 = 0.0
            ELSEIF (IEQ .EQ. 10) THEN
            FK22 = 0.0
            ENDIF
            RETURN
            END

C*********************************************************************
C
C              FUNCTION FK12
C             -------------
C
C*********************************************************************
      REAL*8 FUNCTION FK12(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C

            IF (IEQ .EQ. 1) THEN
            FK12 = 0.0
            ELSEIF (IEQ .EQ. 2) THEN
            FK12 = 0.0
            ELSEIF (IEQ .EQ. 3) THEN
            FK12 = 0.0
            ELSEIF (IEQ .EQ. 4) THEN
            FK12 = 0.0
```

```
          ELSEIF (IEQ .EQ. 5) THEN
          FK12 = 0.0
          ELSEIF (IEQ .EQ. 6) THEN
          FK12 = 0.0
          ELSEIF (IEQ .EQ. 7) THEN
          FK12 = 0.0
          ELSEIF (IEQ .EQ. 8) THEN
          FK12 = 0.0
          ELSEIF (IEQ .EQ. 9) THEN
          FK12 = 0.0
          ELSEIF (IEQ .EQ. 10) THEN
          FK12 = 0.0
          ENDIF
          RETURN
          END

C*********************************************************************
C
C           FUNCTION FM1
C          ------------
C
C*********************************************************************
      REAL*8 FUNCTION FM1(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C

          IF (IEQ .EQ. 1) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 2) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 3) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 4) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 5) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 6) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 7) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 8) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 9) THEN
          FM1 = 0.0
          ELSEIF (IEQ .EQ. 10) THEN
          FM1 = 0.0
          ENDIF
          RETURN
          END

C*********************************************************************
C
C           FUNCTION FM2
C          ------------
```

```
C
C*******************************************************************
      REAL*8 FUNCTION FM2(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C

          IF (IEQ .EQ. 1) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 2) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 3) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 4) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 5) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 6) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 7) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 8) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 9) THEN
          FM2 = 0.0
          ELSEIF (IEQ .EQ. 10) THEN
          FM2 = 0.0
          ENDIF
          RETURN
          END

C*******************************************************************
C
C            FUNCTION FBB
C           ------------
C
C*******************************************************************
      REAL*8 FUNCTION FBB(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/YSPLUS/YPLUSA,SPLUSA,ALLA,YA,SA,DFPA,DFCA,ALPA,ALCA,
     . TWYA,TWSA
      DIMENSION YPLUSA(2350)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C
          IF (IEQ .EQ. 1) THEN
          FBB = 0.0
C         FBB = UELEM(1,NEL)*315.0
          ELSEIF (IEQ .EQ. 2) THEN
C USED WITH FFF
           IF(FPROP.EQ.'FIXED')THEN
        FBB = -1.*DEN*CP*UELEM(1,NEL)/DZ
           ELSEIF (FPROP.EQ.'FIXTB')THEN
```

```
       FBB = -1.*DENF(TAVE)*CPF(TAVE)*UELEM(1,NEL)/DZ
           ELSE
       FBB = -1.*DENF(UELEM(2,NEL))*CPF(UELEM(2,NEL))*UELEM(1,NEL)/DZ
           ENDIF
C         FBB = 0.0
Csys another way: applicalbe for a tube geometry, laminar flow,
C           constant tube wall temperature
C         FBB = (DEN*CP*UELEM(1,NEL))/
C    .    (TW-TBULK(1))*DTDZ
          ELSEIF (IEQ .EQ. 3) THEN
          FBB = 0.0
          ELSEIF (IEQ .EQ. 4) THEN
           D=C1*F1(NEL)/DABS(UELEM(3,NEL))*VISTT(NEL)*
     .   (GRADX(1,NEL)**2.+GRADY(1,NEL)**2.)
           E=-C2*F2(NEL)*DEN*DABS(UELEM(4,NEL))/DABS(UELEM(3,NEL))
          FBB = E
C      WRITE(52,99)IEQ,NEL,VISTT(NEL),F1(NEL),F2(NEL),FMUKE(NEL),
C    .     YPLUSA(NEL),D,E,FBB
C99     FORMAT(1X,2I4,14(1X,E10.4))
          ELSEIF (IEQ .EQ. 5) THEN
          FBB = 0.0
          ELSEIF (IEQ .EQ. 6) THEN
          FBB = 0.0
          ELSEIF (IEQ .EQ. 7) THEN
          FBB = 0.0
          ELSEIF (IEQ .EQ. 8) THEN
          FBB = 0.0
          ELSEIF (IEQ .EQ. 9) THEN
          FBB = 0.0
          ELSEIF (IEQ .EQ. 10) THEN
          FBB = 0.0
          ENDIF
          RETURN
          END

C*********************************************************************
C
C           FUNCTION FFF
C          -----------
C
C*********************************************************************
      REAL*8 FUNCTION FFF(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/YSPLUS/YPLUSA,SPLUSA,ALLA,YA,SA,DFPA,DFCA,ALPA,ALCA,
     . TWYA,TWSA
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO,
     .     NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
C
      INCLUDE 'THVAR.H'
C
      COMMON/ELGRID/XG_EL,YG_EL
      DIMENSION UELEM(10,1),XG_EL(2350),YG_EL(2350),YPLUSA(2350)
C
          IF (IEQ .EQ. 1) THEN

C          IF(NEL.EQ.1)THEN
C           VISW = -1.0*AMUST(NEL) - VIS
```

```
C          VISWP1 = -1.0*AMUST(NEL+1) - VIS
C          GX1=GRADX(1,NEL)
C          GX2=GRADX(1,NEL+1)
C          TWX=(GX2*VISWP1-GX1*VISW)/(XG_EL(NEL+1)-XG_EL(NEL))
C          GY1=GRADY(1,NEL)
C          GY2=GRADY(1,NEL+1)
C          TWY=(GY2*VISWP1-GY1*VISW)/(YG_EL(NEL+1)-YG_EL(NEL))
C         ELSEIF(NEL.EQ.NELEM)THEN
C          VISW = -1.0*AMUST(NEL) - VIS
C          VISWM1 = -1.0*AMUST(NEL-1) - VIS
C          GX1=GRADX(1,NEL-1)
C          GX2=GRADX(1,NEL)
C          TWX=(GX2*VISW-GX1*VISWM1)/(XG_EL(NEL)-XG_EL(NEL-1))
C          GY1=GRADY(1,NEL-1)
C          GY2=GRADY(1,NEL)
C          TWY=(GY2*VISW-GY1*VISWM1)/(YG_EL(NEL)-YG_EL(NEL-1))
C         ELSE
C          VISWM1 = -1.0*AMUST(NEL-1) - VIS
C          VISWP1 = -1.0*AMUST(NEL+1) - VIS
C          GX1=GRADX(1,NEL-1)
C          GX2=GRADX(1,NEL+1)
C          TWX=(GX2*VISWP1-GX1*VISWM1)/(XG_EL(NEL+1)-XG_EL(NEL-1))
C          GY1=GRADY(1,NEL-1)
C          GY2=GRADY(1,NEL+1)
C          TWY=(GY2*VISWP1-GY1*VISWM1)/(YG_EL(NEL+1)-YG_EL(NEL-1))
C         ENDIF

        FFF = DPDZ

C     WRITE(52,99)IEQ,NEL,AMUST(NEL),YPLUSA(NEL),
C    .  DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),
C    .  -TWX,-TWY,-FFF,-TWX-TWY-FFF

        ELSEIF (IEQ .EQ. 2) THEN

C         IF(NEL.EQ.1)THEN
C          VIST = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
C          VISTP1 = -1.0*AK*AMUST(NEL+1)/VIS*PR/PRT(NEL+1) - AK
C          GX1=GRADX(2,NEL)
C          GX2=GRADX(2,NEL+1)
C          TTX=(GX2*VISTP1-GX1*VIST)/(XG_EL(NEL+1)-XG_EL(NEL))
C          GY1=GRADY(2,NEL)
C          GY2=GRADY(2,NEL+1)
C          TTY=(GY2*VISTP1-GY1*VIST)/(YG_EL(NEL+1)-YG_EL(NEL))
C         ELSEIF(NEL.EQ.NELEM)THEN
C          VIST = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
C          VISTM1 = -1.0*AK*AMUST(NEL-1)/VIS*PR/PRT(NEL-1) - AK
C          GX1=GRADX(2,NEL-1)
C          GX2=GRADX(2,NEL)
C          TTX=(GX2*VIST-GX1*VISTM1)/(XG_EL(NEL)-XG_EL(NEL-1))
C          GY1=GRADY(2,NEL-1)
C          GY2=GRADY(2,NEL)
C          TTY=(GY2*VIST-GY1*VISTM1)/(YG_EL(NEL)-YG_EL(NEL-1))
C         ELSE
C          VISTM1 = -1.0*AK*AMUST(NEL-1)/VIS*PR/PRT(NEL-1) - AK
C          VISTP1 = -1.0*AK*AMUST(NEL+1)/VIS*PR/PRT(NEL+1) - AK
C          GX1=GRADX(2,NEL-1)
C          GX2=GRADX(2,NEL+1)
```

```
C           TTX=(GX2*VISTP1-GX1*VISTM1)/(XG_EL(NEL+1)-XG_EL(NEL-1))
C           GY1=GRADY(2,NEL-1)
C           GY2=GRADY(2,NEL+1)
C           TTY=(GY2*VISTP1-GY1*VISTM1)/(YG_EL(NEL+1)-YG_EL(NEL-1))
C          ENDIF

C     FFF = 0.0
C          FFF = DEN*CP*WAVE*(TAVE-TIN)
C.. fa8, constant heat flux, 0.6m length
C          FFF = -1.*(DEN*CP*UELEM(1,NEL))*TIN/DZ
Csys applicable for the geometry of tube and annulus, laminar flow,
C   constant heat flux, e1sfg6.inp
           IF(FPROP.EQ.'FIXED')THEN
            FFF = DEN*CP*UELEM(1,NEL)*DTDZ
           ELSEIF (FPROP.EQ.'FIXTB')THEN
            FFF = (DENF(TAVE)*CPF(TAVE)*UELEM(1,NEL))*DTDZ
           ELSE
            FFF = (DENF(UELEM(2,NEL))*CPF(UELEM(2,NEL))*UELEM(1,NEL))*DTDZ
           ENDIF
C.. e1sfg6.inp constant wall
C          FFF = (DEN*CP*UELEM(1,NEL))*
C    .     ((TW-UELEM(2,NEL))/(TW-TBULK(2)))*DTDZ
C  applicalbe for a tube geometry, laminar flow, constant tube wall
C   temperature
C          FFF = (DEN*CP*UELEM(1,NEL))*
C    .     ((TW-UELEM(2,NEL))/(TW-TBULK(1)))*DTDZ
C  another way (use FBB and FFF): applicalbe for a tube geometry,
C   laminar flow, constant tube wall temperature
C          FFF = (DEN*CP*UELEM(1,NEL))*
C    .     TW/(TW-TBULK(1))*DTDZ
C
C      WRITE(52,99)IEQ,NEL,AMUST(NEL),YPLUSA(NEL),
C    .  DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),
C    .  -TTX,-TTY,-FFF,-TTX-TTY-FFF
           ELSEIF (IEQ .EQ. 3) THEN

      SIGMAK = 1.0
           IF(NEL.EQ.1)THEN
            VISK = - VIS - VISTT(NEL)/SIGMAK
            VISKP1 = - VIS - VISTT(NEL+1)/SIGMAK
            GX1=GRADX(3,NEL)
            GX2=GRADX(3,NEL+1)
            TKX=(GX2*VISKP1-GX1*VISK)/(XG_EL(NEL+1)-XG_EL(NEL))
            GY1=GRADY(3,NEL)
            GY2=GRADY(3,NEL+1)
            TKY=(GY2*VISKP1-GY1*VISK)/(YG_EL(NEL+1)-YG_EL(NEL))
           ELSEIF(NEL.EQ.NELEM)THEN
            VISK = - VIS - VISTT(NEL)/SIGMAK
            VISKM1 = - VIS - VISTT(NEL-1)/SIGMAK
            GX1=GRADX(3,NEL-1)
            GX2=GRADX(3,NEL)
            TKX=(GX2*VISK-GX1*VISKM1)/(XG_EL(NEL)-XG_EL(NEL-1))
            GY1=GRADY(3,NEL-1)
            GY2=GRADY(3,NEL)
            TKY=(GY2*VISK-GY1*VISKM1)/(YG_EL(NEL)-YG_EL(NEL-1))
           ELSE
            VISKP1 = - VIS - VISTT(NEL+1)/SIGMAK
            VISKM1 = - VIS - VISTT(NEL-1)/SIGMAK
```

```
          GX1=GRADX(3,NEL-1)
          GX2=GRADX(3,NEL+1)
          TKX=(GX2*VISKP1-GX1*VISKM1)/(XG_EL(NEL+1)-XG_EL(NEL-1))
          GY1=GRADY(3,NEL-1)
          GY2=GRADY(3,NEL+1)
          TKY=(GY2*VISKP1-GY1*VISKM1)/(YG_EL(NEL+1)-YG_EL(NEL-1))
         ENDIF

         A= -VISTT(NEL)*(GRADX(1,NEL)**2.+GRADY(1,NEL)**2.)
         B= DEN*DABS(UELEM(4,NEL))

         IF(KEMODEL.EQ.'LS'.OR.KEMODEL.EQ.'NA')THEN
          IF(NEL.EQ.1)THEN
           C=VIS/2./DABS(UELEM(3,NEL))*((UELEM(3,NEL+1)-UELEM(3,NEL))/
     .      (YY(NEL+1)-YY(NEL)))**2.
          ELSEIF(NEL.EQ.NELEM)THEN
           C=VIS/2./DABS(UELEM(3,NEL))*((UELEM(3,NEL)-UELEM(3,NEL-1))/
     .      (YY(NEL)-YY(NEL-1)))**2.
          ELSE
           C=VIS/2./DABS(UELEM(3,NEL))*((UELEM(3,NEL+1)-UELEM(3,NEL-1))/
     .      (YY(NEL+1)-YY(NEL-1)))**2.
          ENDIF
C
         ELSEIF(KEMODEL.EQ.'CH')THEN
      C=2.*VIS*UELEM(3,NEL)/YY(NEL)**2.
C
         ELSE
          C=0.0
         ENDIF

         FFF= A+B+C
C
C     WRITE(52,99)IEQ,NEL,YPLUSA(NEL),
C   .   DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),
C   .   GRADX(1,NEL),GRADY(1,NEL),
C   .   GRADX(2,NEL),GRADY(2,NEL),GRADX(3,NEL),GRADY(3,NEL),
C   .   GRADX(4,NEL),GRADY(4,NEL)
C      WRITE(52,99)IEQ,NEL,VISTT(NEL),YPLUSA(NEL),
C   .  DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),-TKX,-TKY,-A,-B,-C,
C   .          -TKX-TKY-A-B-C
        ELSEIF (IEQ .EQ. 4) THEN
C
         SIGMAE = 1.3
         IF(NEL.EQ.1)THEN
          VISE = - VIS - VISTT(NEL)/SIGMAE
          VISEP1 = - VIS - VISTT(NEL+1)/SIGMAE
          GX1=GRADX(4,NEL)
          GX2=GRADX(4,NEL+1)
          TEX=(GX2*VISEP1-GX1*VISE)/(XG_EL(NEL+1)-XG_EL(NEL))
          GY1=GRADY(4,NEL)
          GY2=GRADY(4,NEL+1)
          TEY=(GY2*VISEP1-GY1*VISE)/(YG_EL(NEL+1)-YG_EL(NEL))
         ELSEIF(NEL.EQ.NELEM)THEN
          VISE = - VIS - VISTT(NEL)/SIGMAE
          VISEM1 = - VIS - VISTT(NEL-1)/SIGMAE
          GX1=GRADX(4,NEL-1)
          GX2=GRADX(4,NEL)
          TEX=(GX2*VISE-GX1*VISEM1)/(XG_EL(NEL)-XG_EL(NEL-1))
```

```
       GY1=GRADY(4,NEL-1)
       GY2=GRADY(4,NEL)
       TEY=(GY2*VISE-GY1*VISEM1)/(YG_EL(NEL)-YG_EL(NEL-1))
      ELSE
       VISEP1 = - VIS - VISTT(NEL+1)/SIGMAE
       VISEM1 = - VIS - VISTT(NEL-1)/SIGMAE
       GX1=GRADX(4,NEL-1)
       GX2=GRADX(4,NEL+1)
       TEX=(GX2*VISEP1-GX1*VISEM1)/(XG_EL(NEL+1)-XG_EL(NEL-1))
       GY1=GRADY(4,NEL-1)
       GY2=GRADY(4,NEL+1)
       TEY=(GY2*VISEP1-GY1*VISEM1)/(YG_EL(NEL+1)-YG_EL(NEL-1))
      ENDIF


      D=-C1*F1(NEL)*DABS(UELEM(4,NEL))/DABS(UELEM(3,NEL))*VISTT(NEL)*
     .(GRADX(1,NEL)**2.+GRADY(1,NEL)**2.)
      E=C2*F2(NEL)*DEN*DABS(UELEM(4,NEL))**2./DABS(UELEM(3,NEL))
C
      IF(NEL.EQ.1)THEN
       GX1=GRADX(1,NEL)
       GX2=GRADX(1,NEL+1)
       GGX=(GX2-GX1)/(XG_EL(NEL+1)-XG_EL(NEL))
       GY1=GRADY(1,NEL)
       GY2=GRADY(1,NEL+1)
       GGY=(GY2-GY1)/(YG_EL(NEL+1)-YG_EL(NEL))
       GGXY=(GY2-GY1)/(XG_EL(NEL+1)-XG_EL(NEL))
      ELSEIF(NEL.EQ.NELEM)THEN
       GX1=GRADX(1,NEL-1)
       GX2=GRADX(1,NEL)
       GGX=(GX2-GX1)/(XG_EL(NEL)-XG_EL(NEL-1))
       GY1=GRADY(1,NEL-1)
       GY2=GRADY(1,NEL)
       GGY=(GY2-GY1)/(YG_EL(NEL)-YG_EL(NEL-1))
       GGXY=(GY2-GY1)/(XG_EL(NEL)-XG_EL(NEL-1))
      ELSE
       GX1=GRADX(1,NEL-1)
       GX2=GRADX(1,NEL+1)
       GGX=(GX2-GX1)/(XG_EL(NEL+1)-XG_EL(NEL-1))
       GY1=GRADY(1,NEL-1)
       GY2=GRADY(1,NEL+1)
       GGY=(GY2-GY1)/(YG_EL(NEL+1)-YG_EL(NEL-1))
       GGXY=(GY2-GY1)/(XG_EL(NEL+1)-XG_EL(NEL-1))
      ENDIF

      IF(KEMODEL.EQ.'LS')THEN
C...Launder and Sharma (1974)
       F = -2.*VIS*VISTT(NEL)/DEN*(GGX+GGY+2.*GGXY)**2.
      ELSEIF(KEMODEL.EQ.'NA')THEN
C... Nagano (1987)
       F = -(1.-FMUKE(NEL))*VIS*VISTT(NEL)/DEN*(GGX+GGY+2.*GGXY)**2.
      ELSEIF(KEMODEL.EQ.'CH')THEN
C... Chien (1982)
       F = 2.*UELEM(4,NEL)*VIS/YY(NEL)**2.*DEXP(-0.5*YPLUSA(NEL))
      ELSE
       F=0.0
      ENDIF
C
```

```
         FFF = D+F
C
C      WRITE(52,99)IEQ,NEL,VISTT(NEL),YPLUSA(NEL),
C   .   DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),-TEX,-TEY,-D,-E,-F,
C   .         -TEX-TEY-D-E-F
C      WRITE(52,99)IEQ,NEL,YPLUSA(NEL),AMUST(NEL),XG_EL(NEL),
C   .    YG_EL(NEL),DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),
C   .    RO-DSQRT(XG_EL(NEL)**2.+YG_EL(NEL)**2.),YY(NEL),
C   .    UELEM(1,NEL),UELEM(2,NEL),UELEM(3,NEL),UELEM(4,NEL)
C99      FORMAT(1X,2I4,14(1X,E10.4))
         ELSEIF (IEQ .EQ. 5) THEN
         FFF = 0.0
         ELSEIF (IEQ .EQ. 6) THEN
         FFF = 0.0
         ELSEIF (IEQ .EQ. 7) THEN
         FFF = 0.0
         ELSEIF (IEQ .EQ. 8) THEN
         FFF = 0.0
         ELSEIF (IEQ .EQ. 9) THEN
         FFF = 0.0
         ELSEIF (IEQ .EQ. 10) THEN
         FFF = 0.0
         ENDIF
         RETURN
         END

C*****************************************************************
C
C          FUNCTION FMU
C          ------------
C
C*****************************************************************
      REAL*8 FUNCTION FMU(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C
         IF (IEQ .EQ. 1) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 2) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 3) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 4) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 5) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 6) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 7) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 8) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 9) THEN
         FMU = 0.0
         ELSEIF (IEQ .EQ. 10) THEN
```

```
          FMU = 0.0
          ENDIF
          RETURN
          END
C*******************************************************************
C
C            FUNCTION FRHO1
C           -------------
C
C*******************************************************************
      REAL*8 FUNCTION FRHO1(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C
          IF (IEQ .EQ. 1) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 2) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 3) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 4) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 5) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 6) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 7) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 8) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 9) THEN
          FRHO1 = 0.0
          ELSEIF (IEQ .EQ. 10) THEN
          FRHO1 = 0.0
          ENDIF
          RETURN
          END
C*******************************************************************
C
C            FUNCTION FRHO2
C           -------------
C
C*******************************************************************
      REAL*8 FUNCTION FRHO2(IEQ,UELEM,NEL,MAT)

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UELEM(10,1)
C
      INCLUDE 'THVAR.H'
C
          IF (IEQ .EQ. 1) THEN
          FRHO2 = 0.0
          ELSEIF (IEQ .EQ. 2) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 3) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 4) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 5) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 6) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 7) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 8) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 9) THEN
      FRHO2 = 0.0
      ELSEIF (IEQ .EQ. 10) THEN
      FRHO2 = 0.0
      ENDIF
      RETURN
      END

C*********************************************************************
C       SUBROUTINE CTBULK
C       ------------
C  CALCULATES AVG U (W AND TBULK), RE, NU, CF
C*********************************************************************
      SUBROUTINE CTBULK(IMAT,NE,MAT,NODES,X,U,TIME,WAREA,WNODES,
  .  WELEM,AMATA,WBAR,UELEM,SIGMA,PROP)

   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
   COMMON/CINT/XIQ(9,2,3),WQ(9,3)
   COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
   COMMON/FILENAMES/INFILE,JTITLE
   COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
  . ,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
   COMMON/RM_UMAX/IRM,RM_CAL,RMKAY,UMAX,CKARMANI
C
   INCLUDE 'THVAR.H'
C
   DIMENSION UELEM(10,1),PSI(9),DPSI(9,2)
      DIMENSION NE(1),MAT(1),NODES(9,1),X(2,1),U(10,1)
      DIMENSION WAREA(10,1),WNODES(1),WELEM(10)
      DIMENSION AMATA(10,1),WBAR(10,45)
      DIMENSION ANUSELT(45),IMAT(10,10,45)
   DIMENSION PROP(10,10,1),TVAR(10,10,1,20),VAR(10,10,1,20)
   DIMENSION  SIGMA(1)
      REAL*8 LINELEN,QUADAREA
      REAL*8 PX(9),PY(9)
      REAL*8 A,B,C,D,P,Q
   DATA PI,PI2  /3.141592654,1.570796327/
   CHARACTER*20 INFILE

      DO 10 IM=1,NMAT
      DO 11 IEQ=1,NPDE
       AMATA(IEQ,IM) = 0.0
       WAREA(IEQ,IM) = 0.0
11    CONTINUE
      TBULK(IM) = 0.0
```

```
10        CONTINUE
C..

     SFLOW1 = 0.0
     SFLOW2 = 0.0
     SAREA1 = 0.0
     SAREA2 = 0.0
C..
          DO 20 IE=1,NELEM
          DO 21 IEQ=1,NPDE
           MATNO = MAT(IE)
C
C... OBTAIN ELEMENT VALUE BASED AT SINGLE GAUSS POINT (L=1)
C
     NEL=IE
     N=NE(NEL)
     L=1
     NN=1
     IF(N.EQ.4) GO TO 15
     IF(N.EQ.8) GO TO 25
     IF(N.EQ.9) GO TO 35
C
15    CALL SHAPE4 (XIQ(L,1,NN),XIQ(L,2,NN),N,PSI,DPSI)

     GO TO 66

25    CALL SHAPE8 (XIQ(L,1,NN),XIQ(L,2,NN),N,PSI,DPSI,NODES(1,NEL))

     GO TO 66

35    CALL SHAPE9 (XIQ(L,1,NN),XIQ(L,2,NN),N,PSI,DPSI)

66    UH=0.
     DO 17 I=1,N
17    UH=UH+PSI(I)*U(IEQ,NODES(I,NEL))
C
           WELEM(IEQ) = UH
C
           DO 30 IN=1,NE(IE)

            PX(IN) = X(1,NODES(IN,IE))
            PY(IN) = X(2,NODES(IN,IE))
30         CONTINUE
            A = LINELEN( PX(1),PY(1),PX(2),PY(2) )
            B = LINELEN( PX(2),PY(2),PX(3),PY(3) )
            C = LINELEN( PX(3),PY(3),PX(4),PY(4) )
            D = LINELEN( PX(4),PY(4),PX(1),PY(1) )
            P = LINELEN( PX(2),PY(2),PX(4),PY(4) )
            Q = LINELEN( PX(1),PY(1),PX(3),PY(3) )
            ELAREA = QUADAREA(A,B,C,D,P,Q)
            WAREA(IEQ,MATNO) = WAREA(IEQ,MATNO) + ELAREA*WELEM(IEQ)
            AMATA(IEQ,MATNO) = AMATA(IEQ,MATNO) + ELAREA
C..
C.. CALCULATE SUBCHANNEL MASS FLOWS
C..
       IF(IEQ .EQ. 1 .AND. MATNO .EQ. 2)THEN
        DO 177 IGO = 0,29
         IF(IE.GE.(21+IGO*48).AND.IE.LE.(48+IGO*48))THEN
```

```
      FLOW2=DEN*WELEM(1)*ELAREA
      SFLOW2=SFLOW2+FLOW2
      SAREA2=SAREA2+ELAREA
      GOTO 21
      ENDIF
177     CONTINUE
      DO 178 IGOGO = 0,29
      IF(IE.LT.(21+IGOGO*48))THEN
      FLOW1=DEN*WELEM(1)*ELAREA
      SFLOW1=SFLOW1+FLOW1
      SAREA1=SAREA1+ELAREA
      GOTO 21
      ENDIF
178     CONTINUE
     ENDIF
C..
21       CONTINUE
         TBULK(MATNO) = TBULK(MATNO) + ELAREA*WELEM(1)*WELEM(2)
20       CONTINUE
C
    TFLOW=SFLOW1+SFLOW2
    TAREA=SAREA1+SAREA2
    WRITE(3,*)'SFLOW1,SFLOW2,TFLOW,SFLOW1/TFLOW,SFLOW2/TFLOW'
    WRITE(3,*)SFLOV/1,SFLOW2,TFLOW,SFLOW1/TFLOW,SFLOW2/TFLOW
    WRITE(3,*)'SAREA1,SAREA2,TAREA,SAREA1/TAREA,SAREA2/TAREA'
    WRITE(3,*)SAREA1,SAREA2,TAREA,SAREA1/TAREA,SAREA2/TAREA
C..
    PR = VIS*CP/AK
    RRATIO = RI/RO
C..
         DO 31 IM=1,NMAT
         IF (AMATA(1,IM).NE.0.0.AND.WAREA(1,IM).NE.0.0)THEN
         WBAR(1,IM) = WAREA(1,IM)/AMATA(1,IM)
         TBULK(IM) = TBULK(IM)/WAREA(1,IM)
         WAVE = WBAR(1,2)
         TAVE = TBULK(2)
C        PRINT *,'IM,WAREA(1,IM),AMATA(1,IM),WBAR(1,IM)'
C        PRINT *,IM,WAREA(1,IM),AMATA(1,IM),WBAR(1,IM)
C********
C.. TUBE*
C********
         IF (NGEOMTYPE.EQ.0) THEN
     OPEN (1,FILE=INFILE(1:JTITLE)//'.da1',STATUS='UNKNOWN')
     RE = DEN*DH*WBAR(1,IM)/VIS
         CF = -0.5*DPDZ*DH/DEN/WBAR(1,IM)**2
         CFRE = RE*CF
C..KAY'S, PG199, 3E4<RE<1E6
     CF_OTHER=2.*0.023*RE**(-0.2)
     CF_DIF=DABS(CF-CF_OTHER)/CF_OTHER*100.
         QFLUX = RAD*DEN*CP*WBAR(1,IM)*DTDZ
         HTC = QFLUX/(TW-TBULK(IM))
     FK = 1.0*AK*AMUST(NELEM)/VIS*PR/PRT(NELEM) + AK
     Q_1 = FK*(U(2,NELEM+1)-U(2,NELEM))/(X(1,NELEM+1)-X(1,NELEM))
         ANUSELT(IM) = HTC*DH/AK
C... KAY'S, PG 241-242
     WRITE(1,*)'IJ,R,Y/RO,TTEL,TEL_1,YPLUS,TPLUS,TLOG,AMUST(IJ),AKT'
C      WRITE(1,*)'IJ,REL,RRO,TEL,TEL_1,YPLUS,TPLUS,TLOG,AMUST(IJ),AKT'
         DO 19 IJ=1,NELEM
      RELEM = (X(1,IJ)+X(1,IJ+1))/2.
      RNODE = RELEM/X(1,NELEM+1)
            RRR = (RO-RELEM)/RO
      USTAR = (SIGMA(NELEM)/DEN)**0.5
      CF_1 = 2.*SIGMA(NELEM)/DEN/WBAR(1,IM)**2
      UPLUS = UEL/USTAR
      YPLUS = (RO-RELEM)*DEN*USTAR/VIS
      AKT = 1.0*AK*AMUST(IJ)/VIS*PR/PRT(IJ)
      TEL = (U(2,IJ)+U(2,IJ+1))/2.
      TPLUS=(TW-TEL)*USTAR/QFLUX*DEN*CP
      TTEL = (TEL-U(2,NELEM+1))/(U(2,1)-U(2,NELEM+1))
            IF(YPLUS.LE.13.2)THEN
       TLOG=PR*YPLUS
       TEL_1 = TW -TLOG/USTAR*QFLUX/DEN/CP
      ELSE
       TLOG=2.25*DLOG(YPLUS*1.5*(1.+RELEM/RO)/(1.+2*(RELEM/RO)**2.))
     .    +13.2*PR-5.8
       TEL_1 = TW -TLOG/USTAR*QFLUX/DEN/CP
      ENDIF
      WRITE(1,27)IJ,REL,RRR,TTEL,TEL_1,YPLUS,TPLUS,TLOG,AMUST(IJ),
     .        AKT
C      WRITE(1,27)IJ,REL,RNODE,TEL,TEL_1,YPLUS,TPLUS,TLOG,AMUST(IJ),
C    .        AKT
27         FORMAT (1X,I4,10(2X,E11.5))
19        CONTINUE
C.. SLEICHER AND ROUSE (1975), KAY'S, PG 245-247
C.. CONSTANT HEAT RATE, 1.E4<RE<1.E6, PR<1E4
      IF(RE.LT.1.E4 .OR. RE.GT.1.E6)WRITE(NLG,*)
     . 'S&R NU OUT OF RANGE'
      IF(PR.GT.0.1)THEN
       AA=0.88-0.24/(4.+PR)
       BB=0.333+0.5*DEXP(-0.6*PR)
       ANU_OTHER=5.+0.015*(RE**AA)*(PR**BB)
       ANU_DIF=DABS(ANUSELT(IM)-ANU_OTHER)/ANU_OTHER*100.
      ELSE
       ANU_OTHER=6.3+0.0167*(RE**0.85)*(PR**0.93)
       ANU_DIF=DABS(ANUSELT(IM)-ANU_OTHER)/ANU_OTHER*100.
      ENDIF
C***********
C.. ANNULUS*
C***********
          ELSEIF (NGEOMTYPE.EQ.1) THEN
      OPEN (1,FILE=INFILE(1:JTITLE)//'.dal',STATUS='UNKNOWN')
           IF(FPROP.EQ.'FIXED')THEN
       RE = DEN*DH*WBAR(1,IM)/VIS
            CF = -0.5*DPDZ*DH/DEN/WBAR(1,IM)**2
           ELSEIF(FPROP.EQ.'FIXTB')THEN
       RE = DENF(TAVE)*DH*WBAR(1,IM)/VISF(TAVE)
            CF = -0.5*DPDZ*DH/DENF(TAVE)/WBAR(1,IM)**2
           ELSE
       RE = DENF(TAVE)*DH*WBAR(1,IM)/VISF(TAVE)
            CF = -0.5*DPDZ*DH/DENF(TAVE)/WBAR(1,IM)**2
           ENDIF
           CFRE = RE*CF
C..KAY'S, PG199, 3E4<RE<1E6
      CF_OTHER=2.*0.023*RE**(-0.2)
      CF_DIF=DABS(CF-CF_OTHER)/CF_OTHER*100.
C      QT1=0.
```

```
C      QT2=0.
C         WRITE(1,*) 'I,FK,Q,QL,QT1,QT2'
C      DO 23 I=1,NELEM
C       IF (I.LT.12)THEN
C        FK = DABS(PROP(2,1,1))
C       ELSE
C        FK = 1.0*AK*AMUST(I)/VIS*PR/PRT(I) + AK
C       ENDIF
C       Q = -1.*FK*(U(2,I+1)-U(2,I))/(X(1,I+1)-X(1,I))
C       DS=((X(1,I+60+1)-X(1,I+1))**2.+(X(2,I+60+1)-X(2,I+1))**2.)**.5
C       QL=Q*DS*DZ*360./2.67
C       IF(I.LT.12)QT1=QL+QT1
C       IF(I.GE.12)QT2=QL+QT2
C          WRITE(1,266) I,FK,Q,QL,QT1,QT2
C266       FORMAT (I4,1X,11(2X,E11.5))
C23     CONTINUE
C
           IF(FPROP.EQ.'FIXED')THEN
            PR = VIS*CP/AK
      FK1 = 1.0*AK*AMUST(2)/VIS*PR/PRT(2) + AK
      FK2 = 1.0*AK*AMUST(NELEM)/VIS*PR/PRT(NELEM) + AK
           ELSEIF(FPROP.EQ.'FIXTB')THEN
            PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
      FK1 = 1.0*AKF(TAVE)*AMUST(2)/VISF(TAVE)*PR/PRT(2) + AKF(TAVE)
      FK2 = 1.0*AKF(TAVE)*AMUST(NELEM)/VISF(TAVE)*PR/PRT(NELEM)
     .     + AKF(TAVE)
           ELSE
            PR1 = VISF(UELEM(2,2))*CPF(UELEM(2,2))/AKF(UELEM(2,2))
            PR2 = VISF(UELEM(2,NELEM))*CPF(UELEM(2,NELEM))/
     .     AKF(UELEM(2,NELEM))
      FK1 = 1.0*AKF(UELEM(2,2))*AMUST(2)/VISF(UELEM(2,2))
     .     *PR1/PRT(2) + AKF(UELEM(2,2))
      FK2 = 1.0*AKF(UELEM(2,NELEM))*AMUST(NELEM)/
     .   VISF(UELEM(2,NELEM))*PR2/PRT(NELEM) + AKF(UELEM(2,NELEM))
           ENDIF
      Q_1 = -1.*FK1*(U(2,3)-U(2,2))/(X(1,3)-X(1,2))
      Q_2 = FK2*(U(2,NELEM+1)-U(2,NELEM))/
     .    (X(1,NELEM+1)-X(1,NELEM))
      QTOTAL=Q_1*2.*3.141593*RI*DZ/DZ*THETAN/360.
      IF(Q_1 .GT. Q_2)THEN
           IF(FPROP.EQ.'FIXED')THEN
            QFLUX=(RO**2.-RI**2.)/2./RI*DEN*CP*WBAR(1,IM)*DTDZ
       ANU_OTHER=Q_1/(U(2,2)-TBULK(IM))*DH/AK
           ELSEIF(FPROP.EQ.'FIXTB')THEN
            QFLUX=(RO**2.-RI**2.)/2./RI*DENF(TAVE)*CPF(TAVE)
     .     *WBAR(1,IM)*DTDZ
       ANU_OTHER=Q_1/(U(2,2)-TBULK(IM))*DH/AKF(TAVE)
           ELSE
            QFLUX=(RO**2.-RI**2.)/2./RI*DENF(TAVE)*CPF(TAVE)
     .     *WBAR(1,IM)*DTDZ
       ANU_OTHER=Q_1/(U(2,2)-TBULK(IM))*DH/AKF(TAVE)
           ENDIF
       TW=U(2,2)
      ELSE
           IF(FPROP.EQ.'FIXED')THEN
            QFLUX=(RO**2.-RI**2.)/2./RO*DEN*CP*WBAR(1,IM)*DTDZ
       ANU_OTHER=Q_2/(U(2,NELEM+1)-TBULK(IM))*DH/AK
           ELSEIF(FPROP.EQ.'FIXTB')THEN
           QFLUX=(RO**2.-RI**2.)/2./RO*DENF(TAVE)*CPF(TAVE)
     .    *WBAR(1,IM)*DTDZ
        ANU_OTHER=Q_2/(U(2,NELEM+1)-TBULK(IM))*DH/AKF(TAVE)
          ELSE
           QFLUX=(RO**2.-RI**2.)/2./RO*DENF(TAVE)*CPF(TAVE)
     .    *WBAR(1,IM)*DTDZ
        ANU_OTHER=Q_2/(U(2,NELEM+1)-TBULK(IM))*DH/AKF(TAVE)
          ENDIF
       TW=U(2,NELEM+1)
      ENDIF
          HTC = QFLUX/(TW-TBULK(IM))
          IF(FPROP.EQ.'FIXED')THEN
           ANUSELT(IM) = HTC*DH/AK
          ELSEIF(FPROP.EQ.'FIXTB')THEN
           ANUSELT(IM) = HTC*DH/AKF(TAVE)
          ELSE
           ANUSELT(IM) = HTC*DH/AKF(TAVE)
          ENDIF
C... KAY'S, PG 241-242
C       WRITE(1,*)'IJ,RNODE,TEL,TEL_1,YPLUS,TPLUS,TLOG,AMUST(IJ),AKT
          DO 18 IJ=2,NELEM
      RELEM = (X(1,IJ)+X(1,IJ+1))/2.
      RNODE = RELEM/X(1,NELEM+1)
           IF(FPROP.EQ.'FIXED')THEN
            PR = VIS*CP/AK
        USTAR = (SIGMA(NELEM)/DEN)**0.5
        YPLUS = (RO-RELEM)*DEN*USTAR/VIS
        AKT = 1.0*AK*AMUST(IJ)/VIS*PR/PRT(IJ)
        TEL = (U(2,IJ)+U(2,IJ+1))/2.
        TPLUS=(TW-TEL)*USTAR/QFLUX*DEN*CP
           ELSEIF(FPROP.EQ.'FIXTB')THEN
            PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
        USTAR = (SIGMA(NELEM)/DENF(TAVE))**0.5
        YPLUS = (RO-RELEM)*DENF(TAVE)*USTAR/VISF(TAVE)
        AKT = 1.0*AKF(TAVE)*AMUST(IJ)/VISF(TAVE)*PR/PRT(IJ)
        TEL = (U(2,IJ)+U(2,IJ+1))/2.
        TPLUS=(TW-TEL)*USTAR/QFLUX*DENF(TAVE)*CPF(TAVE)
           ELSE
            PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
        USTAR = (SIGMA(NELEM)/DENF(UELEM(2,NEL)))**0.5
        YPLUS = (RO-RELEM)*DENF(UELEM(2,NEL))*USTAR/
     .     VISF(UELEM(2,NEL))
        AKT = 1.0*AKF(UELEM(2,NEL))*AMUST(IJ)/VISF(UELEM(2,NEL))
     .    *PR/PRT(IJ)
        TEL = (U(2,IJ)+U(2,IJ+1))/2.
        TPLUS=(TW-TEL)*USTAR/QFLUX*DENF(UELEM(2,NEL))
     .    *CPF(UELEM(2,NEL))
           ENDIF
      IF(YPLUS.LE.13.2)THEN
        TLOG=PR*YPLUS
            IF(FPROP.EQ.'FIXED')THEN
         TEL_1 = TW -TLOG/USTAR*QFLUX/DEN/CP
            ELSEIF(FPROP.EQ.'FIXTB')THEN
         TEL_1 = TW -TLOG/USTAR*QFLUX/DENF(TAVE)/CPF(TAVE)
            ELSE
         TEL_1 = TW -TLOG/USTAR*QFLUX/DENF(UELEM(2,NEL))
     .      /CPF(UELEM(2,NEL))
            ENDIF
      ELSE
       TLOG=2.25*DLOG(YPLUS*1.5*(1.+RELEM/RO)/(1.+2*(RELEM/RO)**2.))
     .     +13.2*PR-5.8
           IF(FPROP.EQ.'FIXED')THEN
       TEL_1 = TW -TLOG/USTAR*QFLUX/DEN/CP
           ELSEIF(FPROP.EQ.'FIXTB')THEN
       TEL_1 = TW -TLOG/USTAR*QFLUX/DENF(TAVE)/CPF(TAVE)
           ELSE
       TEL_1 = TW -TLOG/USTAR*QFLUX/DENF(UELEM(2,NEL))
     .     /CPF(UELEM(2,NEL))
           ENDIF
      ENDIF
C      WRITE(1,27)IJ,RNODE,TEL,TEL_1,YPLUS,TPLUS,TLOG,AMUST(IJ),AKT
18    CONTINUE
C******************************
C.. PATANKAR'S FINNED ANNULUS*
C******************************
         ELSEIF (NGEOMTYPE.EQ.11) THEN
      RE = DEN*DH*WBAR(1,IM)/VIS
         CF = -0.5*DPDZ*DH/DEN/WBAR(1,IM)**2.
         CFRE = RE*CF
      QLN = DEN*WBAR(1,IM)*AFLOW*CP*DTDZ
C
C... SHEATH
C
     QTOTAL=0.
         TWTOTAL=0.
         HTCTOTAL=0.
         SAREAT=0.
     IEL=0
C
     ELINC=50
     NODINC=51
C
     WRITE(3,259)
     WRITE(3,*) 'SAREAT,QFLXL,TWL,HTCL,-XK'
         DO 56 I=1021,52,-NODINC
         R1P = DSQRT(X(1,I+1)**2.+X(2,I+1)**2.)
         R1 = DSQRT(X(1,I)**2.+X(2,I)**2.)
     DS1P = R1P-R1
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
     IF(FLOWTYPE.EQ.'TURBULENT')THEN
      IEQ=2
      NEL=951-ELINC*IEL
      IEL=IEL+1
      CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),
    >          PROP,UELEM,IMAT,VAR,TVAR,NEL,IEQ)
      FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
      ELSE
      XK=AK
     ENDIF
         IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+1)-U(2,I))/DS1P
     QFLXL1=QFLXL1P
         R2P = DSQRT(X(1,I-NODINC+1)**2.+X(2,I-NODINC+1)**2.)
         R2 = DSQRT(X(1,I-NODINC)**2.+X(2,I-NODINC)**2.)
     DS2P = R2P-R2
         IF(DS2P.NE.0.)QFLXL2P=-1.*DABS(XK)*(U(2,I-NODINC+1)
     .            -U(2,I-NODINC))/DS2P
```

```
      QFLXL2=QFLXL2P
          SAREA = DSQRT((X(1,I)-X(1,I-NODINC))**2.+(X(2,I)
    .     -X(2,I-NODINC))**2.)
      QFLXL = (QFLXL1+QFLXL2)/2.
          TWL = (U(2,I)+U(2,I-NODINC))/2.
          TWTOTAL = TWTOTAL + TWL*SAREA
          HTCL = QFLXL/(TWL-TAVE)
          HTCTOTAL = HTCTOTAL + HTCL*SAREA
      QLOCAL = QFLXL*SAREA
          QTOTAL = QTOTAL + QLOCAL
      SAREAT = SAREAT + SAREA
          WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
56        CONTINUE
      QTOTAL1 = QTOTAL
      SAREAT1 = SAREAT
C
C... FIN SIDE
C
      IEL=0
          DO 57 I=1,20
          DS1P = X(2,I+NODINC)-X(2,I)
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
      IF(FLOWTYPE.EQ.'TURBULENT')THEN
       IEQ=2
       INEL=1+IEL
       IEL=IEL+1
       CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(INEL),
    >        PROP,UELEM,IMAT,VAR,TVAR,INEL,IEQ)
       FK11 = -1.0*AK*AMUST(INEL)/VIS*PR/PRT(INEL) - AK
       ELSE
       XK=AK
      ENDIF
          IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+NODINC)-U(2,I))/DS1P
      QFLXL1=QFLXL1P
          DS2P = X(2,I+NODINC+1)-X(2,I+1)
          IF(DS2P.NE.0.)QFLXL2P = -1.*DABS(XK)*(U(2,I+NODINC+1)
    .              -U(2,I+1))/DS2P
      QFLXL2=QFLXL2P
          SAREA = X(1,I+1)-X(1,I)
      QFLXL = (QFLXL1+QFLXL2)/2.
          TWL = (U(2,I)+U(2,I+1))/2.
          TWTOTAL = TWTOTAL + TWL*SAREA
          HTCL = QFLXL/(TWL-TAVE)
          HTCTOTAL = HTCTOTAL + HTCL*SAREA
      QLOCAL = QFLXL*SAREA
          QTOTAL = QTOTAL + QLOCAL
      SAREAT = SAREAT + SAREA
          WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
57        CONTINUE
      QTOTAL2 = QTOTAL - QTOTAL1
      SAREAT2 = SAREAT - SAREAT1
      QFLUX = QTOTAL/SAREAT
          TW = TWTOTAL/SAREAT
C       HTC = HTCTOTAL/SAREAT
      HTC = QFLUX/(TW-TAVE)
          ANUSELT(IM) = HTC*DH/AK
C
          Q_1 = AFLOW*THETAN/360./SAREAT*DEN*CP*WAVE*DTDZ
      ANU_OTHER = Q_1/(TW-TAVE)*DH/AK
C********************************
C.. PATANKAR'S UNFINNED ANNULUS*
C********************************
          ELSEIF (NGEOMTYPE.EQ.12) THEN
       RE = DEN*DH*WBAR(1,IM)/VIS
           CF = -0.5*DPDZ*DH/DEN/WBAR(1,IM)**2.
           CFRE = RE*CF
       QLN = DEN*WBAR(1,IM)*AFLOW*CP*DTDZ
C
C... SHEATH
C
      QTOTAL=0.
          TWTOTAL=0.
          HTCTOTAL=0.
          SAREAT=0.
      IEL=0
C
      ELINC=50
      NODINC=51
C
      WRITE(3,259)
      WRITE(3,*) 'SAREAT,QFLXL,TWL,HTCL,-XK'
          DO 556 I=1021,52,-NODINC
          R1P = DSQRT(X(1,I+1)**2.+X(2,I+1)**2.)
          R1 = DSQRT(X(1,I)**2.+X(2,I)**2.)
      DS1P = R1P-R1
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
      IF(FLOWTYPE.EQ.'TURBULENT')THEN
       IEQ=2
       NEL=951-ELINC*IEL
       IEL=IEL+1
       CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),
     >          PROP,UELEM,IMAT,VAR,TVAR,NEL,IEQ)
       FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
       ELSE
       XK=AK
      ENDIF
          IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+1)-U(2,I))/DS1P
      QFLXL1=QFLXL1P
          R2P = DSQRT(X(1,I-NODINC+1)**2.+X(2,I-NODINC+1)**2.)
          R2 = DSQRT(X(1,I-NODINC)**2.+X(2,I-NODINC)**2.)
      DS2P = R2P-R2
          IF(DS2P.NE.0.)QFLXL2P=-1.*DABS(XK)*(U(2,I-NODINC+1)
     .              -U(2,I-NODINC))/DS2P
      QFLXL2=QFLXL2P
          SAREA = DSQRT((X(1,I)-X(1,I-NODINC))**2.+(X(2,I)
     .    -X(2,I-NODINC))**2.)
      QFLXL = (QFLXL1+QFLXL2)/2.
          TWL = (U(2,I)+U(2,I-NODINC))/2.
          TWTOTAL = TWTOTAL + TWL*SAREA
          HTCL = QFLXL/(TWL-TAVE)
          HTCTOTAL = HTCTOTAL + HTCL*SAREA
      QLOCAL = QFLXL*SAREA
          QTOTAL = QTOTAL + QLOCAL
      SAREAT = SAREAT + SAREA
          WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
556       CONTINUE
```

```
      QTOTAL1 = QTOTAL
      SAREAT1 = SAREAT
C
      QFLUX = QTOTAL/SAREAT
          TW = TWTOTAL/SAREAT
C       HTC = HTCTOTAL/SAREAT
      HTC = QFLUX/(TW-TAVE)
          ANUSELT(IM) = HTC*DH/AK
C
          Q_1 = AFLOW*THETAN/360./SAREAT*DEN*CP*WAVE*DTDZ
      ANU_OTHER = Q_1/(TW-TAVE)*DH/AK
C*****************
C.. FA8 UNFINNED*
C*****************
          ELSEIF (NGEOMTYPE.EQ.22) THEN
           DHNEW=2.*(RO-RI)
           IF(FPROP.EQ.'FIXED')THEN
       RE = DEN*DHNEW*WBAR(1,IM)/VIS
           CF = -0.5*DPDZ*DHNEW/DEN/WBAR(1,IM)**2.
           CFRE = RE*CF
       QLN = DEN*WBAR(1,IM)*AFLOW*CP*DTDZ
           ELSEIF(FPROP.EQ.'FIXTB')THEN
       RE = DENF(TAVE)*DHNEW*WBAR(1,IM)/VISF(TAVE)
           CF = -0.5*DPDZ*DHNEW/DENF(TAVE)/WBAR(1,IM)**2.
           CFRE = RE*CF
       QLN = DENF(TAVE)*WBAR(1,IM)*AFLOW*CPF(TAVE)*DTDZ
           ELSE
       RE = DENF(TAVE)*DHNEW*WBAR(1,IM)/VISF(TAVE)
           CF = -0.5*DPDZ*DHNEW/DENF(TAVE)/WBAR(1,IM)**2.
           CFRE = RE*CF
       QLN = DENF(TAVE)*WBAR(1,IM)*AFLOW*CPF(TAVE)*DTDZ
           ENDIF
C
C... SHEATH
C
      QTOTAL=0.
          TWTOTAL=0.
          HTCTOTAL=0.
          SAREAT=0.
      IEL=0
C
      ELINC=48
      NODINC=49
C
      WRITE(3,259)
      WRITE(3,*) 'SAREAT,QFLXL,TWL,HTCL,-XK'
          DO 566 I=1474,53,-NODINC
          R1P = DSQRT(X(1,I+1)**2.+X(2,I+1)**2.)
          R1 = DSQRT(X(1,I)**2.+X(2,I)**2.)
      DS1P = R1P-R1
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
      IF(FLOWTYPE.EQ.'TURBULENT')THEN
       IEQ=2
       NEL=1396-ELINC*IEL
       IEL=IEL+1
       CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),
   >         PROP,UELEM,IMAT,VAR,TVAR,NEL,IEQ)
           IF(FPROP.EQ.'FIXED')THEN
          PR = VIS*CP/AK
      FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
          ELSEIF(FPROP.EQ.'FIXTB')THEN
          PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
      FK11 = -1.0*AKF(TAVE)*AMUST(NEL)/VISF(TAVE)*PR/PRT(NEL)
     .      - AKF(TAVE)
          ELSE
          PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
      FK11 = -1.0*AKF(UELEM(2,NEL))*AMUST(NEL)/VISF(UELEM(2,NEL))
     .      *PR/PRT(NEL) - AKF(UELEM(2,NEL))
          ENDIF
     ELSE
          IF(FPROP.EQ.'FIXED')THEN
      XK=AK
          ELSEIF(FPROP.EQ.'FIXTB')THEN
      XK=AKF(TAVE)
          ELSE
      XK=AKF(UELEM(2,NEL))
          ENDIF
    ENDIF
        IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+1)-U(2,I))/DS1P
    QFLXL1=QFLXL1P
        R2P = DSQRT(X(1,I-NODINC+1)**2.+X(2,I-NODINC+1)**2.)
        R2 = DSQRT(X(1,I-NODINC)**2.+X(2,I-NODINC)**2.)
    DS2P = R2P-R2
        IF(DS2P.NE.0.)QFLXL2P=-1.*DABS(XK)*(U(2,I-NODINC+1)
     .            -U(2,I-NODINC))/DS2P
    QFLXL2=QFLXL2P
        SAREA = DSQRT((X(1,I)-X(1,I-NODINC))**2.+(X(2,I)
     .  -X(2,I-NODINC))**2.)
    QFLXL = (QFLXL1+QFLXL2)/2.
        TWL = (U(2,I)+U(2,I-NODINC))/2.
        TWTOTAL = TWTOTAL + TWL*SAREA
        HTCL = QFLXL/(TWL-TAVE)
        HTCTOTAL = HTCTOTAL + HTCL*SAREA
    QLOCAL = QFLXL*SAREA
        QTOTAL = QTOTAL + QLOCAL
    SAREAT = SAREAT + SAREA
        WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
566     CONTINUE
    QTOTAL1 = QTOTAL
    SAREAT1 = SAREAT
C
    QFLUX = QTOTAL/SAREAT
        TW = TWTOTAL/SAREAT
C     HTC = HTCTOTAL/SAREAT
    HTC = QFLUX/(TW-TAVE)
        IF(FPROP.EQ.'FIXED')THEN
         ANUSELT(IM) = HTC*DHNEW/AK
         Q_1 = AFLOW*THETAN/360./SAREAT*DEN*CP*WAVE*DTDZ
     ANU_OTHER = Q_1/(TW-TAVE)*DHNEW/AK
        ELSEIF(FPROP.EQ.'FIXTB')THEN
         ANUSELT(IM) = HTC*DHNEW/AKF(TAVE)
         Q_1 = AFLOW*THETAN/360./SAREAT*DENF(TAVE)*CPF(TAVE)*WAVE*DTDZ
     ANU_OTHER = Q_1/(TW-TAVE)*DHNEW/AKF(TAVE)
        ELSE
         ANUSELT(IM) = HTC*DHNEW/AKF(TAVE)
         Q_1 = AFLOW*THETAN/360./SAREAT*DENF(TAVE)*CPF(TAVE)*WAVE*DTDZ
      ANU_OTHER = Q_1/(TW-TAVE)*DHNEW/AKF(TAVE)
          ENDIF
C*************************
C.. FINNED TUBE (SOLIMAN)*
C*************************
         ELSEIF (NGEOMTYPE.EQ.2 .OR. NGEOMTYPE.EQ.3)THEN
      RE = DEN*DH*WBAR(1,IM)/VIS
          CF = -0.5*DPDZ*DH/DEN/WBAR(1,IM)**2.
          CFRE = RE*CF
C.. SOLIMAN'S PAPER
      CF_OTHER = -2.*PI*DPDZ*RO**4./VIS/WBAR(1,IM)/AFLOW/RE
      CF_DIF=DABS(CF-CF_OTHER)/CF_OTHER*100.
      CFRE_OTHER = -2.*PI*DPDZ*RO**4./VIS/WBAR(1,IM)/AFLOW
          QLN = DEN*WBAR(1,IM)*AFLOW*CP*DTDZ
          QFLUX = QLN/(2*PI*RO)
          HTC = QFLUX/(TW-TBULK(IM))
          ANUSELT(IM) = HTC*DH/AK
          ANU_OTHER = HTC*2.*RO/AK
      ANU_DIF=DABS(ANUSELT(IM)-ANU_OTHER)/ANU_OTHER*100.
C*******
C.. FA8*
C*******
      ELSEIF (NGEOMTYPE.EQ.21) THEN
          DHNEW=2.*(RO-RI)
          IF(FPROP.EQ.'FIXED')THEN
      RE = DEN*DHNEW*WBAR(1,IM)/VIS
          CF = -0.5*DPDZ*DHNEW/DEN/WBAR(1,IM)**2.
          CFRE = RE*CF
      QLN = DEN*WBAR(1,IM)*AFLOW*CP*DTDZ
          ELSEIF (FPROP.EQ.'FIXTB')THEN
      RE = DENF(TAVE)*DHNEW*WBAR(1,IM)/VISF(TAVE)
          CF = -0.5*DPDZ*DHNEW/DENF(TAVE)/WBAR(1,IM)**2.
          CFRE = RE*CF
      QLN = DENF(TAVE)*WBAR(1,IM)*AFLOW*CPF(TAVE)*DTDZ
          ELSE
      RE = DENF(TAVE)*DHNEW*WBAR(1,IM)/VISF(TAVE)
          CF = -0.5*DPDZ*DHNEW/DENF(TAVE)/WBAR(1,IM)**2.
          CFRE = RE*CF
      QLN = DENF(TAVE)*WBAR(1,IM)*AFLOW*CPF(TAVE)*DTDZ
          ENDIF
C
C...  SHEATH
C
     QTOTAL=0.
         TWTOTAL=0.
         HTCTOTAL=0.
         SAREAT=0.
     IEL=0
     WRITE(3,259)
     WRITE(3,*) 'SAREAT,QFLXL,TWL,HTCL,-XK'
         DO 39 I=1474,347,-49
         R1P = DSQRT(X(1,I+1)**2.+X(2,I+1)**2.)
         R1 = DSQRT(X(1,I)**2.+X(2,I)**2.)
         R1M = DSQRT(X(1,I-1)**2.+X(2,I-1)**2.)
     DS1P = R1P-R1
     DS1M = R1-R1M
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
     IF(FLOWTYPE.EQ.'TURBULENT')THEN
```

```
      IEQ=2
      NEL=1396-48*IEL
      IEL=IEL+1
      CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),
     >       PROP,UELEM,IMAT,VAR,TVAR,NEL,IEQ)
          IF(FPROP.EQ.'FIXED')THEN
       PR = VIS*CP/AK
       FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
          ELSEIF (FPROP.EQ.'FIXTB')THEN
       PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
       FK11 = -1.0*AKF(TAVE)*AMUST(NEL)/VISF(TAVE)*
     .   PR/PRT(NEL) - AKF(TAVE)
          ELSE
       PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
       FK11 = -1.0*AKF(UELEM(2,NEL))*AMUST(NEL)/VISF(UELEM(2,NEL))*
     .   PR/PRT(NEL) - AKF(UELEM(2,NEL))
          ENDIF
      ELSE
          IF(FPROP.EQ.'FIXED')THEN
       XK=AK
          ELSEIF (FPROP.EQ.'FIXTB')THEN
       XK=AKF(TAVE)
          ELSE
       XK=AKF(UELEM(2,NEL))
          ENDIF
      ENDIF
          IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+1)-U(2,I))/DS1P
          IF(DS1M.NE.0.)QFLXL1M=-1.*DABS(PROP(2,1,3))*(U(2,I)-U(2,I-1))
     .        /DS1M
      QFLXL1=QFLXL1P
          R2P = DSQRT(X(1,I-49+1)**2.+X(2,I-49+1)**2.)
          R2 = DSQRT(X(1,I-49)**2.+X(2,I-49)**2.)
          R2M = DSQRT(X(1,I-49-1)**2.+X(2,I-49-1)**2.)
      DS2P = R2P-R2
      DS2M = R2-R2M
          IF(DS2P.NE.0.)QFLXL2P=-1.*DABS(XK)*(U(2,I-49+1)-U(2,I-49))/DS2P
          IF(DS2M.NE.0.)QFLXL2M=-1.*DABS(PROP(2,1,3))*(U(2,I-49)
     .        -U(2,I-49-1))/DS2M
      QFLXL2=QFLXL2P
          SAREA = DSQRT((X(1,I)-X(1,I-49))**2.+(X(2,I)-X(2,I-49))**2.)
      QFLXL = (QFLXL1+QFLXL2)/2.
          TWL = (U(2,I)+U(2,I-49))/2.
          TWTOTAL = TWTOTAL + TWL*SAREA
          HTCL = QFLXL/(TWL-TAVE)
          HTCTOTAL = HTCTOTAL + HTCL*SAREA
      QLOCAL = QFLXL*SAREA
          QTOTAL = QTOTAL + QLOCAL
      SAREAT = SAREAT + SAREA
          WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
39        CONTINUE
      QTOTAL1 = QTOTAL
      SAREAT1 = SAREAT
C
C... FIN SIDE
C
      IEL=0
          DO 41 I=298,314
          DS1P = X(2,I+49)-X(2,I)
          DS1M = X(2,I)-X(2,I-49)
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
      IF(FLOWTYPE.EQ.'TURBULENT')THEN
      IEQ=2
      NEL=292+IEL
      IEL=IEL+1
      CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),
     >          PROP,UELEM,IMAT,VAR,TVAR,NEL,IEQ)
          IF(FPROP.EQ.'FIXED')THEN
       PR = VIS*CP/AK
       FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
          ELSEIF (FPROP.EQ.'FIXTB')THEN
       PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
       FK11 = -1.0*AKF(TAVE)*AMUST(NEL)/VISF(TAVE)*
     .     PR/PRT(NEL) - AKF(TAVE)
          ELSE
       PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
       FK11 = -1.0*AKF(UELEM(2,NEL))*AMUST(NEL)/VISF(UELEM(2,NEL))*
     .     PR/PRT(NEL) - AKF(UELEM(2,NEL))
          ENDIF
      ELSE
          IF(FPROP.EQ.'FIXED')THEN
       XK=AK
          ELSEIF (FPROP.EQ.'FIXTB')THEN
       XK=AKF(TAVE)
          ELSE
       XK=AKF(UELEM(2,NEL))
          ENDIF
      ENDIF
          IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+49)-U(2,I))/DS1P
          IF(DS1M.NE.0.)QFLXL1M=-1.*DABS(PROP(2,1,3))*(U(2,I)-U(2,I-49))
     .        /DS1M
      QFLXL1=QFLXL1P
          DS2P = X(2,I+49+1)-X(2,I+1)
          DS2M = X(2,I+1)-X(2,I-49+1)
          IF(DS2P.NE.0.)QFLXL2P = -1.*DABS(XK)*(U(2,I+49+1)-U(2,I+1))/DS2P
          IF(DS2M.NE.0.)QFLXL2M=-1.*DABS(PROP(2,1,3))*(U(2,I+1)
     .             -U(2,I-49+1))/DS2M
      QFLXL2=QFLXL2P
          SAREA = X(1,I+1)-X(1,I)
      QFLXL = (QFLXL1+QFLXL2)/2.
          TWL = (U(2,I)+U(2,I+1))/2.
          TWTOTAL = TWTOTAL + TWL*SAREA
          HTCL = QFLXL/(TWL-TAVE)
          HTCTOTAL = HTCTOTAL + HTCL*SAREA
      QLOCAL = QFLXL*SAREA
          QTOTAL = QTOTAL + QLOCAL
      SAREAT = SAREAT + SAREA
          WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
41        CONTINUE
      QTOTAL2 = QTOTAL - QTOTAL1
      SAREAT2 = SAREAT - SAREAT1
C
C... FIN TIP
C
      IEL=0
          DO 43 I=315,70,-49
          DS1P = X(1,I+1)-X(1,I)
          DS1M = X(1,I)-X(1,I-1)
C... ONLY FOR ENERGY EQUATION, I.E. IEQ=2
       IF(FLOWTYPE.EQ.'TURBULENT')THEN
       IEQ=2
       NEL=261-IEL*48
       IEL=IEL+1
       CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),
     >        PROP,UELEM,IMAT,VAR,TVAR,NEL,IEQ)
          IF(FPROP.EQ.'FIXED')THEN
       PR = VIS*CP/AK
       FK11 = -1.0*AK*AMUST(NEL)/VIS*PR/PRT(NEL) - AK
          ELSEIF (FPROP.EQ.'FIXTB')THEN
       PR = VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
       FK11 = -1.0*AKF(TAVE)*AMUST(NEL)/VISF(TAVE)*
     .      PR/PRT(NEL) - AKF(TAVE)
          ELSE
       PR = VISF(UELEM(2,NEL))*CPF(UELEM(2,NEL))/AKF(UELEM(2,NEL))
       FK11 = -1.0*AKF(UELEM(2,NEL))*AMUST(NEL)/VISF(UELEM(2,NEL))*
     .      PR/PRT(NEL) - AKF(UELEM(2,NEL))
          ENDIF
      ELSE
          IF(FPROP.EQ.'FIXED')THEN
       XK=AK
          ELSEIF (FPROP.EQ.'FIXTB')THEN
       XK=AKF(TAVE)
          ELSE
       XK=AKF(UELEM(2,NEL))
          ENDIF
      ENDIF
          IF(DS1P.NE.0.)QFLXL1P = -1.*DABS(XK)*(U(2,I+1)-U(2,I))/DS1P
          IF(DS1M.NE.0.)QFLXL1M=-1.*DABS(PROP(2,1,3))*(U(2,I+1)-U(2,I))
     .             /DS1M
      QFLXL1=QFLXL1P
          DS2P = X(1,I-49+1)-X(1,I-49)
          DS2M = X(1,I-49)-X(1,I-49-1)
          IF(DS2P.NE.0.)QFLXL2P = -1.*DABS(XK)*(U(2,I-49+1)-U(2,I-49))
     .             /DS2P
          IF(DS2M.NE.0.)QFLXL2M = -1.*DABS(PROP(2,1,3))*(U(2,I-49)
     .             -U(2,I-49-1))/DS2M
      QFLXL2=QFLXL2P
          SAREA = X(2,I)-X(2,I-49)
      QFLXL = (QFLXL1+QFLXL2)/2.
          TWL = (U(2,I)+U(2,I-49))/2.
          TWTOTAL = TWTOTAL + TWL*SAREA
          HTCL = QFLXL/(TWL-TAVE)
          HTCTOTAL = HTCTOTAL + HTCL*SAREA
      QLOCAL = QFLXL*SAREA
          QTOTAL = QTOTAL + QLOCAL
      SAREAT = SAREAT + SAREA
          WRITE(3,26) SAREAT,QFLXL,TWL,HTCL,-XK
43        CONTINUE
          QTOTAL3 = QTOTAL - QTOTAL1 - QTOTAL2
      SAREAT3 = SAREAT - SAREAT1 - SAREAT2
C
      QFLUX = QTOTAL/SAREAT
          TW = TWTOTAL/SAREAT
C      HTC = HTCTOTAL/SAREAT
      HTC = QFLUX/(TW-TAVE)
```

```
           IF(FPROP.EQ.'FIXED')THEN
            ANUSELT(IM) = HTC*DHNEW/AK
            Q_1 = AFLOW*THETAN/360./SAREAT*DEN*CP*WAVE*DTDZ
      ANU_OTHER = Q_1/(TW-TAVE)*DHNEW/AK
           ELSEIF (FPROP.EQ.'FIXTB')THEN
            ANUSELT(IM) = HTC*DHNEW/AKF(TAVE)
            Q_1 = AFLOW*THETAN/360./SAREAT*DENF(TAVE)*CPF(TAVE)*WAVE*DTDZ
      ANU_OTHER = Q_1/(TW-TAVE)*DHNEW/AKF(TAVE)
           ELSE
            ANUSELT(IM) = HTC*DHNEW/AKF(TAVE)
            Q_1 = AFLOW*THETAN/360./SAREAT*DENF(TAVE)*CPF(TAVE)*WAVE*DTDZ
      ANU_OTHER = Q_1/(TW-TAVE)*DHNEW/AKF(TAVE)
           ENDIF
          ELSE
          ENDIF
C
C.. TOTAL HEAT GENERATED (W)
C.. PROP(EQN#, INDEX # FOR FFF, MATERIAL#), AMATA(EQN#, MAT#)
C
      HEAT1=DABS(PROP(2,7,1))*AMATA(2,1)*DZ*360./THETAN
C
C.. HEAT TO FLUID
C
           IF(FPROP.EQ.'FIXED')THEN
      HEAT2=DEN*CP*WAVE*DTDZ*AMATA(2,2)*DZ*360./THETAN
           ELSEIF (FPROP.EQ.'FIXTB')THEN
      HEAT2=DENF(TAVE)*CPF(TAVE)*WAVE*DTDZ*AMATA(2,2)*DZ*360./THETAN
           ELSE
      HEAT2=DENF(TAVE)*CPF(TAVE)*WAVE*DTDZ*AMATA(2,2)*DZ*360./THETAN
           ENDIF
C
C.. HEAT LEAVING THE SURFACE
C
      HEAT3=QTOTAL*DZ*360./THETAN
C
           WRITE(NLG,98) IM,WBAR(1,IM),CF,CF_1,CF_OTHER,CF_DIF,RE,CFRE,
   .          CFRE_OTHER
           WRITE(NLG,99)IM,HTC,TW,TBULK(IM),QFLUX,Q_1,Q_2,ANUSELT(IM),
   .         ANU_OTHER,ANU_DIF
           WRITE(NLG,101)HEAT1,HEAT2,HEAT3
           WRITE(NLG,100)PR,RRATIO,DPDZ,IRM,RM_CAL,RMKAY,UMAX,CKARMANI

      IF (NGEOMTYPE.EQ.1.OR.NGEOMTYPE.EQ.21.OR.NGEOMTYPE.EQ.22) THEN
      OPEN (35,FILE=INFILE(1:JTITLE)//'.da3',STATUS='UNKNOWN')
            IF(NPDE.EQ.1)TAVE=TIN
            IF(FPROP.EQ.'FIXED')THEN
        PRAVE=VIS*CP/AK
        RMDOT=DEN*WAVE*AMATA(1,2)*360./THETAN
             WRITE(NLG,*)'AVG.DEN,VIS,CP,AK,PR,T,W,RE,M FLOW,-DPDZ'
             WRITE(NLG,26)DEN,VIS,CP,AK,PRAVE,TAVE,WAVE,RE,RMDOT,-DPDZ
             WRITE(35,*)'AVG.DEN,VIS,CP,AK,PR,T,W,RE,M FLOW,-DPDZ'
             WRITE(35,26)DEN,VIS,CP,AK,PRAVE,TAVE,WAVE,RE,RMDOT,-DPDZ
            ELSEIF (FPROP.EQ.'FIXTB')THEN
        PRAVE=VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
        RMDOT=DENF(TAVE)*WAVE*AMATA(1,2)*360./THETAN
             WRITE(NLG,*)'AVG.DEN,VIS,CP,AK,PR,T,W,RE,M FLOW,-DPDZ'
             WRITE(NLG,26)DENF(TAVE),VISF(TAVE),CPF(TAVE),AKF(TAVE),
   .          PRAVE,TAVE,WAVE,RE,RMDOT,-DPDZ
```

```
            WRITE(35,*)'AVG.DEN,VIS,CP,AK,PR,T,W,RE,M FLOW,-DPDZ'
            WRITE(35,26)DENF(TAVE),VISF(TAVE),CPF(TAVE),AKF(TAVE),
     .        PRAVE,TAVE,WAVE,RE,RMDOT,-DPDZ
           ELSE
       PRAVE=VISF(TAVE)*CPF(TAVE)/AKF(TAVE)
       RMDOT=DENF(TAVE)*WAVE*AMATA(1,2)*360./THETAN
            WRITE(NLG,*)'AVG.DEN,VIS,CP,AK,PR,T,W,RE,M FLOW,-DPDZ'
            WRITE(NLG,26)DENF(TAVE),VISF(TAVE),CPF(TAVE),AKF(TAVE),
     .        PRAVE,TAVE,WAVE,RE,RMDOT,-DPDZ
            WRITE(35,*)'AVG.DEN,VIS,CP,AK,PR,T,W,RE,M FLOW,-DPDZ'
            WRITE(35,26)DENF(TAVE),VISF(TAVE),CPF(TAVE),AKF(TAVE),
     .        PRAVE,TAVE,WAVE,RE,RMDOT,-DPDZ
           ENDIF
          WRITE(3,*)'Qs, Qfs ,Qft, Qt'
          WRITE(3,*)QTOTAL1, QTOTAL2, QTOTAL3, QTOTAL
          WRITE(3,*)'As, Afs, Aft, At'
          WRITE(3,*)SAREAT1, SAREAT2, SAREAT3, SAREAT
          WRITE(3,*)'Qs/Qt, As/At, Qfs/Qt, Afs/At, Qft/Qt, Aft/At'
          WRITE(3,*)QTOTAL1/QTOTAL,SAREAT1/SAREAT,QTOTAL2/QTOTAL,
     .    SAREAT2/SAREAT,QTOTAL3/QTOTAL,SAREAT3/SAREAT
      ENDIF
         ENDIF
31       CONTINUE
C
26       FORMAT (1X,11(2X,E12.5))
98       FORMAT(1X,'MAT # = ',I2,1X,'WBAR = ',E13.6,1X,'CF = ',E13.6,1X,
     . 'CF_1 = ',E13.6,1X,'CF_OTHER = ',E13.6,1X, 'CF_DIF (%) = ',
     . E13.6,1X,'RE = ',E13.6,1X,' CFRE = ',E13.6,1X,
     . 'CFRES = ',E13.6)
99       FORMAT(1X,'MAT # = ',I2,1X,'H = ',E13.6,1X,'TW = ',E13.6,1X,
     . ' TB = ',E13.6,1X,'QFLUX = ',E10.3,1X,'Q_1 = ',E10.3,1X,
     . 'Q_2 = ',E10.3,1X,'NU = ',E13.6,1X,'NU_OTHER = ',E13.6,1X,
     . 'NU_DIF (%) = ',E13.6)
100      FORMAT(1X,'PR = ',1X,E13.6,1X,'RI/RO = ',E13.6,1X,
     . 'DPDZ = ',E10.3,1X,'I_RM = ',I4,1X,'RM_CAL = ',1X,E13.6,
     . 1X,'RM_KAY = ',E13.6, 1X,'UMAX = ',E13.6,1X,'Ki = ',E13.6)
101    FORMAT(  ' TOTAL INTERNAL HEAT GENERATION  (W) = ',E13.6,
     .   /,' TOTAL HEAT TRANSFERRED TO FLUID (W) = ',E13.6,
     .   /,' TOTAL HEAT LEAVING THE SURFACE  (W) = ',E13.6)
259    FORMAT(2X,'POSITION   ',2X,'QFLXL      ',2X,'TWL        ',
     .   2X,'HTCL      ',2X,'          ',2X,'          ',
     .   2X,'          ')
C
         RETURN
         END
C*****************************************************************
C          SUBROUTINE INTERP
C          ---------------
C*****************************************************************
      SUBROUTINE INTERP(AXX1,AYY1,INPT1,AXIN1,AYOUT1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AXX1(1),AYY1(1),AXI1(4),AYI1(4)
      AXF1=AXIN1
      INS1=INPT1
      IL1=INS1
      IKS1=INS1
      IKONST1=0
      IF(INS1.LE.3) GO TO 210
      DO 50 II1=1,INS1
      III1=II1
      IF(AXF1-AXX1(I1)) 100,100,50
 50   CONTINUE
 100  IF(III1.GE.3.AND.III1.LT.INS1) GO TO 300
      IF(III1.EQ.INS1) IKONST1=INS1-3
      IKS1=3
      IL1=3
 210  DO 250 IK1=1,IKS1
      IKK1=IK1+IKONST1
      AXI1(IK1)=AXX1(IKK1)
      AYI1(IK1)=AYY1(IKK1)
 250  CONTINUE
      GO TO 400
 300  IL1=4
      DO 350 IK1=1,4
      IKK1=IK1+III1-3
      AXI1(IK1)=AXX1(IKK1)
      AYI1(IK1)=AYY1(IKK1)
 350  CONTINUE
 400  AF1=0.D0
      DO 500 II1=1,IL1
      AC1=1.D0
      DO 450 IJ1=1,IL1
      IF(IJ1.EQ.II1) GO TO 450
      AC1=AC1*(AXF1-AXI1(IJ1))/(AXI1(II1)-AXI1(IJ1))
 450  CONTINUE
 500  AF1=AF1+AC1*AYI1(II1)
      AYOUT1=AF1

      RETURN
      END
C*******************************************************************
          SUBROUTINE MATAREA(NE,MAT,NODES,X,U,TIME,IIIIP,WAREA,WNODES,
    . WELEM,AMATA,WBAR,AMUST,SIGMA,UTER2,PRT,YY,VISTT)
C*******************************************************************
C         Program calculates the X-Sectional area of each material in
C         each 2D model.
C*******************************************************************

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
          COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
    . ,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT

          DIMENSION NE(1),MAT(1),NODES(9,1),X(2,1),U(10,1)
          DIMENSION WAREA(10,1),WNODES(1),WELEM(1)
          DIMENSION AMATA(10,1),WBAR(10,1)
          DIMENSION AMUST(1),SIGMA(1),UTER2(1),PRT(1),YY(1),VISTT(1)

C----- Function declaration
          REAL*8 LINELEN,QUADAREA

          REAL*8 PX(9),PY(9),X1,X2,Y1,Y2
          REAL*8 A,B,C,D,P,Q
          INTEGER IA(8),IB(8),IC(8),ID(8),IP(8),IQ(8)
          DATA IA/1,5,9,8,5,2,6,9/
```

```
        DATA IB/5,2,6,9,9,6,3,7/
        DATA IC/9,6,3,7,8,9,7,4/
        DATA ID/8,9,7,4,1,5,9,8/
        DATA IP/5,2,6,9,8,9,7,4/
        DATA IQ/1,5,9,8,9,6,3,7/

        DO 10 IM=1,NMAT
        DO 11 IEQ=1,NPDE
         AMATA(IEQ,IM) = 0.0
         WAREA(IEQ,IM) = 0.0
11      CONTINUE
10      CONTINUE

        IF (NPRNT3 .NE. 0) THEN
        IF (NOUT .EQ. 1 .AND. (NSTEP .EQ. NSTEPT .OR. IIIIP.EQ.0))THEN
        WRITE(NOU,36) (IEQ,IEQ=1,NPDE)
36      FORMAT(//,' ELEMENT',2X,'ELEMENT AREA',10(2X,'U ELEMENT(',
     .   I2,')'))
        ENDIF
        ENDIF

        DO 20 IE=1,NELEM
        DO 21 IEQ=1,NPDE
         MATNO = MAT(IE)
         WNODES(IEQ) = 0.0
         DO 30 IN=1,NE(IE)
          PX(IN) = X(1,NODES(IN,IE))
          PY(IN) = X(2,NODES(IN,IE))
          WNODES(IEQ) = WNODES(IEQ) + U(IEQ,NODES(IN,IE))
30       CONTINUE

         WELEM(IEQ) = WNODES(IEQ)/NE(IE)

         IF( NE(IE) .EQ. 4 ) THEN
          A = LINELEN( PX(1),PY(1),PX(2),PY(2) )
          B = LINELEN( PX(2),PY(2),PX(3),PY(3) )
          C = LINELEN( PX(3),PY(3),PX(4),PY(4) )
          D = LINELEN( PX(4),PY(4),PX(1),PY(1) )
          P = LINELEN( PX(2),PY(2),PX(4),PY(4) )
          Q = LINELEN( PX(1),PY(1),PX(3),PY(3) )
          ELAREA = QUADAREA(A,B,C,D,P,Q)
          WAREA(IEQ,MATNO) = WAREA(IEQ,MATNO) + ELAREA*WELEM(IEQ)
          AMATA(IEQ,MATNO) = AMATA(IEQ,MATNO) + ELAREA
         ELSE !------- 8 OR 9 NODED ELEMENTS
          CALL PSOLVE(PX,PY,1,3,2,4,9)
          DO 50 J=1,4
           A = LINELEN( PX(IA(J)),PY(IA(J)),PX(IA(J+4)),PY(IA(J+4)) )
           B = LINELEN( PX(IB(J)),PY(IB(J)),PX(IB(J+4)),PY(IB(J+4)) )
           C = LINELEN( PX(IC(J)),PY(IC(J)),PX(IC(J+4)),PY(IC(J+4)) )
           D = LINELEN( PX(ID(J)),PY(ID(J)),PX(ID(J+4)),PY(ID(J+4)) )
           P = LINELEN( PX(IP(J)),PY(IP(J)),PX(IP(J+4)),PY(IP(J+4)) )
           Q = LINELEN( PX(IQ(J)),PY(IQ(J)),PX(IQ(J+4)),PY(IQ(J+4)) )
           ELAREA = QUADAREA(A,B,C,D,P,Q)
           WAREA(IEQ,MATNO) = WAREA(IEQ,MATNO) + ELAREA*WELEM(IEQ)
           AMATA(IEQ,MATNO) = AMATA(IEQ,MATNO) + ELAREA
50        CONTINUE
         END IF
21      CONTINUE
```

```
          IF (NPRNT3 .NE. 0) THEN
          IF (NOUT .EQ. 1 .AND. (NSTEP .EQ. NSTEPT .OR. IIIIP .EQ. 0))THEN
          WRITE(NOU,45) IE,ELAREA,(WELEM(IEQ),IEQ=1,NPDE)
45        FORMAT(2X,I4,11(4X,1PE11.3))
          ENDIF
          ENDIF
20        CONTINUE

          IF (NPRNT3 .NE. 0) THEN
          IF (NOUT .EQ. 1 .AND. (NSTEP .EQ. NSTEPT .OR. IIIIP .EQ. 0))THEN
          WRITE(NOU,35)
          DO 33 IM=1,NMAT
          DO 33 IEQ=1,NPDE
          IF (AMATA(IEQ,IM).EQ.0.0)GO TO 33
          WRITE(NOU,40) TIME,IM,AMATA(IEQ,IM),V/AREA(IEQ,IM)/AMATA(IEQ,IM),
     .       IEQ
33        CONTINUE
          ENDIF
          ENDIF

35        FORMAT(/,5X,'TIME',5X,'MATERIAL',4X,'AREA',3X,4X,'AVG. U',
     .   4X,'EQN NO')
40        FORMAT(2X,F8.4,6X,I3,3X,1PE11.3,1X,1PE11.3,4X,I3)
      WRITE(NOU,110) (I,AMUST(I),I=1,NELEM)
110     FORMAT(/,1X,3('ELEM',8X,'VISC',15X)/,3(I5,5X,1PE11.4,10X))
      WRITE(NOU,113) (I,PRT(I),I=1,NELEM)
113     FORMAT(/,1X,3('ELEM',8X,'PRT ',15X)/,3(I5,5X,1PE11.4,10X))
      WRITE(NOU,111) (I,SIGMA(I),I=1,NELEM)
111     FORMAT(/,1X,3('ELEM',8X,'STRESS',14X)/,3(I5,5X,1PE11.4,10X))
      WRITE(NOU,112) (I,UTER2(I),I=1,NNODE)
112     FORMAT(/,1X,3('NODE',8X,'UOLD',15X)/,3(I5,5X,1PE11.4,10X))
      WRITE(NOU,114) (I,YY(I),I=1,NELEM)
114     FORMAT(/,1X,3('NODE',8X,'Y',15X)/,3(I5,5X,1PE11.4,10X))

          RETURN
          END

C**********************************************************************
          REAL*8 FUNCTION LINELEN(X1,Y1,X2,Y2)
          REAL*8 X1,Y1,X2,Y2
          LINELEN = DSQRT( (X1-X2)*(X1-X2) + (Y1-Y2)*(Y1-Y2) )
          RETURN
          END
C**********************************************************************
          REAL*8 FUNCTION QUADAREA(A,B,C,D,P,Q)
          REAL*8 A,B,C,D,P,Q,TERM1,TERM2,DIFF,ABSDIFF
          TERM1 = 4.0 * P*P * Q*Q
          TERM2 = B*B + D*D - A*A - C*C
          DIFF = TERM1 - (TERM2**2.0)
          ABSDIFF = DABS(DIFF)
C--       Since the area is its magnitude value not vector, so we can
C--       use its absolute to prevent any square root of a negative number.
          QUADAREA = 0.25 * DSQRT(ABSDIFF)
          RETURN
          END
C
C********************************************************
          SUBROUTINE PSOLVE(PX,PY,J1,J2,J3,J4,K)
```

```
        IMPLICIT REAL*8 (A-H,O-Z)
C**********************************************************
        REAL*8 PX(1),PY(1)
        DX1 = PX(J2)-PX(J1)
        DY1 = PY(J2)-PY(J1)
        DX2 = PX(J4)-PX(J3)
        DY2 = PY(J4)-PY(J3)
        NUM = 1
        IF(DY2.NE.0.0) NUM = NUM + 1
        IF(DX2.NE.0.0) NUM = NUM + 2
        IF(DY1.NE.0.0) NUM = NUM + 4
        IF(DX1.NE.0.0) NUM = NUM + 8

C-----------0 1 2 3 4 5 6  7 8  9 10 11 12 13 14 15
        GO TO(1,1,1,1,1,1,5,10,1,15, 1,20, 1,25,30,35) NUM

1       PRINT 100,NUM
100     FORMAT('  IN PSOLVE.NO SOLUTION. NUM=',I5)
        RETURN
5       PX(K) = PX(J1)
        PY(K) = PY(J3)
        RETURN
10      PX(K) = PX(J1)
        PY(K) = PY(J3) + (PX(K)-PX(J3))*DY2/DX2
        RETURN
15      PX(K) = PX(J3)
        PY(K) = PY(J1)
        RETURN
20      PY(K) = PY(J1)
        PX(K) = PX(J3) + (PY(K)-PY(J3))*DX2/DY2
        RETURN
25      PX(K) = PX(J3)
        PY(K) = PY(J1) + (PX(K)-PX(J1))*DY1/DX1
        RETURN
30      PY(K) = PY(J3)
        PX(K) = PX(J1) + (PY(K)-PY(J1))*DX1/DY1
        RETURN
35      F1 = DX1/DY1
        F2 = DX2/DY2
        PY(K) = (PX(J1)-PY(J1)*F1-PX(J3)+PY(J3)*F2)/(F2-F1)
        PX(K) = PX(J1)+(PY(K)-PY(J1))*F1
        RETURN
        END
C**********************************************************************
C
C          FUNCTION PROPRTES
C         --------------
C
C**********************************************************************
    REAL*8 FUNCTION PROPRTES(MATNUM,IEQ,JREG,PTEMPP,IMAT,VAR,TVAR)

    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION TVAR(10,10,1,20),VAR(10,10,1,20),IMAT(10,10,1)
    REAL*8 TMP(20),PROPS(20)

     DO 10006 J=1,IMAT(IEQ,MATNUM,JREG)
      TMP(J) = TVAR(IEQ,MATNUM,JREG,J)
      PROPS(J)= VAR(IEQ,MATNUM,JREG,J)
```

```
10006   CONTINUE

     CALL INTERP(TMP,PROPS,IMAT(IEQ,MATNUM,JREG),PTEMPP,POUT)
     PROPRTES = POUT

    RETURN
    END
C*************************************************************
   SUBROUTINE RESULTS(ITER,REN,X,U,START2,SIGMA,UELEM)
C*************************************************************
C.....
C.....CALCULATES STRESS FROM SHAPE FUNCTIONS
C.  FOR QUADRILATERAL ELEMENTS
C.....
C   CALLED BY:
C
C   CALLS   : SHAPE4
C
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)

   COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
   COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
  . ,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
   COMMON/FILENAMES/INFILE,JTITLE
   COMMON/RM_UMAX/IRM,RM_CAL,RMKAY,UMAX,CKARMANI,IRMA,RM_CALA,UMAXA
   COMMON/YSPLUS/YPLUSA,SPLUSA,ALLA,YA,SA,DFPA,DFCA,ALPA,ALCA,
  .  TWYA,TWSA
   CHARACTER*20 INFILE
C
   INCLUDE 'THVAR.H'
C
   DIMENSION X(2,1650),U(10,1),ARES(4)
   DIMENSION SIGMA(1)
   DIMENSION UELEM(10,1)
   DIMENSION YPLUSA(1550),SPLUSA(1550),ALLA(1550),YA(1550),SA(1550),
  .DFPA(1550),DFCA(1550),ALPA(1550),ALCA(1550),TWYA(1550),TWSA(1550)
   DIMENSION IRMA(50),RM_CALA(50),UMAXA(50)
   LOGICAL START2

C
        IF (START2) THEN
    OPEN (1,FILE=INFILE(1:JTITLE)//'.da1',STATUS='UNKNOWN')
C       START2 = .FALSE.
        ENDIF
C
     DO 5 NEL=1,NELEM
     WRITE(52,99)NEL,CMU,FMUKE(NEL),DEN,UELEM(3,NEL),
  .   UELEM(4,NEL),VISTT(NEL)
99    FORMAT(1X,I4,14(1X,E10.4))
5     CONTINUE
    WRITE(52,113) (I,VISTT(I),I=1,NELEM)
113   FORMAT(/,1X,3('NODE',8X,'VISTT',15X)/,3(I5,5X,1PE11.4,10X))
C
   WRITE(1,*)'NEL,Y,Y+,DP,S,S+,DC,ALP,ALC,ALL,TY,TS,VIS,VISE,SIG,U,T'
   DO 122 I=1,NELEM
        IF(FPROP.EQ.'FIXED')THEN
     VISEFF=AMUST(I)+VIS
```

```
      WRITE(1,121)I,YA(I),YPLUSA(I),DFPA(I),SA(I),SPLUSA(I),DFCA(I),
     . ALPA(I),ALCA(I),ALLA(I),TWYA(I),TWSA(I),
     . VIS,VISEFF,SIGMA(I),UELEM(1,I),UELEM(2,I)
           ELSEIF (FPROP.EQ.'FIXTB')THEN
      VISEFF=AMUST(I)+VISF(TAVE)
      WRITE(1,121)I,YA(I),YPLUSA(I),DFPA(I),SA(I),SPLUSA(I),DFCA(I),
     . ALPA(I),ALCA(I),ALLA(I),TWYA(I),TWSA(I),
     . VISF(TAVE),VISEFF,SIGMA(I),UELEM(1,I),UELEM(2,I)
           ELSE
      VISEFF=AMUST(I)+VISF(UELEM(2,I))
      WRITE(1,121)I,YA(I),YPLUSA(I),DFPA(I),SA(I),SPLUSA(I),DFCA(I),
     . ALPA(I),ALCA(I),ALLA(I),TWYA(I),TWSA(I),
     . VISF(UELEM(2,I)),VISEFF,SIGMA(I),UELEM(1,I),UELEM(2,I)
           ENDIF
121   FORMAT(1X,I4,16(1X,1PE11.4))
122   CONTINUE
C
     IF(RMOPT.EQ.'RMCALL')THEN
      WRITE(1,*)'IRM,RM_CAL,UMAX'
      IF(NGEOMTYPE.EQ.21 .OR. NGEOMTYPE.EQ.22)IEND=30
      IF(NGEOMTYPE.EQ.11)IEND=20
      DO 1222 I=1,IEND
      WRITE(1,121)IRMA(I),RM_CALA(I),UMAXA(I)
1222  CONTINUE
     ENDIF
C
     ZERO = 0.0
     ZONE = 1.0
C*******
C..TUBE*
C*******
     IF (NGEOMTYPE.EQ.0) THEN
           WRITE(1,25)ITER,DPDZ,REN
           WRITE(1,*)'IJ,RELEM,RRO,UEL,UOUCL,TAU,TAUN,YPLUS,UPLUS,ULOG'
           DO 19 IJ=1,NELEM
      UEL = (U(1,IJ)+U(1,IJ+1))/2.
      RELEM = (X(1,IJ)+X(1,IJ+1))/2.
      RNODE = RELEM/X(1,NELEM+1)
      UOUCL = UEL/U(1,1)
      USTAR = (SIGMA(NELEM)/DEN)**0.5
      UPLUS = UEL/USTAR
      YPLUS = (RO-RELEM)*DEN*USTAR/VIS
      TAUN = SIGMA(IJ)/SIGMA(NELEM)
C... KAY'S, PG 171
      IF(YPLUS.GT.10.8)THEN
       ULOG = 2.44*DLOG(YPLUS) + 5.0
      ELSE
       ULOG = YPLUS
      ENDIF
      WRITE(1,27)IJ,RELEM,RNODE,UEL,UOUCL,SIGMA(IJ),TAUN,YPLUS,UPLUS,
     .     ULOG
27    FORMAT (1X,I4,17(2X,E9.3))
19         CONTINUE
           WRITE(1,*)'DPDZ*DH/4 = ',DPDZ*DH/4.
           WRITE(1,*)'DPDZ*AFLOW = ',DPDZ*AFLOW
           WRITE(1,*)'TAWAVG*PWET = ',SIGMA(NELEM)*PWET
           ENDIF
C***********
```

```
C.. ANNULUS*
C***********
   IF (NGEOMTYPE.EQ.1) THEN
C... INNER WALL OF ANNULUS
        APOS1 = DSQRT((X(1,72)-X(1,2))**2.+(X(2,72)-X(2,2))**2.)
C... OUTER WALL OF ANNULUS
        APOS2 = DSQRT((X(1,141)-X(1,71))**2.+(X(2,141)-X(2,71))**2.)
        STRAVE = (SIGMA(2)*APOS1+SIGMA(70)*APOS2)/(APOS1+APOS2)
C
        RMR_KAY = RMKAY/RI
        RMR_CAL = RM_CAL/RI
        RMR_LAM = DSQRT(((RO/RI)**2.-1.)/2./DLOG(RO/RI))
        WRITE(NLG,*)'RM/RI_KAY,RM/RI_CAL,RM/RI_LAM'
        WRITE(NLG,*)RMR_KAY,RMR_CAL,RMR_LAM
C
        TAUR = RO/RI*(RM_CAL**2.-RI**2.)/(RO**2.-RM_CAL**2.)
C
    WRITE(NLG,*)'DS INNER, DS OUTER'
    WRITE(NLG,*)APOS1,APOS2
        WRITE(NLG,*)'TAU_I, TAU_O, TAU_AVG, TAU_I/TAU_O_CAL, _THEORY'
        WRITE(NLG,*)SIGMA(2),SIGMA(70),STRAVE,SIGMA(2)/SIGMA(70),TAUR
        WRITE(NLG,*)'DPDZ*DH/4 = ',DPDZ*DH/4.
        WRITE(NLG,*)'DPDZ*AFLOW = ',DPDZ*AFLOW
        WRITE(NLG,*)'TAWAVG*PWET = ',STRAVE*PWET
C... KAY'S, PG 241-242
        WRITE(NLG,*)'IJ,RR,VIST/VIS,TAU,TAUN,Y+,U+,ULOG,XI,YI,XO,YO'
        WRITE(NLG,*)'(RO-R)/(O-I),U/UM,(R-RI)/(O-I),T,(TI-T)/(TI-TO)'
        DO 18 IJ=2,NELEM
    UEL = (U(1,IJ)+U(1,IJ+1))/2.
    TEL = (U(2,IJ)+U(2,IJ+1))/2.
    RELEM = (X(1,IJ)+X(1,IJ+1))/2.
    RNODE = RELEM/X(1,NELEM+1)
    UOUCL = UEL/UMAX
        RR1 = (RO-RELEM)/(RO-RI)
        RR2 = (RELEM-RI)/(RO-RI)
        TRR = (U(2,2)-TEL)/(U(2,2)-U(2,70))
    IF (RELEM .LE. RM_CAL)THEN
     USTAR = (SIGMA(2)/DEN)**0.5
     YPLUS = (RELEM-RI)*DEN*USTAR/VIS
     TAUN = SIGMA(IJ)/SIGMA(2)
         XIN = (RM_CAL-RELEM)/(RM_CAL-RI)
         YIN = (UMAX-UEL)/USTAR
         XOUT = 0.0
         YOUT = 0.0
    ELSE
     USTAR = (SIGMA(NELEM)/DEN)**0.5
     YPLUS = (RO-RELEM)*DEN*USTAR/VIS
     TAUN = SIGMA(IJ)/SIGMA(70)
         XIN=0.0
         YIN=0.0
         XOUT = (RELEM-RM_CAL)/(RO-RM_CAL)
         YOUT = (UMAX-UEL)/USTAR
    ENDIF
    UPLUS = UEL/USTAR
    IF (RELEM .LE. RM_CAL)THEN
C... BARROW ET AL. (1965) FOR RO/RI < 10
     IF(YPLUS.GT.10.8)THEN
     ULOG = 2.7*(RI/RO)**0.353*DLOG(YPLUS) + 3.6*(RI/RO)**-0.439
```

```
        ELSE
         ULOG = YPLUS
        ENDIF
       ELSE
C... KAY'S, PG 171
        IF(YPLUS.GT.10.8)THEN
         ULOG = 2.44*DLOG(YPLUS) + 5.0
        ELSE
         ULOG = YPLUS
        ENDIF
       ENDIF
       WRITE(NLG,27)IJ,RR2,AMUST(IJ)/VIS,SIGMA(IJ),TAUN,
     .    YPLUS,UPLUS,ULOG,XIN,YIN,XOUT,YOUT,RR1,UOUCL,
     .    RR2,UELEM(2,IJ),TRR
18      CONTINUE
      ENDIF
C*************************
C..EDWARDS FINNED TUBE #1*
C*************************
      IF (NGEOMTYPE.EQ.3) THEN
          WRITE(1,25)ITER,DPDZ,REN
25        FORMAT (/,5X,'ITER',6X,'DPDZ',7X,'REYNOLDS',/,
     .    4X,I5,2(2X,E12.6),/)
          WRITE(1,251)
          WRITE(1,255)
          WRITE(1,26) ZERO,ZONE,ZONE,ZONE,ZONE,ZONE,ZONE
          DO 20 IJ=0,38
          IJ2 = 0
          DO 10 INODE=470+IJ,704+IJ,78
          IJ2 = IJ2 + 1
          ARES(IJ2) = U(1,INODE)/U(1,1)
10        CONTINUE
          APOS = X(1,IJ+2)/RO
          BRES1 = U(1,IJ+2)/U(1,1)
          BRES2 = U(1,IJ+80)/U(1,1)
C---------------- X   9.0    13.5   18.0   22.5  DEG ----
          WRITE(1,26) APOS,BRES1,BRES2,ARES(1),ARES(2),ARES(3),ARES(4)
26        FORMAT (1X,7(2X,E11.5))
251       FORMAT (2X,'POSITION AND U/U(1,1) ALONG CONSTANT ANGLE')
255   FORMAT(2X,'        X',2X,'   0.0 DEG',2X,'   4.5 DEG',
     .   2X,'   9.0 DEG',2X,'  13.5 DEG',2X,'  18.0 DEG',
     .   2X,'  22.5 DEG')
20        CONTINUE
          ENDIF
C*************************
C..EDWARDS FINNED TUBE #2*
C*************************
      IF (NGEOMTYPE.EQ.4) THEN
          WRITE(1,25)ITER,DPDZ,REN
          WRITE(1,26) ZERO,ZONE,ZONE,ZONE,ZONE,ZONE,ZONE
          DO 21 IJ=0,23
          IJ2 = 0
          DO 11 INODE=74+IJ,218+IJ,48
          IJ2 = IJ2 + 1
          ARES(IJ2) = U(1,INODE)/U(1,1)
11        CONTINUE
          APOS = X(1,IJ+2)/X(1,25)
          BRES1 = U(1,IJ+2)/U(1,1)
          BRES2 = U(1,IJ+50)/U(1,1)
C--------------------X   9.0    13.5    18.0    22.5  DEG ----
          WRITE(1,26) APOS,BRES1,BRES2,ARES(1),ARES(2),ARES(3),ARES(4)
21        CONTINUE
          ENDIF
C*************************
C..EDWARDS FINNED TUBE #3*
C*************************
    IF (NGEOMTYPE.EQ.5) THEN
          WRITE(1,25)ITER,DPDZ,REN
          WRITE(1,26) ZERO,ZONE,ZONE,ZONE,ZONE,ZONE,ZONE
          DO 22 IJ=0,23
          IJ2 = 0
          DO 12 INODE=74+IJ,170+IJ,48
          IJ2 = IJ2 + 1
          ARES(IJ2) = U(1,INODE)/U(1,1)
12        CONTINUE
          APOS = X(1,IJ+2)/X(1,25)
          BRES1 = U(1,IJ+2)/U(1,1)
          BRES2 = U(1,IJ+50)/U(1,1)
C--------------------X   0.0   2.86  3.75    7.5    11.25 DEG ----
          WRITE(1,26) APOS,BRES1,BRES2,ARES(1),ARES(2),ARES(3)
22        CONTINUE
          ENDIF
C
C..EDWARDS TUBE #1 TAU WALL ALONG THE PERIPHERY
C
    IF (NGEOMTYPE.EQ.3) THEN
          APOST = 0.0
          STRAVE = 0.0
          DO 31 IJ=702,117,-39
          APOS1 = DATAN((X(1,IJ+1)-X(1,IJ+40))/(X(2,IJ+1)-X(2,IJ+40)))
          APOS = (X(1,IJ+1)-X(1,IJ+40))/DSIN(-1.0*APOS1)
          APOST = APOS + APOST
          STRAVE = SIGMA(IJ)*APOS + STRAVE
          IF (IJ .EQ. 117) STRAVE = SIGMA(IJ)*APOS + STRAVE
31        CONTINUE
          DO 32 IJ=116,99,-1
          APOS = X(1,IJ+1)-X(1,IJ)
          APOST = APOS + APOST
          STRAVE = SIGMA(IJ)*APOS + STRAVE
32        CONTINUE
          DO 33 IJ=59,20,-39
          APOS = X(2,IJ+40)-X(2,IJ+1)
          APOST = APOS + APOST
          STRAVE = SIGMA(IJ)*APOS + STRAVE
33        CONTINUE
          STRAVE = STRAVE/APOST

          IF (START2) THEN
     OPEN (1,FILE=INFILE(1:JTITLE)//'.da1',STATUS='UNKNOWN')
          START2 = .FALSE.
          ENDIF
          WRITE(1,25)ITER,DPDZ,REN
          APOS = 0.0
          APOS1 = 0.0
          APOS2 = 0.0
          DO 34 IJ=702,117,-39
          ARSS = SIGMA(IJ)/STRAVE
          ANG1 = DATAN((X(1,IJ+1)-X(1,IJ+40))/(X(2,IJ+1)-X(2,IJ+40)))
          APOS1 = APOS
          APOS3 = APOS2
          APOS2 = ((X(1,IJ+1)-X(1,IJ+40))/DSIN(-1.0*ANG1))/2.0
          APOS  = APOS1 + APOS2 + APOS3
          IF(IJ.EQ.702)WRITE(1,261)
          IF(IJ.EQ.702)WRITE(1,*)'AVG. SHEAR STRESS = ',STRAVE
          IF(IJ.EQ.702)WRITE(1,*)'DPDZ*DH/4 = ',DPDZ*DH/4.
          IF(IJ.EQ.702)WRITE(1,*)'DPDZ*AFLOW = ',DPDZ*AFLOW
          IF(IJ.EQ.702)WRITE(1,*)'TAWAVG*PWET = ',STRAVE*PWET
          WRITE(1,26) APOS,ARSS,SIGMA(IJ)
261   FORMAT(2X,'DIST FROM THE MID BTN FINS,', 2X,
     . 'LOCAL TAU WALL/AV. TAU WALL,',2X,'LOCAL TAU WALL')
34        CONTINUE
          DO 35 IJ=117,99,-1
          ARSS = SIGMA(IJ)/STRAVE
          APOS1 = APOS
          APOS3 = APOS2
          APOS2 = (X(1,IJ+1)-X(1,IJ))/2.0
          APOS  = APOS1 + APOS2 + APOS3
          WRITE(1,26) APOS,ARSS,SIGMA(IJ)
35        CONTINUE
          DO 36 IJ=98,20,-39
          ARSS = SIGMA(IJ)/STRAVE
          APOS1 = APOS
          APOS3 = APOS2
          APOS2 = (X(2,IJ+40)-X(2,IJ+1))/2.0
          APOS  = APOS1 + APOS2 + APOS3
          IF (IJ .LT. 98) WRITE(1,26) APOS,ARSS,SIGMA(IJ)
36        CONTINUE
          ENDIF
C****************************
C...PATANKAR'S FINNED ANNULUS*
C****************************
      IF (NGEOMTYPE.EQ.11) THEN
          APOST = 0.0
          STRAVE = 0.0
      IJ0=0
C
      ELINC=50
      NODINC=51
C
C... SHEATH
          DO 51 IJ=1021,52,-NODINC
          APOS = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .     (X(2,IJ)-X(2,IJ-NODINC))**2.)
          APOST = APOS + APOST
      NELIJ=951-IJ0*ELINC
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
51        CONTINUE
      APOST1=APOST
      STRAVE1=STRAVE
      STRSH=STRAVE1/APOST1
C... FIN SIDE
      IJ0=0
          DO 52 IJ=1,20
          APOS = DSQRT((X(1,IJ+1)-X(1,IJ))**2.-(X(2,IJ+1)-X(2,IJ))**2.)
          APOST = APOS + APOST
      NELIJ=1+IJ0
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
52        CONTINUE
      APOST2=APOST-APOST1
      STRAVE2=STRAVE-STRAVE1
      STRFS=STRAVE2/APOST2
C... INNER SURFACE AVG STRESS
          STRAVEIN = STRAVE/APOST
C... TUBE
      IJ0=0
          DO 53 IJ=1071,102,-NODINC
          APOS = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .     (X(2,IJ)-X(2,IJ-NODINC))**2.)
          APOST = APOS + APOST
      NELIJ=1000-ELINC*IJ0
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
53        CONTINUE
      APOST4=APOST-APOST1-APOST2
      STRAVE4=STRAVE-STRAVE1-STRAVE2
      STRTB=STRAVE4/APOST4
C..
      STRAVE = STRAVE/APOST
C..
      WRITE(1,*)'LENGTH OF SHEATH, FIN SIDE, FIN TIP, TUBE'
      WRITE(1,*)APOST1,APOST2,APOST3,APOST4
          WRITE(1,*)'STRESS ON SH, F SIDE, F TIP, IN AVG., TUBE, ALL AVG'
          WRITE(1,*)STRSH,STRFS,STRFT,STRAVEIN,STRTB,STRAVE
          WRITE(1,*)'DPDZ*DH/4 =  ',DPDZ*DH/4.
          WRITE(1,*)'DPDZ*AFLOW =  ',DPDZ*AFLOW
          WRITE(1,*)'TAWAVG*PWET = ',STRAVE*PWET
          WRITE(1,261)
          APOS = 0.0
          APOS1 = 0.0
C... SHEATH
      WRITE(1,*)'OVER THE SHEATH'
      IJ0=0
          DO 54 IJ=1021,52,-NODINC
      NELIJ=951-ELINC*IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRSH
          APOS1 = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2.+(X(2,IJ)
     .    -X(2,IJ-NODINC))**2.)
          APOS = APOS + APOS1
          WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
54        CONTINUE
C... FIN SIDE
      APOS=0.0
      APOS1=0.0
      IJ0=0
      WRITE(1,*)'OVER THE FIN SIDE'
          DO 57 IJ=1,20
      NELIJ=1+IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRFS
```

```
           APOS1 = DSQRT((X(1,IJ+1)-X(1,IJ))**2.-(X(2,IJ+1)-X(2,IJ))**2.)
           APOS = APOS + APOS1
           WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
57         CONTINUE
C... TUBE
      APOS = 0.0
      APOS2 = 0.0
      IJ0=0
      WRITE(1,*)'OVER THE TUBE SURFACE'
           DO 58 IJ=1071,102,-NODINC
      NELIJ=1000-ELINC*IJ0
      IJ0=IJ0+1
           ARSS = SIGMA(NELIJ)/STRTB
           APOS2 = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .      (X(2,IJ)-X(2,IJ-NODINC))**2.)
           APOS = APOS + APOS2
           WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
58         CONTINUE
           ENDIF
C*******************************
C...PATANKAR'S UNFINNED ANNULUS*
C*******************************
      IF (NGEOMTYPE.EQ.12) THEN
           APOST = 0.0
           STRAVE = 0.0
      IJ0=0
C
      ELINC=50
      NODINC=51
C
C... SHEATH
           DO 551 IJ=1021,52,-NODINC
           APOS = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .      (X(2,IJ)-X(2,IJ-NODINC))**2.)
           APOST = APOS + APOST
      NELIJ=951-IJ0*ELINC
      IJ0=IJ0+1
           STRAVE = SIGMA(NELIJ)*APOS + STRAVE
551        CONTINUE
      APOST1=APOST
      STRAVE1=STRAVE
      STRSH=STRAVE1/APOST1
C... INNER SURFACE AVG STRESS
           STRAVEIN = STRAVE/APOST
C... TUBE
      IJ0=0
           DO 553 IJ=1071,102,-NODINC
           APOS = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .      (X(2,IJ)-X(2,IJ-NODINC))**2.)
           APOST = APOS + APOST
      NELIJ=1000-ELINC*IJ0
      IJ0=IJ0+1
           STRAVE = SIGMA(NELIJ)*APOS + STRAVE
553        CONTINUE
      APOST4=APOST-APOST1
      STRAVE4=STRAVE-STRAVE1
      STRTB=STRAVE4/APOST4
C..
```

```
      STRAVE = STRAVE/APOST
C..
      WRITE(1,*)'LENGTH OF SHEATH, FIN SIDE, FIN TIP, TUBE'
      WRITE(1,*)APOST1,APOST2,APOST3,APOST4
         WRITE(1,*)'STRESS ON SH, F SIDE, F TIP, IN AVG., TUBE, ALL AVG'
         WRITE(1,*)STRSH,STRFS,STRFT,STRAVEIN,STRTB,STRAVE
         WRITE(1,*)'DPDZ*DH/4 =  ',DPDZ*DH/4.
         WRITE(1,*)'DPDZ*AFLOW = ',DPDZ*AFLOW
         WRITE(1,*)'TAWAVG*PWET = ',STRAVE*PWET
         WRITE(1,261)
         APOS = 0.0
         APOS1 = 0.0
C... SHEATH
      WRITE(1,*)'OVER THE SHEATH'
      IJ0=0
         DO 554 IJ=1021,52,-NODINC
      NELIJ=951-ELINC*IJ0
      IJ0=IJ0+1
         ARSS = SIGMA(NELIJ)/STRSH
         APOS1 = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2.+(X(2,IJ)
     .   -X(2,IJ-NODINC))**2.)
         APOS = APOS + APOS1
         WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
554      CONTINUE
C... TUBE
      APOS = 0.0
      APOS2 = 0.0
      IJ0=0
      WRITE(1,*)'OVER THE TUBE SURFACE'
         DO 558 IJ=1071,102,-NODINC
      NELIJ=1000-ELINC*IJ0
      IJ0=IJ0+1
         ARSS = SIGMA(NELIJ)/STRTB
         APOS2 = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .    (X(2,IJ)-X(2,IJ-NODINC))**2.)
         APOS = APOS + APOS2
         WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
558      CONTINUE
         ENDIF
C****************
C...FA8 UNFINNED*
C****************
      IF (NGEOMTYPE.EQ.22) THEN
         APOST = 0.0
         STRAVE = 0.0
      IJ0=0
C
      ELINC=48
      NODINC=49
C
C... SHEATH
         DO 561 IJ=1474,53,-NODINC
         APOS = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .    (X(2,IJ)-X(2,IJ-NODINC))**2.)
         APOST = APOS + APOST
      NELIJ=1396-IJ0*ELINC
      IJ0=IJ0+1
         STRAVE = SIGMA(NELIJ)*APOS + STRAVE
```

```
561       CONTINUE
      APOST1=APOST
      STRAVE1=STRAVE
      STRSH=STRAVE1/APOST1
C... INNER SURFACE AVG STRESS
          STRAVEIN = STRAVE/APOST
C... TUBE
      IJ0=0
          DO 563 IJ=1519,98,-NODINC
          APOS = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .     (X(2,IJ)-X(2,IJ-NODINC))**2.)
          APOST = APOS + APOST
      NELIJ=1440-ELINC*IJ0
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
563       CONTINUE
      APOST4=APOST-APOST1
      STRAVE4=STRAVE-STRAVE1
      STRTB=STRAVE4/APOST4
C..
      STRAVE = STRAVE/APOST
C..
      WRITE(1,*)'LENGTH OF SHEATH, FIN SIDE, FIN TIP, TUBE'
      WRITE(1,*)APOST1,APOST2,APOST3,APOST4
          WRITE(1,*)'STRESS ON SH, F SIDE, F TIP, IN AVG., TUBE, ALL AVG'
          WRITE(1,*)STRSH,STRFS,STRFT,STRAVEIN,STRTB,STRAVE
          WRITE(1,*)'DPDZ*DH/4 =  ',DPDZ*DH/4.
          WRITE(1,*)'DPDZ*AFLOW =  ',DPDZ*AFLOW
          WRITE(1,*)'TAWAVG*PWET = ',STRAVE*PWET
          WRITE(1,261)
          APOS = 0.0
          APOS1 = 0.0
C... SHEATH
      WRITE(1,*)'OVER THE SHEATH'
      IJ0=0
          DO 564 IJ=1474,53,-NODINC
      NELIJ=1396-ELINC*IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRSH
          APOS1 = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2.+(X(2,IJ)
     .    -X(2,IJ-NODINC))**2.)
          APOS = APOS + APOS1
          WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
564       CONTINUE
C... TUBE
      APOS = 0.0
      APOS2 = 0.0
      IJ0=0
      WRITE(1,*)'OVER THE TUBE SURFACE'
          DO 568 IJ=1519,98,-NODINC
      NELIJ=1440-ELINC*IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRTB
          APOS2 = DSQRT((X(1,IJ)-X(1,IJ-NODINC))**2. +
     .     (X(2,IJ)-X(2,IJ-NODINC))**2.)
          APOS = APOS + APOS2
          WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
568       CONTINUE
```

```
          ENDIF
C********
C... FA8*
C********
      IF (NGEOMTYPE.EQ.21) THEN
          WRITE(1,25)ITER,DPDZ,REN
          APOST = 0.0
          STRAVE = 0.0
      IJ0=0
C... SHEATH
          DO 41 IJ=1474,347,-49
          APOS = DSQRT((X(1,IJ)-X(1,IJ-49))**2. +
     .     (X(2,IJ)-X(2,IJ-49))**2.)
          APOST = APOS + APOST
      NELIJ=1396-IJ0*48
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
41        CONTINUE
      APOST1=APOST
      STRAVE1=STRAVE
      STRSH=STRAVE1/APOST1
C... FIN SIDE
      IJ0=0
          DO 42 IJ=298,314
          APOS = DSQRT((X(1,IJ+1)-X(1,IJ))**2.-(X(2,IJ+1)-X(2,IJ))**2.)
          APOST = APOS + APOST
      NELIJ=292+IJ0
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
42        CONTINUE
      APOST2=APOST-APOST1
      STRAVE2=STRAVE-STRAVE1
      STRFS=STRAVE2/APOST2
C... FIN TIP
      IJ0=0
          DO 43 IJ=315,70,-49
          APOS = DSQRT((X(1,IJ)-X(1,IJ-49))**2.+(X(2,IJ)-X(2,IJ-49))**2.)
      NELIJ=261-48*IJ0
      IJ0=IJ0+1
          APOST = APOS + APOST
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
43        CONTINUE
      APOST3=APOST-APOST1-APOST2
      STRAVE3=STRAVE-STRAVE1-STRAVE2
      STRFT=STRAVE3/APOST3
C... INNER SURFACE AVG STRESS
          STRAVEIN = STRAVE/APOST
C... TUBE
      IJ0=0
          DO 44 IJ=1519,98,-49
          APOS = DSQRT((X(1,IJ)-X(1,IJ-49))**2. +
     .     (X(2,IJ)-X(2,IJ-49))**2.)
          APOST = APOS + APOST
      NELIJ=1440-48*IJ0
      IJ0=IJ0+1
          STRAVE = SIGMA(NELIJ)*APOS + STRAVE
44        CONTINUE
      APOST4=APOST-APOST1-APOST2-APOST3
      STRAVE4=STRAVE-STRAVE1-STRAVE2-STRAVE3
      STRTB=STRAVE4/APOST4
C..
      STRAVE = STRAVE/APOST
C..
      WRITE(1,*)'APOST1,APOST2,APOST3,APOST4'
      WRITE(1,*)APOST1,APOST2,APOST3,APOST4
          WRITE(1,*)'STRSH,STRFS,STRFT,STRAVEIN,STRTB,STRAVE'
          WRITE(1,*)STRSH,STRFS,STRFT,STRAVEIN,STRTB,STRAVE
          WRITE(1,*)'DPDZ*DH/4 = ',DPDZ*DH/4.
          WRITE(1,*)'DPDZ*AFLOW = ',DPDZ*AFLOW
          WRITE(1,*)'TAWAVG*PWET = ',STRAVE*PWET
          WRITE(1,261)
          APOS = 0.0
          APOS1 = 0.0
C... SHEATH
      WRITE(1,*)'OVER THE FINNED SURFACE'
      IJ0=0
          DO 45 IJ=1474,347,-49
      NELIJ=1396-48*IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRAVEIN
          APOS1 = DSQRT((X(1,IJ)-X(1,IJ-49))**2.+(X(2,IJ)-X(2,IJ-49))**2.)
          APOS = APOS + APOS1
          WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
45        CONTINUE
C... FIN SIDE
      IJ0=0
          DO 46 IJ=298,314
      NELIJ=292+IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRAVEIN
          APOS1 = DSQRT((X(1,IJ+1)-X(1,IJ))**2.-(X(2,IJ+1)-X(2,IJ))**2.)
          APOS = APOS + APOS1
          WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
46        CONTINUE
C... FIN TIP
      IJ0=0
          DO 47 IJ=315,70,-49
      NELIJ=261-48*IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRAVEIN
          APOS1 = DSQRT((X(1,IJ)-X(1,IJ-49))**2.+(X(2,IJ)-X(2,IJ-49))**2.)
          APOS = APOS + APOS1
          WRITE(1,26) APOS,ARSS,SIGMA(NELIJ)
47        CONTINUE
C... TUBE
      APOS = 0.0
      APOS2 = 0.0
      IJ0=0
      WRITE(1,*)'OVER THE TUBE SURFACE'
          DO 48 IJ=1519,98,-49
      NELIJ=1440-48*IJ0
      IJ0=IJ0+1
          ARSS = SIGMA(NELIJ)/STRTB
          APOS2 = DSQRT((X(1,IJ)-X(1,IJ-49))**2. +
     .     (X(2,IJ)-X(2,IJ-49))**2.)
          APOS = APOS + APOS2
```

```
        WRITE(1,26) APOS,ARSS,SIGMA(NELI)
48      CONTINUE
        ENDIF
        RETURN
        END
C**********************************************************
   SUBROUTINE STRESS(NE,X,NODES,U,IEQ,MAT,PROP,VAR,UELEM,
   .        IMAT,TVAR,SIGMA)
C**********************************************************
C.....
C.....CALCULATES STRESS FROM SHAPE FUNCTIONS
C.  FOR QUADRILATERAL ELEMENTS
C.....
C   CALLED BY:
C
C   CALLS   : SHAPE4
C
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
   COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
   COMMON/CINT/XIQ(9,2,3),WQ(9,3)
   COMMON /PLTOUT/ XG,YG,SIGHX,SIGHY
   COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
   ..NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
C
   INCLUDE 'THVAR.H'
C
   DIMENSION NE(1),MAT(1),NODES(9,1),X(2,1),U(10,1)
   DIMENSION PSI(9),DPSI(9,2),XX(2,9)
   DIMENSION DPSIX(9),DPSIY(9),DXDS(2,2),DSDX(2,2)
   DIMENSION XI(9,2,3)
   DIMENSION SIGMA(1)
   DATA PI,PI2 /3.141592654,1.570796327/
C
C.....CALCULATE U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C
C.....BEGIN INTEGRATION POINT LOOP
C
   DO 11 NEL=1,NELEM
   NN=1
   L=1
   N=NE(NEL)
   DO 15 I=1,N
   XX(1,I)=X(1,NODES(I,NEL))
 15 XX(2,I)=X(2,NODES(I,NEL))
C
   CALL SHAPE4 (XIQ(L,1,NN),XIQ(L,2,NN),N,PSI,DPSI)
C
C.....CALCULATE DXDS
C
   DO 20 I=1,2
   DO 20 J=1,2
   DXDS(I,J)=0.0
   DO 20 K=1,N
 20 DXDS(I,J)=DXDS(I,J)+DPSI(K,J)*XX(I,K)
C
C.....CALCULATE DSDX
```

```
C
      DETJ=DXDS(1,1)*DXDS(2,2)-DXDS(1,2)*DXDS(2,1)
      DSDX(1,1)=DXDS(2,2)/DETJ
      DSDX(2,2)=DXDS(1,1)/DETJ
      DSDX(1,2)=-DXDS(1,2)/DETJ
      DSDX(2,1)=-DXDS(2,1)/DETJ
C
C.....CALCULATE D(PSI)/DX
C
      DO 30 I=1,N
      DPSIX(I)=DPSI(I,1)*DSDX(1,1)+DPSI(I,2)*DSDX(2,1)
   30 DPSIY(I)=DPSI(I,1)*DSDX(1,2)+DPSI(I,2)*DSDX(2,2)
      UH=0.
      DUHDX=0.
      DUHDY=0.
      XG=0.
      YG=0.
      DO 10 I=1,N
      XG=XG+PSI(I)*XX(1,I)
      YG=YG+PSI(I)*XX(2,I)
      UH=UH+PSI(I)*U(IEQ,NODES(I,NEL))
      DUHDX=DUHDX+DPSIX(I)*U(IEQ,NODES(I,NEL))
   10 DUHDY=DUHDY+DPSIY(I)*U(IEQ,NODES(I,NEL))

      CALL GETMAT (XK,YK,XYK,XM,YM,XB,XF,RMU,XRHO1,XRHO2,MAT(NEL),PROP,
     >         UELEM,IMAT,VAR,TVAR,NEL,IEQ)

      SIGHX=-XK*DUHDX
      SIGHY=-YK*DUHDY
      IF(IEQ.EQ.1)SIGMA(NEL) = DSQRT(SIGHX*SIGHX+SIGHY*SIGHY)
      GRADX(IEQ,NEL)=DUHDX
      GRADY(IEQ,NEL)=DUHDY
11    CONTINUE
C
      RETURN
      END
C*****************************************************************
      SUBROUTINE VISCKE (NEL,N,XX,NODES,U,IEQ,ITER,MAT,SIGMA,X,UELEM)
C*****************************************************************
C.....
C.....CALCULATES U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C.  FOR QUADRILATERAL ELEMENTS
C.....
C   CALLED BY:
C
C   CALLS   : SHAPE4
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
     .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/YSPLUS/YPLUSA,SPLUSA,ALLA,YA,SA,DFPA,DFCA,ALPA,ALCA,
     .  TWYA,TWSA
C
      INCLUDE 'THVAR.H'
C
```

```
      DIMENSION XX(2,9),NODES(9,1),U(10,1),SIGMA(2350),X(2,1),
     .     UELEM(10,1)
      DIMENSION YPLUSA(2350)
      LOGICAL FIRST

      CMU=0.09

      IF(KEMODEL.EQ.'MY')THEN
       C1=1.4
       C2=1.8
      ELSEIF(KEMODEL.EQ.'CH')THEN
       C1=1.35
       C2=1.8
      ELSE
       C1=1.44
       C2=1.92
      ENDIF

      REK = DEN*YY(NEL)*DABS(UELEM(3,NEL))**0.5/VIS
      RET = DEN*UELEM(3,NEL)**2./VIS/DABS(UELEM(4,NEL))

      IF(KEMODEL.EQ.'LS')THEN
C... Launder and Sharma (Cho and Goldstein,1994)
       FMUKE(NEL) = DEXP(-3.4/(1.+(RET/50.)**2.))
      ELSEIF(KEMODEL.EQ.'NA')THEN
C... Nagano
       FMUKE(NEL) = (1.-DEXP(-YPLUSA(NEL)/26.5))**2.
      ELSEIF(KEMODEL.EQ.'HE')THEN
C... Herrero (Int. J. Heat Mass Transfer, 1990)
       FMUKE(NEL) = (1.-DEXP(-0.0066*REK))**2.*
     .     (1.+500.*DEXP(-0.0055*REK)/RET)
      ELSEIF(KEMODEL.EQ.'LB')THEN
C... Lam-Bremhorst (1981)
       FMUKE(NEL) = (1.-DEXP(-0.0165*REK))**2.*(1.+20.5/RET)
      ELSEIF(KEMODEL.EQ.'MY')THEN
C... Myong (1990)
       FMUKE(NEL) = (1.+3.45/DSQRT(DABS(RET)))*
     .       (1.-DEXP(-YPLUSA(NEL)/70.))
      ELSEIF(KEMODEL.EQ.'CH')THEN
C... Chien
       FMUKE(NEL) = 1.-DEXP(-0.0115*YPLUSA(NEL))
      ELSE
       FMUKE(NEL)=1.0
      ENDIF

      IF(KEMODEL.EQ.'LS'.OR.KEMODEL.EQ.'NA')THEN
       F1(NEL)=1.
       F2(NEL)=1.-0.3*DEXP(-RET**2.)
      ELSEIF(KEMODEL.EQ.'HE')THEN
       F1(NEL)=1.+(0.05/FMUKE(NEL))**2.
       F2(NEL)=1.-(0.3/(1.-0.7*DEXP(-REK)))*DEXP(-RET**2.)
      ELSEIF(KEMODEL.EQ.'LB')THEN
       F1(NEL)=1.+(0.05/FMUKE(NEL))**2.
       F2(NEL)=1.-DEXP(-RET**2.)
      ELSEIF(KEMODEL.EQ.'MY')THEN
       F1(NEL)=1.
       F2(NEL)=(1.-2./9.*DEXP(-(RET/6.)**2.))*
     .     (1.-DEXP(-YPLUSA(NEL)/5.))**2.
```

```
      ELSEIF(KEMODEL.EQ.'CH')THEN
       F1(NEL)=1.
       F2(NEL)=1.-0.22*DEXP(-(RET/6.)**2.)
      ELSE
       F1(NEL)=1.0
       F2(NEL)=1.0
      ENDIF

      IF (FIRST) THEN
       VISTT(NEL)=AMUST(NEL)
       IF(NEL.EQ.NELEM)FIRST = .FALSE.
      ELSE
       VISTNEW = CMU*FMUKE(NEL)*DEN*UELEM(3,NEL)**2./
     .       DABS(UELEM(4,NEL))
       VISTT(NEL) = VISTNEW*RELAXVIST + VISTT(NEL)*(1.-RELAXVIST)
      ENDIF

      RETURN
      END
C***********************************
         REAL*8 FUNCTION DENF(TIN)
C***********************************

         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         REAL*8 TL_DB(12),DENF_DB(12)
C... TL, C
         DATA TL_DB/-1.E20,20.,40.,50.,60.,70.,
     .     80.,90.,100.,140.,180.,1.E20/
C... DENF, KG/M3
         DATA DENF_DB/998.2,998.2,992.2,988.0,983.2,977.7,
     .      971.8,965.3,958.3,926.1,886.9,886.9/
C
         DO 20 I=1,11
          IF(TIN.GE.TL_DB(I).AND.TIN.LE.TL_DB(I+1))THEN
           DENF=DENF_DB(I)+(DENF_DB(I+1)-DENF_DB(I))
     .    *(TIN-TL_DB(I))/(TL_DB(I+1)-TL_DB(I))
C... RETURNS DENF, KG/M3 FOR GIVEN TB, C
           GO TO 21
          ENDIF
20    CONTINUE
21    CONTINUE
         RETURN
         END
C***********************************
         REAL*8 FUNCTION CPF(TIN)
C***********************************

         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         REAL*8 TL_DB(12),CPF_DB(12)
C... TL, C
         DATA TL_DB/-1.E20,20.,40.,50.,60.,70.,
     .     80.,90.,100.,140.,180.,1.E20/
C... CPF, J/(KG K)
         DATA CPF_DB/4182.,4182.,4179.,4181.,4185.,4190.,
     .      4197.,4205.,4216.,4285.,4408.,4408./
C
         DO 20 I=1,11
          IF(TIN.GE.TL_DB(I).AND.TIN.LE.TL_DB(I+1))THEN
```

```
        CPF=CPF_DB(I)+(CPF_DB(I+1)-CPF_DB(I))
     .   *(TIN-TL_DB(I))/(TL_DB(I+1)-TL_DB(I))
C... RETURNS CPF, J/(KG K) FOR GIVEN TB, C
        GO TO 21
        ENDIF
20    CONTINUE
21    CONTINUE
       RETURN
       END
C***********************************
       REAL*8 FUNCTION VISF(TIN)
C***********************************

       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       REAL*8 TL_DB(12),VISF_DB(12)
C... TL, C
       DATA TL_DB/-1.E20,20.,40.,50.,60.,70.,
     .     80.,90.,100.,140.,180.,1.E20/
C... VISF, PA S
       DATA VISF_DB/10.03E-4,10.03E-4,65.31E-5,54.71E-5,46.68E-5.
     . 40.44E-5,35.49E-5,31.50E-5,28.22E-5,19.61E-5,14.94E-5,14.94E-5/
C
       DO 20 I=1,11
        IF(TIN.GE.TL_DB(I).AND.TIN.LE.TL_DB(I+1))THEN
        VISF=VISF_DB(I)+(VISF_DB(I+1)-VISF_DB(I))
     .   *(TIN-TL_DB(I))/(TL_DB(I+1)-TL_DB(I))
C... RETURNS VISF, PA S FOR GIVEN TB, C
        GO TO 21
        ENDIF
20    CONTINUE
21    CONTINUE
       RETURN
       END
C***********************************
       REAL*8 FUNCTION AKF(TIN)
C***********************************

       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       REAL*8 TL_DB(12),AKF_DB(12)
C... TL, C
       DATA TL_DB/-1.E20,20.,40.,50.,60.,70.,
     .   80.,90.,100.,140.,180.,1.E20/
C... AKF, W/(M K)
       DATA AKF_DB/.6,.6,.629,.64,.651,.659,
     .   .667,.673,.677,.685,.674,.674/
C
       DO 20 I=1,11
        IF(TIN.GE.TL_DB(I).AND.TIN.LE.TL_DB(I+1))THEN
        AKF=AKF_DB(I)+(AKF_DB(I+1)-AKF_DB(I))
     .   *(TIN-TL_DB(I))/(TL_DB(I+1)-TL_DB(I))
C... RETURNS AKF, W/(M K) FOR GIVEN TB, C
        GO TO 21
        ENDIF
20    CONTINUE
21    CONTINUE
       RETURN
       END
C******************************************************************
```

```
      SUBROUTINE VISC1 (NEL,N,XX,NODES,U,IEQ,ITER,MAT,SIGMA,X,UELEM)
C*****************************************************************
C.....
C.....CALCULATES U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C.  FOR QUADRILATERAL ELEMENTS
C.....
C   CALLED BY:
C
C   CALLS   : SHAPE4
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CINT/XIQ(9,2,3),WQ(9,3)
      COMMON /PLTOUT/ XG,YG,SIGHX,SIGHY
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
     ..NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
      COMMON/RM_UMAX/IRM,RM_CAL,RMKAY,UMAX,CKARMANI,IRMA,RM_CALA,UMAXA
      COMMON/YSPLUS/YPLUSA,SPLUSA,ALLA,YA,SA,DFPA,DFCA,ALPA,ALCA,
     .  TWYA,TWSA
      COMMON/ELGRID/XG_EL,YG_EL
C
      INCLUDE 'THVAR.H'
C
      DIMENSION NODES(9,1)
      DIMENSION U(10,1),NE(1)
      DIMENSION UELEM(10,1)
      DIMENSION PSI(9),DPSI(9,2),XX(2,9)
      DIMENSION DPSIX(9),DPSIY(9),DXDS(2,2),DSDX(2,2)
      DIMENSION XI(9,2,3)
      DIMENSION SIGMA(2350)
      DIMENSION YPLUSA(2350),SPLUSA(2350),ALLA(2350),YA(2350),SA(2350),
     .DFPA(2350),DFCA(2350),ALPA(2350),ALCA(2350),TWYA(2350),TWSA(2350)
      DIMENSION X(2,1)
      DIMENSION XG_EL(2350),YG_EL(2350)
      DIMENSION IRMA(50),RM_CALA(50),UMAXA(50)
C
      DATA PI,PI2  /3.141592654,1.570796327/
C
C.....INITIALIZE
C
      ALP=0.0
      ALC=0.0
      Y=0.0
      S=0.0
      DFP=0.0
      DFC=0.0
      YPLUS=0.0
      SPLUS=0.0
      ALL=0.0
      TWY = 0.0
      TWS = 0.0
C
C.....CALCULATE U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C
C.....BEGIN INTEGRATION POINT LOOP
```

```
C
   NN=1
   L=1
C
   CALL SHAPE4 (XIQ(L,1,NN),XIQ(L,2,NN),N,PSI,DPSI)
C
C.....CALCULATE DXDS
C
   DO 20 I=1,2
   DO 20 J=1,2
   DXDS(I,J)=0.0
   DO 20 K=1,N
 20 DXDS(I,J)=DXDS(I,J)+DPSI(K,J)*XX(I,K)
C
C.....CALCULATE DSDX
C
   DETJ=DXDS(1,1)*DXDS(2,2)-DXDS(1,2)*DXDS(2,1)
   DSDX(1,1)=DXDS(2,2)/DETJ
   DSDX(2,2)=DXDS(1,1)/DETJ
   DSDX(1,2)=-DXDS(1,2)/DETJ
   DSDX(2,1)=-DXDS(2,1)/DETJ
C
C.....CALCULATE D(PSI)/DX
C
   DO 30 I=1,N
   DPSIX(I)=DPSI(I,1)*DSDX(1,1)+DPSI(I,2)*DSDX(2,1)
 30 DPSIY(I)=DPSI(I,1)*DSDX(1,2)+DPSI(I,2)*DSDX(2,2)
   UH=0.
   DUHDX=0.
   DUHDY=0.
   XG=0.
   YG=0.
   DO 10 I=1,N
   XG=XG+PSI(I)*XX(1,I)
   YG=YG+PSI(I)*XX(2,I)
   UH=UH+PSI(I)*U(IEQ,NODES(I,NEL))
   DUHDX=DUHDX+DPSIX(I)*U(IEQ,NODES(I,NEL))
 10 DUHDY=DUHDY+DPSIY(I)*U(IEQ,NODES(I,NEL))
C
   IF (MAT .EQ. 2) THEN
C...
   THETANRAD = THETAN*PI/180.0
         IF(FPROP.EQ.'FIXED')THEN
     ANU  = VIS/DEN
         ELSEIF (FPROP.EQ.'FIXTB')THEN
     ANU  = VISF(TAVE)/DENF(TAVE)
         ELSE
     ANU  = VISF(UELEM(2,NEL))/DENF(UELEM(2,NEL))
         ENDIF
   R = DSQRT(XG*XG + YG*YG)
C...
   IF (NGEOMTYPE.EQ.1) NFEL = 0
C
C... TURBULENT FLOW IN ANNULUS OR FINNED ANNULUS (FA8*)
C
   IF(NGEOMTYPE.EQ.1)THEN
C
C.. DETERMINE RM AT WHICH UMAX OCCURS
```

```
C*************
C... ANNULUS*
C*************
    IF (NGEOMTYPE.EQ.1) THEN
         IF (NEL.EQ.2)THEN
     UMAX = 0.
     DO 59 I = 1,NELEM
     IF (UELEM(1,I) .GT. UMAX) THEN
      UMAX = UELEM(1,I)
      IRM = I
     ENDIF
59     CONTINUE
         ENDIF
     RM=DSQRT(X(1,IRM)**2.+X(2,IRM)**2.)
         IF(ITER.GT.2)RM = DMAX1 (RM_CAL, RM)
     RM_CAL = RM
C...
     TURVIS1 = 0.
     TURVIS2 = 0.
     IF (R .LE. RM)THEN
      DO 598 I = 1,IRM
      IF(AMUST(I) .GT. TURVIS1) THEN
       TURVIS1 = AMUST(I)
       IVIS1 = I
          ENDIF
598     CONTINUE
     ELSE
      DO 599 I = IRM,NELEM
          IF (AMUST(I) .GT. TURVIS2) THEN
       TURVIS2 = AMUST(I)
       IVIS2 = I
          ENDIF
599     CONTINUE
     ENDIF
    ENDIF
C
C...USE RM OF KAYS
C
    RMKAY = RI*(1.+(RO/RI)**0.657)/(1.+(RI/RO)**0.343)
        IF(RMOPT.EQ.'RMKAYS')THEN
     RM=RMKAY
        ELSEIF(RMOPT.EQ.'RMUSER')THEN
         RM=RMVALUE
        ENDIF
    Y0M = RO - RM
C
C...
C
C*************
C... ANNULUS*
C*************
    IF (NGEOMTYPE.EQ.1) THEN
     Y = R - RI
     YM = RM - RI
    ENDIF
C...
C...
C...
```

```
      IF (R .LE. RM) THEN
C
       IF (NEL .GT. NFEL) THEN
        RREF = RI
       ELSE
        RREF = RI + FHT
       ENDIF
       RR = RO/RREF
       OM = RM/RREF
C
            IF(KARMANOPT.EQ.'KVAR')THEN
        CKARMANI = 0.4*(RR-OM)/(OM-1.)*
     .       DSQRT((RR**2.-OM**2.)/RR/(OM**2.-1.))
            ELSE
             CKARMANI =  CKARMAN
            ENDIF
       B1 = 0.14*(RR-OM)/(OM-1.)
       B2 = 2.*B1-0.5*CKARMANI
       B3 = 0.5*CKARMANI-B1
       ALP = YM*(B1-B2*(1.-Y/YM)**2.-B3*(1.-Y/YM)**4.)
C
      ELSE
C
       IF(KARMANOPT.EQ.'KFIX')THEN
C        CKARMANO = CKARMAN
        CKARMANO = 0.4
       ELSE
        CKARMANO = 0.4
       ENDIF
       AA1 = 0.14
       AA2 = 2.*AA1-0.5*CKARMANO
       AA3 = 0.5*CKARMANO-AA1
       ALP =Y0M*(AA1-AA2*(Y-YM)**2./Y0M**2.-AA3*(Y-YM)**4./Y0M**4.)
C
      ENDIF
     ENDIF
C...
           TWY = 0.0
           TWS = 0.0
C...
     IF (ITER .GT. 1) THEN
C************
C... ANNULUS*
C************
     IF (NGEOMTYPE.EQ.1) THEN
           TWS = 0.0
      IF (R.LE.RM)THEN
            TWY = SIGMA(2)
            TWYI=TWY
      ELSE
       TWY = SIGMA(NELEM)
      ENDIF
     ENDIF
     ELSE
     TWY = -1.0*DPDZ*DH/4.0
     TWS = -1.0*DPDZ*DH/4.0
C... END OF IF(ITER.GT.1)
     ENDIF
```

```
 C************
 C... ANNULUS*
 C************
     IF (NGEOMTYPE .EQ. 1) THEN
      IF (NEL .GT. NFEL) THEN
       IF(R.LE.RM)THEN
        Y=R-RI
       ELSE
        Y=RO-R
       ENDIF
      ELSE
       IF(R.LE.RM)THEN
        Y=R-RI-FHT
       ELSE
        Y=RO-R
       ENDIF
      ENDIF
    ENDIF
C...
    IF(FPROP.EQ.'FIXED')THEN
     YPLUS = Y*DSQRT(TWY/DEN)/ANU
     SPLUS = S*DSQRT(TWS/DEN)/ANU
    ELSEIF (FPROP.EQ.'FIXTB')THEN
     YPLUS = Y*DSQRT(TWY/DENF(TAVE))/ANU
     SPLUS = S*DSQRT(TWS/DENF(TAVE))/ANU
    ELSE
     YPLUS = Y*DSQRT(TWY/DENF(UELEM(2,NEL)))/ANU
     SPLUS = S*DSQRT(TWS/DENF(UELEM(2,NEL)))/ANU
    ENDIF
C
    DFP = 1.0 - DEXP(-1.0*YPLUS/APLUS)
    DFC = 1.0 - DEXP(-1.0*SPLUS/APLUS)
C
    IF(KARMANOPT.EQ.'KFIX')THEN
     CKARMANFS = 0.4
C     CKARMANFS = CKARMAN
    ELSE
     CKARMANFS = 0.4
    ENDIF
    A2 = 2.*A1-0.5*CKARMANFS
    A3 = 0.5*CKARMANFS-A1
    ALC = SO*(A1-A2*((1.0-S/SO)**2.0)-A3*((1.0-S/SO)**4.0))
    ALP = DFP*ALP
    ALC = DFC*ALC
C************
C... ANNULUS*
C************
    IF (NGEOMTYPE .EQ. 1) THEN
     ALL = ALP
    ELSE
     ALL = (ALC*ALP)/(ALC+ALP)
    ENDIF
C
    ALLA(NEL)=ALL
C
    UGRAD = DSQRT(DUHDX*DUHDX+DUHDY*DUHDY)
C...
C...
```

```
      IF (MIXMODEL .EQ. 'M4' .AND. ITER.GT.1 .AND.
     .      NGEOMTYPE .EQ. 1) THEN
C
       IF(FPROP.EQ.'FIXED')THEN
        USTAR = (TWYI/DEN)**0.5
       ELSEIF (FPROP.EQ.'FIXTB')THEN
        USTAR = (TWYI/DENF(TAVE))**0.5
       ELSE
        USTAR = (TWYI/DENF(UELEM(2,NEL)))**0.5
       ENDIF
       ALLRM = 0.14*Y0M
       UGRADMIN=CKARMANI/6.*(RM-RREF)*USTAR/ALLRM**2.
       UGRADMIN=DMAX1(UGRAD,UGRADMIN)
      ENDIF
C...
C...
      IF(FPROP.EQ.'FIXED')THEN
       AMUT = DEN*ALLA(NEL)*ALLA(NEL)*UGRAD
      ELSEIF (FPROP.EQ.'FIXTB')THEN
       AMUT = DENF(TAVE)*ALLA(NEL)*ALLA(NEL)*UGRAD
      ELSE
       AMUT = DENF(UELEM(2,NEL))*ALLA(NEL)*ALLA(NEL)*UGRAD
      ENDIF
      AMU = AMUT
C...
C... DEISSER AND REICHARDT
C...
      IF (MIXMODEL .EQ. 'M2' .AND. ITER.GT.1 .AND.
     .      (NGEOMTYPE .EQ. 1.OR.NGEOMTYPE .EQ. 21)) THEN
       IF (R .LE. RM)THEN
C         ETAI=(RM-R)/(RM-RI)
C         ETAPLUSI=1.5*YPLUS*(1.+ETAI)/(1.+2.*ETAI**2.)
C     IF (ETAPLUSI .LE. 26.)THEN
       IF (YPLUS .LE. 26.)THEN
          USTAR = (SIGMA(2)/DEN)**0.5
        UEL = (U(1,NEL)+U(1,NEL+1))/2.
        UPLUS = UEL/USTAR
C         AMU=VIS*0.0154*UPLUS*YPLUS*(1-DEXP(-0.0154*UPLUS*YPLUS))
          AMU=AMUT
         ELSE
          USTAR = (SIGMA(2)/DEN)**0.5
        UEL = (U(1,NEL)+U(1,NEL+1))/2.
        UPLUS = UEL/USTAR
          AMU=DEN*CKARMANI/6.*(RM-RI)*USTAR*(1.-((RM-R)/(RM-RI))**2.)*
     .      (1.+2.*(RM-R)/(RM-RI))
         ENDIF
       ELSE
C         ETAO=(RM-R)/(RM-RO)
C         ETAPLUSO=1.5*YPLUS*(1.+ETAO)/(1.+2.*ETAO**2.)
C     IF (ETAPLUSO .LE. 42.)THEN
       IF (YPLUS .LE. 26.)THEN
          USTAR = (SIGMA(NELEM)/DEN)**0.5
        UEL = (U(1,NEL)+U(1,NEL+1))/2.
        UPLUS = UEL/USTAR
C         AMU=VIS*0.0154*UPLUS*YPLUS*(1-DEXP(-0.0154*UPLUS*YPLUS))
          AMU=AMUT
         ELSE
          USTAR = (SIGMA(NELEM)/DEN)**0.5
       UEL = (U(1,NEL)+U(1,NEL+1))/2.
       UPLUS = UEL/USTAR
           AMU=DEN*CKARMANO/6.*(RO-RM)*USTAR*(1.-((R-RM)/(RO-RM))**2.)*
    .        (1.+2.*(R-RM)/(RO-RM))
          ENDIF
      ENDIF
     ENDIF
C...
     ELSE
     AMU = 0.0
C... END OF IF(MAT.EQ.2)
     ENDIF
C...
C... DEISSER AND REICHARDT
C...
     IF (MIXMODEL .EQ. 'M3' .AND. ITER.GT.1 .AND.
    .    (NGEOMTYPE .EQ. 1.OR.NGEOMTYPE .EQ. 21)) THEN
      USTAR = (SIGMA(2)/DEN)**0.5
      AMUMIN=DEN*CKARMANI/6.*(RM-RI)*USTAR
      IF (NEL .GE. IVIS1 .AND. NEL .LE. IRM)THEN
          AMU=DMAX1(AMU,AMUMIN)
      ELSEIF (NEL .GT. IRM .AND. NEL .LE. IVIS2)THEN
          AMU=DMAX1(AMU,AMUMIN)
      ENDIF
C      WRITE(2,*)'NEL,IVIS1,IVIS2,IRM,AMUMIN,AMU'
C      WRITE(2,121)NEL,IVIS1,IVIS2,IRM,AMUMIN,AMU
C121   FORMAT(1X,4I4,16(1X,1PE11.4))
     ENDIF
C
C... CALCULATE TURBULENT PRANDTL NUMBER
C
     IF(FPROP.EQ.'FIXED')THEN
      VISTEMP = VIS
      CPTEMP = CP
      AKTEMP = AK
      PR = VIS*CP/AK
     ELSEIF (FPROP.EQ.'FIXTB')THEN
      VISTEMP = VISF(TAVE)
      CPTEMP = CPF(TAVE)
      AKTEMP = AKF(TAVE)
      PR = VISTEMP*CPTEMP/AKTEMP
     ELSE
      VISTEMP = VISF(UELEM(2,NEL))
      CPTEMP = CPF(UELEM(2,NEL))
      AKTEMP = AKF(UELEM(2,NEL))
      PR = VISTEMP*CPTEMP/AKTEMP
     ENDIF
     IF(IPRT.EQ.0)THEN
       PRT(NEL)=PRT0
     ELSEIF(IPRT.EQ.1)THEN
       PET=AMU/VISTEMP*PR
       PRT(NEL)=2./PET+0.85
     ELSEIF(IPRT.EQ.2)THEN
       PRT(NEL)=1./(0.5882+0.228*(AMU/VISTEMP)-
    .        0.0441*(AMU/VISTEMP)**2.*
    .        (1-DEXP(-5.165/(AMU/VISTEMP))))
     ELSEIF(IPRT.EQ.3)THEN
      IF (NEL .GT. NFEL) THEN
```

```
        PRT(NEL)=1+0.855-DTANH(0.2*(YPLUS-7.5))
       ELSE
        IF (YPLUS.GT5.0 .OR. SPLUS.GT5.0)THEN
         PRTY=1+0.855-DTANH(0.2*(YPLUS-7.5))
         PRTS=1+0.855-DTANH(0.2*(SPLUS-7.5))
         PRT(NEL)=PRTY*PRTS/(PRTY+PRTS)
        ELSE
         PRT(NEL)=1.0
        ENDIF
       ENDIF
      ELSE
        WRITE(NLG,*)'SPECIFY IPRT (TURBULENT PRANDTL NUMBER OPTION)'
      ENDIF
C...
      ALPA(NEL)=ALP
      ALCA(NEL)=ALC
      YA(NEL)=Y
      SA(NEL)=S
      DFPA(NEL)=DFP
      DFCA(NEL)=DFC
      YPLUSA(NEL)=YPLUS
      SPLUSA(NEL)=SPLUS
      TWSA(NEL)=TWS
      TWYA(NEL)=TWY
      XG_EL(NEL)=XG
      YG_EL(NEL)=YG
C
      AMUST(NEL) = AMU
C
C... CALCULATE DISTANCE FROM THE WALL
      IF (NEL .GT. NFEL) THEN
        YY(NEL)=Y
      ELSE
        YY(NEL)=Y*S/(Y+S)
      ENDIF
C...
      RETURN
      END
C**********************************************************************
      SUBROUTINE VISC2122 (NEL,N,XX,NODES,U,IEQ,ITER,MAT,SIGMA,X,UELEM)
C**********************************************************************
C... FA8*
C********
C.....
C.....CALCULATES U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C.  FOR QUADRILATERAL ELEMENTS
C.....
C   CALLED BY:
C
C   CALLS   : SHAPE4
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/FILES/NIN,NOU,NLG,NFILE,NPLOT
      COMMON/CINT/XIQ(9,2,3),WQ(9,3)
      COMMON /PLTOUT/ XG,YG,SIGHX,SIGHY
      COMMON/CCON/NNODE,NELEM,NMAT,NPOINT,NOUT,NINTO
```

```
     .,NPRNT1,NPRNT2,NPRNT3,NPRNT4,NPTYPE,NPDE
      COMMON/TIMES/T0,TF,DELTAT,NSTEP,NSTEPT
      COMMON/RM_UMAX/IRM,RM_CAL,RMKAY,UMAX,CKARMANI,IRMA,RM_CALA,UMAXA
      COMMON/YSPLUS/YPLUSA,SPLUSA,ALLA,YA,SA,DFPA,DFCA,ALPA,ALCA,
     .  TWYA,TWSA
      COMMON/ELGRID/XG_EL,YG_EL
C
      INCLUDE 'THVAR.H'
C
      DIMENSION NODES(9,1)
      DIMENSION U(10,1),NE(1)
      DIMENSION UELEM(10,1)
      DIMENSION PSI(9),DPSI(9,2),XX(2,9)
      DIMENSION DPSIX(9),DPSIY(9),DXDS(2,2),DSDX(2,2)
      DIMENSION XI(9,2,3)
      DIMENSION SIGMA(1550)
      DIMENSION YPLUSA(1550),SPLUSA(1550),ALLA(1550),YA(1550),SA(1550),
     .DFPA(1550),DFCA(1550),ALPA(1550),ALCA(1550),TWYA(1550),TWSA(1550)
      DIMENSION X(2,1)
      DIMENSION XG_EL(1550),YG_EL(1550)
      DIMENSION IRMA(50),RM_CALA(50),UMAXA(50)
C
      DATA PI,PI2 /3.141592654,1.570796327/
C
C.....INITIALIZE
C
      ALP=0.0
      ALC=0.0
      Y=0.0
      S=0.0
      DFP=0.0
      DFC=0.0
      YPLUS=0.0
      SPLUS=0.0
      ALL=0.0
      TWY = 0.0
      TWS = 0.0
C
C.....CALCULATE U, SIG-X (QX), AND SIG-Y (QY) FROM SHAPE FUNCTIONS
C
C.....BEGIN INTEGRATION POINT LOOP
C
      NN=1
      L=1
C
      CALL SHAPE4 (XIQ(L,1,NN),XIQ(L,2,NN),N,PSI,DPSI)
C
C.....CALCULATE DXDS
C
      DO 20 I=1,2
      DO 20 J=1,2
      DXDS(I,J)=0.0
      DO 20 K=1,N
   20 DXDS(I,J)=DXDS(I,J)+DPSI(K,J)*XX(I,K)
C
C.....CALCULATE DSDX
C
      DETJ=DXDS(1,1)*DXDS(2,2)-DXDS(1,2)*DXDS(2,1)
      DSDX(1,1)=DXDS(2,2)/DETJ
      DSDX(2,2)=DXDS(1,1)/DETJ
      DSDX(1,2)=-DXDS(1,2)/DETJ
      DSDX(2,1)=-DXDS(2,1)/DETJ
C
C.....CALCULATE D(PSI)/DX
C
      DO 30 I=1,N
      DPSIX(I)=DPSI(I,1)*DSDX(1,1)+DPSI(I,2)*DSDX(2,1)
   30 DPSIY(I)=DPSI(I,1)*DSDX(1,2)+DPSI(I,2)*DSDX(2,2)
      UH=0.
      DUHDX=0.
      DUHDY=0.
      XG=0.
      YG=0.
      DO 10 I=1,N
      XG=XG+PSI(I)*XX(1,I)
      YG=YG+PSI(I)*XX(2,I)
      UH=UH+PSI(I)*U(IEQ,NODES(I,NEL))
      DUHDX=DUHDX+DPSIX(I)*U(IEQ,NODES(I,NEL))
   10 DUHDY=DUHDY+DPSIY(I)*U(IEQ,NODES(I,NEL))
C
      IF (MAT .EQ. 2) THEN
C...
      THETANRAD = THETAN*PI/180.0
             IF(FPROP.EQ.'FIXED')THEN
        ANU  = VIS/DEN
             ELSEIF (FPROP.EQ.'FIXTB')THEN
        ANU  = VISF(TAVE)/DENF(TAVE)
             ELSE
        ANU  = VISF(UELEM(2,NEL))/DENF(UELEM(2,NEL))
             ENDIF
      R = DSQRT(XG*XG + YG*YG)
C...
      IF (NGEOMTYPE.EQ.21) NFEL =288
      IF (NGEOMTYPE.EQ.22) NFEL = 0
C
      IF(NGEOMTYPE.EQ.21.OR.NGEOMTYPE.EQ.22)THEN
C
C.. DETERMINE RM AT WHICH UMAX OCCURS
C... rm on the radial line
        IF(RMOPT.EQ.'RMCALL')THEN
         DO 57 I = 1,30
         UMAX=0.0
         JSTART=1+(I-1)*49
         JEND=49+(I-1)*49
         DO 577 J = JSTART,JEND
         IF (U(1,J) .GT. UMAX) THEN
          UMAX = U(1,J)
          IRM = J
         ENDIF
577        CONTINUE
         IF(NEL.GE.(1+48*(I-1)) .AND. NEL.LE.(48+48*(I-1)))THEN
          RM=(X(1,IRM)**2.+X(2,IRM)**2.)**0.5
               IF(ITER.GT.2)RM = DMAX1 (RM_CALA(I), RM)
          RM_CALA(I) = RM
          IRMA(I)=IRM
          UMAXA(I)=UMAX
        GO TO 5777
       ENDIF
57      CONTINUE
5777    CONTINUE
       RM_CAL = RM
C... rm at a single point
            ELSEIF(RMOPT.EQ.'RMCALP')THEN
       UMAX = 0.
       DO 567 I = 1,NNODE
       IF (U(1,I) .GT. UMAX) THEN
        UMAX = U(1,I)
        IRM = I
       ENDIF
567     CONTINUE
       RM=(X(1,IRM)**2.+X(2,IRM)**2.)**0.5
       RM_CAL = RM
            ENDIF
C
C...USE RM OF KAYS
C
      RMKAY = RI*(1.+(RO/RI)**0.657)/(1.+(RI/RO)**0.343)
           IF(RMOPT.EQ.'RMKAYS')THEN
      RM=RMKAY
           ELSEIF(RMOPT.EQ.'RMUSER')THEN
            RM=RMVALUE
           ENDIF
C
      Y0M = RO - RM
C
      IF (NGEOMTYPE.EQ.21) THEN
       IF (NEL .GT. NFEL) THEN
       Y = R - RI
       YM = RM - RI
       ELSE
       Y = XG - RI - FHT
       YM = RM - RI - FHT
       ENDIF
C
      ELSEIF (NGEOMTYPE.EQ.22) THEN
       Y = R - RI
       YM = RM - RI
      ENDIF
C...
      IF (R .LE. RM) THEN
C
       IF (NEL .GT. NFEL) THEN
        RREF = RI
       ELSE
        RREF = RI + FHT
       ENDIF
       RR = RO/RREF
       OM = RM/RREF
C
            IF(KARMANOPT.EQ.'KVAR')THEN
        CKARMANI = 0.4*(RR-OM)/(OM-1.)*
     .       DSQRT((RR**2.-OM**2.)/RR/(OM**2.-1.))
            ELSE
             CKARMANI =  CKARMAN
```

```
          ENDIF
       B1 = 0.14*(RR-OM)/(OM-1.)
       B2 = 2.*B1-0.5*CKARMANI
       B3 = 0.5*CKARMANI-B1
       ALP = YM*(B1-B2*(1.-Y/YM)**2.-B3*(1.-Y/YM)**4.)
C
      ELSEIF (R .GT. RM) THEN
       IF(KARMANOPT.EQ.'KFIX')THEN
        CKARMANO = 0.4
       ELSE
        CKARMANO = 0.4
       ENDIF
       AA1 = 0.14
       AA2 = 2.*AA1-0.5*CKARMANO
       AA3 = 0.5*CKARMANO-AA1
       ALP =Y0M*(AA1-AA2*(Y-YM)**2./Y0M**2.-AA3*(Y-YM)**4./Y0M**4.)
      ENDIF
     ENDIF
C...
     TWY = 0.0
     TWS = 0.0
C...
     IF (ITER .GT. 1) THEN
C**************************
C... FA8 UNFINNED ANNULUS*
C**************************
     IF (NGEOMTYPE.EQ.22) THEN
      ELINC=48
      NODINC=49
          TWS = 0.0
C
          NN=1440/ELINC
          DO 219 I=1,NN
          IM1=I-1
          IF (NEL .GE. (1+IM1*ELINC) .AND. NEL .LE. (48+IM1*ELINC)) THEN
       IF(R.LE.RM) THEN
        TWY = SIGMA(4+IM1*ELINC)
           TWYI = TWY
       ELSE
        TWY = SIGMA(48+IM1*ELINC)
       ENDIF
          GO TO 221
      ENDIF
219   CONTINUE
     ENDIF
221  CONTINUE
C**************
C... FA8*.INP*
C**************
     IF (NGEOMTYPE.EQ.21) THEN
          NN=1440/48
          DO 111 I=1,NN
          IM1=I-1
          IF (NEL.GE.(1+IM1*48).AND.NEL.LE.(48+IM1*48))THEN
       IF(R.LE.RM) THEN
C... BASED ON FIN TIP
        TWY = SIGMA(21+IM1*48)
           TWYI = TWY
     ELSE
C... BASED ON TUBE SURFACE
      TWY = SIGMA(48+IM1*48)
     ENDIF
          GO TO 112
          ELSEIF(NEL.GE.(289+IM1*48).AND.NEL.LE.(336+IM1*48))THEN
     IF(R.LE.RM) THEN
C... BASED ON SHEATH SURFACE
      TWY = SIGMA(292+IM1*48)
          TWYI = TWY
     ELSE
C... BASED ON TUBE SURFACE
      TWY = SIGMA(336+IM1*48)
     ENDIF
          GO TO 112
         ELSE
          GO TO 111
         ENDIF
111      CONTINUE
112      CONTINUE
C
         DO 113 IJ = 0,16
         DO 113 IN = 292+IJ,1396+IJ,48
          IF (NEL .EQ. IN) THEN
      TWS = SIGMA(292+IJ)
      X_RFIN = (X(1,298+IJ)+X(1,299+IJ))/2.
      Y_RFIN = (X(2,298+IJ)+X(2,299+IJ))/2.
      RFIN = DSQRT(X_RFIN**2+Y_RFIN**2)
      RFINRAD = DATAN(Y_RFIN/X_RFIN)
      GO TO 114
     ENDIF
113      CONTINUE
114      CONTINUE
C
   ENDIF

   ELSE
    TWY = -1.0*DPDZ*DH/4.0
    TWS = -1.0*DPDZ*DH/4.0
C... END OF IF(ITER.GT.1)
   ENDIF
C
C...DETERMINE S AND SO
C
   IF(NGEOMTYPE .EQ. 21)THEN
    S = R*DATAN(YG/XG) - RFIN*RFINRAD
    SO = R*THETANRAD - RFIN*RFINRAD
   ELSEIF(NGEOMTYPE .EQ. 22)THEN
    S = 0.0
    SO = R*THETANRAD
   ENDIF
C*************
C... FA8*.INP*
C*************
   IF (NGEOMTYPE .EQ. 21) THEN
    IF (NEL .GT. NFEL) THEN
     IF(R.LE.RM)THEN
      Y=R-RI
```

```
        ELSE
         Y=RO-R
        ENDIF
       ELSE
        IF(R.LE.RM)THEN
         Y=XG-RI-FHT
        ELSE
         Y=RO-R
        ENDIF
       ENDIF
      ENDIF
C*************************
C... FA8 UNFINNED ANNULUS*
C*************************
      IF (NGEOMTYPE .EQ. 22) THEN
        IF(R.LE.RM)THEN
         Y=R-RI
        ELSE
         Y=RO-R
        ENDIF
      ENDIF
C...
      IF(FPROP.EQ.'FIXED')THEN
       YPLUS = Y*DSQRT(TWY/DEN)/ANU
       SPLUS = S*DSQRT(TWS/DEN)/ANU
      ELSEIF (FPROP.EQ.'FIXTB')THEN
       YPLUS = Y*DSQRT(TWY/DENF(TAVE))/ANU
       SPLUS = S*DSQRT(TWS/DENF(TAVE))/ANU
      ELSE
       YPLUS = Y*DSQRT(TWY/DENF(UELEM(2,NEL)))/ANU
       SPLUS = S*DSQRT(TWS/DENF(UELEM(2,NEL)))/ANU
      ENDIF
C
      DFP = 1.0 - DEXP(-1.0*YPLUS/APLUS)
      DFC = 1.0 - DEXP(-1.0*SPLUS/APLUS)
C
      IF(KARMANOPT.EQ.'KFIX')THEN
       CKARMANFS = 0.4
      ELSE
       CKARMANFS = 0.4
      ENDIF
      A2 = 2.*A1-0.5*CKARMANFS
      A3 = 0.5*CKARMANFS-A1
      ALC = SO*(A1-A2*((1.0-S/SO)**2.0)-A3*((1.0-S/SO)**4.0))
      ALP = DFP*ALP
      ALC = DFC*ALC
C
      IF (NGEOMTYPE .EQ. 21) THEN
            DO 118 I = 1,30
             IF (NEL .GE. 21+48*(I-1) .AND. NEL .LE. 48+48*(I-1)) THEN
        ALL = ALP
        GO TO 119
       ENDIF
118         CONTINUE
            IF(FSOPT.EQ.'FSON')THEN
       ALL = (ALC*ALP)/(ALC+ALP)
            ELSE
             ALL =ALP
```

```
          ENDIF
C*************************
C... FA8 UNFINNED ANNULUS*
C*************************
     ELSEIF (NGEOMTYPE .EQ. 22) THEN
      ALL = ALP
     ELSE
      ALL = (ALC*ALP)/(ALC+ALP)
     ENDIF
119       CONTINUE
C********
C... FA8*
C********
     ALLA(NEL)=ALL
     IF(NGEOMTYPE.EQ.21)THEN
      ELINC=48
      JSMPT=312
      DO 929 I=1,24
      DO 929 J=309,JSMPT
      DIST=DSQRT((XG_EL((JSMPT+1)+(I-1)*ELINC)-XG_EL(308+(I-1)*ELINC))
    .  **2+(YG_EL((JSMPT+1)+(I-1)*ELINC)-YG_EL(308+(I-1)*ELINC))**2)
      DIST1=DSQRT((XG_EL((JSMPT+1)+(I-1)*ELINC)-XG_EL(J+(I-1)*ELINC))
    .  **2+(YG_EL((JSMPT+1)+(I-1)*ELINC)-YG_EL(J+(I-1)*ELINC))**2)
      ALLA(J+(I-1)*ELINC)=10.**((DLOG10(ALLA((JSMPT+1)+(I-1)*ELINC))-
    .      DLOG10(ALLA(308+(I-1)*ELINC)))*(DIST-DIST1)/DIST+
    .      DLOG10(ALLA(308+(I-1)*ELINC)))
929   CONTINUE
     ENDIF
C
     UGRAD = DSQRT(DUHDX*DUHDX+DUHDY*DUHDY)
C...
     IF (MIXMODEL .EQ. 'M4' .AND. ITER.GT.1 .AND.
    .      (NGEOMTYPE .EQ. 21.OR.NGEOMTYPE .EQ. 22)) THEN
C
      IF(FPROP.EQ.'FIXED')THEN
       USTAR = (TWYI/DEN)**0.5
      ELSEIF (FPROP.EQ.'FIXTB')THEN
       USTAR = (TWYI/DENF(TAVE))**0.5
      ELSE
       USTAR = (TWYI/DENF(UELEM(2,NEL)))**0.5
      ENDIF
      ALLRM = 0.14*Y0M
      UGRADMIN=CKARMANI/6.*(RM-RREF)*USTAR/ALLRM**2.
      UGRAD=DMAX1(UGRAD,UGRADMIN)
     ENDIF
C...
     IF(FPROP.EQ.'FIXED')THEN
      AMUT = DEN*ALLA(NEL)*ALLA(NEL)*UGRAD
     ELSEIF (FPROP.EQ.'FIXTB')THEN
      AMUT = DENF(TAVE)*ALLA(NEL)*ALLA(NEL)*UGRAD
     ELSE
      AMUT = DENF(UELEM(2,NEL))*ALLA(NEL)*ALLA(NEL)*UGRAD
     ENDIF
C
     AMU = AMUT
C...
C... DEISSER AND REICHARDT
C...
```

```
      IF (MIXMODEL .EQ. 'M2' .AND. ITER.GT.1 .AND.
     .      (NGEOMTYPE .EQ. 21.OR.NGEOMTYPE .EQ. 22)) THEN
       IF (R .LE. RM)THEN
C              ETAI=(RM-R)/(RM-RI)
C              ETAPLUSI=1.5*YPLUS*(1.+ETAI)/(1.+2.*ETAI**2.)
C         IF (ETAPLUSI .LE. 26.)THEN
        IF (YPLUS .LE. 26.)THEN
                USTAR = (SIGMA(2)/DEN)**0.5
         UEL = (U(1,NEL)+U(1,NEL+1))/2.
         UPLUS = UEL/USTAR
C               AMU=VIS*0.0154*UPLUS*YPLUS*(1-DEXP(-0.0154*UPLUS*YPLUS))
                AMU=AMUT
               ELSE
                USTAR = (SIGMA(2)/DEN)**0.5
         UEL = (U(1,NEL)+U(1,NEL+1))/2.
         UPLUS = UEL/USTAR
                AMU=DEN*CKARMANI/6.*(RM-RI)*USTAR*(1.-((RM-R)/(RM-RI))**2.)*
     .             (1.+2.*(RM-R)/(RM-RI))
               ENDIF
       ELSE
C              ETAO=(RM-R)/(RM-RO)
C              ETAPLUSO=1.5*YPLUS*(1.+ETAO)/(1.+2.*ETAO**2.)
C         IF (ETAPLUSO .LE. 42.)THEN
        IF (YPLUS .LE. 26.)THEN
                USTAR = (SIGMA(NELEM)/DEN)**0.5
         UEL = (U(1,NEL)+U(1,NEL+1))/2.
         UPLUS = UEL/USTAR
C               AMU=VIS*0.0154*UPLUS*YPLUS*(1-DEXP(-0.0154*UPLUS*YPLUS))
                AMU=AMUT
               ELSE
                USTAR = (SIGMA(NELEM)/DEN)**0.5
         UEL = (U(1,NEL)+U(1,NEL+1))/2.
         UPLUS = UEL/USTAR
                AMU=DEN*CKARMANO/6.*(RO-RM)*USTAR*(1.-((R-RM)/(RO-RM))**2.)*
     .             (1.+2.*(R-RM)/(RO-RM))
               ENDIF
       ENDIF
      ENDIF
C...
      ELSE
       AMU = 0.0
C... END OF IF(MAT.EQ.2)
      ENDIF
C...
C... DEISSER AND REICHARDT
C...
      IF (MIXMODEL .EQ. 'M3' .AND. ITER.GT.1 .AND.
     .     NGEOMTYPE .EQ. 21 .OR. NGEOMTYPE .EQ. 22) THEN
       USTAR = (SIGMA(2)/DEN)**0.5
       AMUMIN=DEN*CKARMANI/6.*(RM-RI)*USTAR
       IF (NEL .GE. IVIS1 .AND. NEL .LE. IRM)THEN
               AMU=DMAX1(AMU,AMUMIN)
       ELSEIF (NEL .GT. IRM .AND. NEL .LE. IVIS2)THEN
               AMU=DMAX1(AMU,AMUMIN)
       ENDIF
C      WRITE(2,*)'NEL,IVIS1,IVIS2,IRM,AMUMIN,AMU'
C      WRITE(2,121)NEL,IVIS1,IVIS2,IRM,AMUMIN,AMU
C121   FORMAT(1X,4I4,16(1X,1PE11.4))
```

```
      ENDIF
C
C... CALCULATE TURBULENT PRANDTL NUMBER
C
      IF(FPROP.EQ.'FIXED')THEN
       VISTEMP = VIS
       CPTEMP = CP
       AKTEMP = AK
       PR = VIS*CP/AK
      ELSEIF (FPROP.EQ.'FIXTB')THEN
       VISTEMP = VISF(TAVE)
       CPTEMP = CPF(TAVE)
       AKTEMP = AKF(TAVE)
       PR = VISTEMP*CPTEMP/AKTEMP
      ELSE
       VISTEMP = VISF(UELEM(2,NEL))
       CPTEMP = CPF(UELEM(2,NEL))
       AKTEMP = AKF(UELEM(2,NEL))
       PR = VISTEMP*CPTEMP/AKTEMP
      ENDIF
      IF(IPRT.EQ.0)THEN
        PRT(NEL)=PRT0
      ELSEIF(IPRT.EQ.1)THEN
        PET=AMU/VISTEMP*PR
        PRT(NEL)=2./PET+0.85
      ELSEIF(IPRT.EQ.2)THEN
        PRT(NEL)=1./(0.5882+0.228*(AMU/VISTEMP)-
     .        0.0441*(AMU/VISTEMP)**2.*
     .        (1-DEXP(-5.165/(AMU/VISTEMP))))
      ELSEIF(IPRT.EQ.3)THEN
       IF (NEL .GT. NFEL) THEN
        PRT(NEL)=1+0.855-DTANH(0.2*(YPLUS-7.5))
       ELSE
        IF (YPLUS.GT.5.0 .OR. SPLUS.GT.5.0)THEN
         PRTY=1+0.855-DTANH(0.2*(YPLUS-7.5))
         PRTS=1+0.855-DTANH(0.2*(SPLUS-7.5))
         PRT(NEL)=PRTY*PRTS/(PRTY+PRTS)
        ELSE
         PRT(NEL)=1.0
        ENDIF
       ENDIF
      ELSE
        WRITE(NLG,*)'SPECIFY IPRT (TURBULENT PRANDTL NUMBER OPTION)'
      ENDIF
C...
      ALPA(NEL)=ALP
      ALCA(NEL)=ALC
      YA(NEL)=Y
      SA(NEL)=S
      DFPA(NEL)=DFP
      DFCA(NEL)=DFC
      YPLUSA(NEL)=YPLUS
      SPLUSA(NEL)=SPLUS
      TWSA(NEL)=TWS
      TWYA(NEL)=TWY
      XG_EL(NEL)=XG
      YG_EL(NEL)=YG
C
```

```
      AMUST(NEL) = AMU
C
C... CALCULATE DISTANCE FROM THE WALL
      IF (NEL .GT. NFEL) THEN
       YY(NEL)=Y
      ELSE
       YY(NEL)=Y*S/(Y+S)
      ENDIF
C...
      RETURN
      END
```