GEORGIA INSTITUTE OF TECHNOLOGY | OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

[x] ORIGINAL
[ ] REVISION NO. ____

Project No. E-20-613 | DATE: 7/14/81

Project Director: Dr. Mustafa M. Aral | School/~~Lab~~ Civil Engineering

Sponsor: Beloit Corporation, Paper Machinery Division

Type Agreement: Research Agreement with: P. O. No. 15123

Award Period: From 6/30/81 To ~~6/30/83~~ 8/31/83 (Performance) 6/30/83 (Reports)

Sponsor Amount: $46,297* | Contracted through:

Cost Sharing: None | GTRI/~~GIT~~

Title: An Analysis of Rimming Condensate Flow

ADMINISTRATIVE DATA | OCA CONTACT Faith G. Costello

1) Sponsor Technical Contact: Gregory L. Wedel, Manager of Heat Transfer Research, Beloit Corporation, Paper Machinery Division, 1 St Lawrence Ave., Beloit, WI 53511 Ph: (608) 365-3311 ext 245

2) Sponsor Admin./Contractual Contact: --See Above--

Reports: See Deliverable Schedule | Security Classification: N/A

Defense Priority Rating: N/A

RESTRICTIONS

See Attached N/A Supplemental Information Sheet for Additional Requirements

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with Sponsor, however none proposed

COMMENTS: Currently funded at $10,000 through 9/30/81 on P. O. #15123

COPIES TO:

Administrative Coordinator
Research Property Management
Accounting Office
Procurement/EES Supply Services
Research Security Services
Reports Coordinator (OCA)
Legal Services (OCA)
Library, Technical Reports
EES Research Public Relations
Project File (OCA)
Other: ____

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION SHEET

SR384
2

Date 9/23/83

Project Title: An Analysis of Rimming Condensate Flow

Project No: E-20-613

Project Director: Dr. Mustafa M. Aral

Sponsor: Beloit Corporation, Paper Machinery Division

Effective Termination Date: 8/31/83

Clearance of Accounting Charges: 8/31/83

Grant/Contract Closeout Actions Remaining:

- [X] Final Invoice ~~and Closing Documents~~
- [ ] Final Fiscal Report
- [ ] Final Report of Inventions
- [ ] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other ____________

Assigned to: Civil Engineering (School/~~Laboratory~~)

COPIES TO:

R-A-N ~~Administrative Coordinator~~	Research Security Services	EES Public Relations (2)
Research Property Management	Reports Coordinator (OCA)	Computer Input
Accounting	Legal Services (OCA)	Project File
Procurement/EES Supply Services	Library	Other Aral

FORM OCA 10:781

FINAL REPORT

PROJECT NO.: E-20-613

AN ANALYSIS OF RIMMING CONDENSATE FLOW

BY

MUSTAFA M. ARAL

PREPARED FOR

BELOIT CORPORATION

BELOIT, WISCONSIN

AUGUST, 1983

GEORGIA INSTITUTE OF TECHNOLOGY

A UNIT OF THE UNIVERSITY SYSTEM OF GEORGIA

SCHOOL OF CIVIL ENGINEERING

ATLANTA, GEORGIA 30332

FINAL REPORT

PROJECT NO. E-20-613

AN ANALYSIS OF RIMMING CONDENSATE FLOW

by

Mustafa M. Aral

School of Civil Engineering

Georgia Institute of Technology

Atlanta, Georgia

Prepared for

BELOIT CORPORATION

Beloit, Wisconsin

August, 1983

ACKNOWLEDGEMENTS

This final report is the result of a two year contract between the Beloit Corporation and the Georgia Institute of Technology. The study was conducted in School of Civil Engineering, Georgia Institute of Technology, Atlanta, Georgia.

The principal investigator wishes to take this opportunity to thank Beloit Corporation for the financial support provided to perform this challenging research. Special thanks go to Mr. Gregory L. Wedel of Beloit Corporation who was in charge of administering the project and who patiently addressed several technical and non-technical aspects of the study via telephone dialogs and visits to Georgia Tech campus.

The assistance of Ms. L. LaSalata in typing of this report is also gratefully acknowledged.

Computational services were provided by the School of Civil Engineering at the Georgia Institute of Technology.

The purpose of this study was to develop mathematical/numerical models to predict the thickness of rimming condensate on the inner surface of paper drying machinery designed and manufactured by Beloit Corporation. The rotating drum is modeled as a flat plate and one and two-dimensional mathematical/numerical models are developed which predict the film thickness distribution on the plate surface for various syphon locations, sizes and strengths.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

Fluid flows have been studied mathematically for centuries with the result that many flow systems are well understood, but the supply of interesting, and unsolved, problems in fluid mechanics is by no means exhausted. One such research area comes under the heading of free surface flows, flows in which a liquid surface is free to adjust its location and shape according to the flow conditions beneath it. Analytical methods yield solutions only for very simple systems, thus approximate numerical methods must be used if solutions to complex systems are searched.

The range of problems to which such methods could profitably be applied is indeed large. There are many industrially important instances of viscous free surface flows, including fiber spinning, rotational molding, and all types of coating operations. The paper industry, photographic film industry and adhesives industry in particular employ viscous free surface flows in their processing operations. The extraordinary richness of applications explains the continuing scientific interest in these problems and testifies to the utility and value of techniques for solving such problems.

This study concerns a free surface flow problem which is encountered in paper industry. It is known that one of the most energy intensive areas of the paper manufacturing industry is the stage where the paper web is dried utilizing heated rotating drums, Brit (1970), Gavelin (1972) and Warren (1980). Currently, in a typical operation, steam is injected into rotating dryers which are used to evaporate the water out of the paper and yield a desired moisture content through a heat transfer process between the dryer shell and the paper web. The efficiency and evenness of this drying process is greatly affected by the overall thickness and the distribution of the condensate, rimming on the inner surface of the rotating drum. Since the film is an important resistance to heat transfer, the water that has condensed on the inner surface of the dryer must be removed through a syphon to provide for the condensation of the steam to continue so as to permit maximum heat transfer. The discussion of the mechanics and operation of such a dryer-syphon design is beyond the scope of this study; however, the motion of the condensate on the inner surface of a rotating drum is another example of free surface flows which will be addressed in this study.

It is possible to define several modes of operation for a dryer-syphon operation. At low rotation rates, the condensate lies at the bottom of the cylinder forming a puddle. Depending on the speed of the drum this puddle may slide up the inner wall, but not very far. This dryer operation mode is known as puddling. The next mode is called cascading. In this stage water begins to creep up the inside wall of the dryer until it starts to cascade down because the centrifugal forces, created by the increased speed of the dryer, have not yet exceeded the gravitational forces. At further higher speeds the whole condensate goes virtually into solid body rotation which is identified as the rimming mode. Rimming flow mode is a non-trivial example of a steady, three-dimensional, viscous flow with a free surface. The main aim of this study is the development of a numerical model which may be used in determining the thickness of such a condensate rimming on the inner surface of a rotating drum.

Two numerical models developed in this study can be identified as one-dimensional (1-D) and two-dimensional (2-D) models. However, as will be described in the following sections of this report, since the thickness of the condensate is measured normal to the plane of flow, the flow picture obtained as a result is two-dimensional in case of the (1-D) model and is three-dimensional in case of the (2-D) model. In developing the governing equations for the problem described above one basic assumption is valid for all cases studied, i.e., the condensate in the drum is assumed to be in the rimming mode. In this mode dryer speed is sufficient to make the gravitational forces negligible compared to the centrifugal forces. This assumption allows the dryer can's cylindrical surface to be cut and unrolled into a flat plate with syphon located along the center line, Figure (1). Details of this approach can be found in Beloit (1975). Other than this assumption, the condensate is represented as a Newtonian, viscous, incompressible fluid at constant temperature and steady state conditions arising from a condensate loading condition and syphon location is reached through the solution of non-linear, time dependent Navier Stokes equations.

In the following sections, first the governing equations used in one and two-dimensional models are described. In section three the numerical model, i.e., the finite element Galerkin approximation of the governing equations are given. In section four a detailed description of the computer code is given along with the details of a

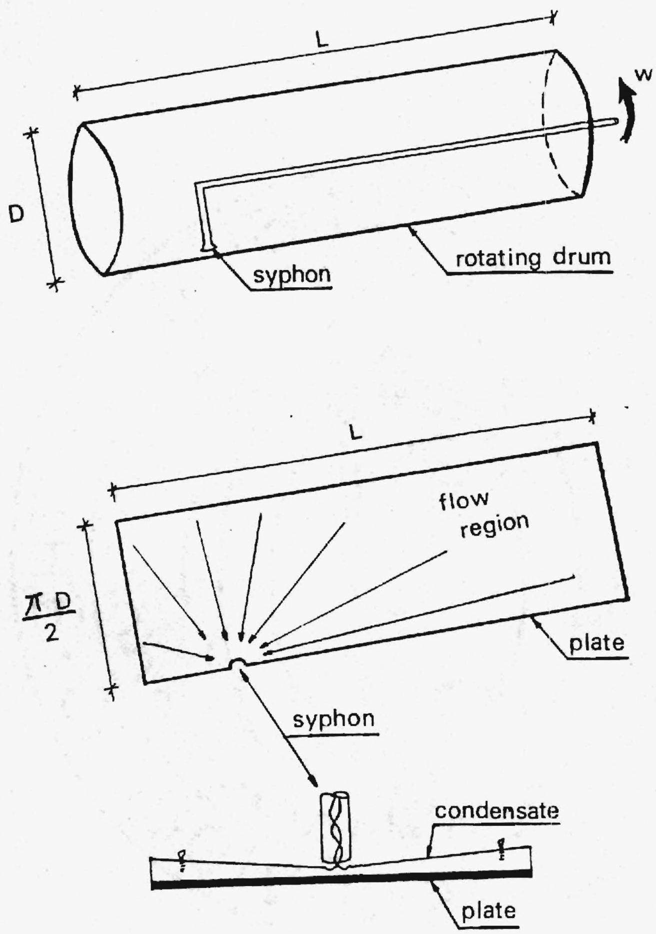

Figure 1. Rotating Drum and Flat Plate

users manual which describes the input-output procedures of the computer code generated. Finally, section six summarizes the conclusions and recommendations of the principal investigator.

SECTION II

MATHEMATICAL MODEL

The rimming condensate flow briefly described in section one is a three dimensional problem which may be described in terms of Navier Stokes equations. The present state of the art and the lack of suitable data in many instances does not justify the use of this complex (3-D) mathematical model for the solution of free surface flow problems as well as rimming condensate problems analyzed in this study. Thus, fully three dimensional solutions are not warranted at this stage as they would require a large amount of extra data and computer time. However, solution of vertically averaged Navier Stokes equations is feasible at this point. In what follows a summary derivation of the vertically averaged equations used in this study for (2-D) plane flow problems are given. The governing equations of a (1-D) problem can be extracted directly from these equations. Thus, details of derivation of (1-D) flow equations are not given here, instead a summary listing of the equations are included at the end of this section for reference.

II.a. Two-Dimensional Model

The form of equations used here has been developed fully elsewhere, Chen and Chow (1971). Therefore only a brief resume is included here. This is sufficient to illustrate the subsequent development of the finite element matrix equations necessary to provide a solution to the generalized governing equations.

The governing equations for the fluid, neglecting temperature effects may be given as,

$$\frac{\partial\ (\rho V_i)}{\partial\ x_i} + \frac{\partial \rho}{\partial t} = 0 \qquad i = 1,2,3 \qquad (1)$$

for the continuity equation and,

$$\rho \frac{D(V_m)}{Dt} = - \frac{\partial P}{\partial x_k} + \frac{\partial \sigma_{ik}}{\partial x_i} + \rho\ \bar{b}_k \qquad i,k = 1,2,3 \qquad (2)$$

for the momentum equation. Here $V[\bar{u}_k]$ are velocities in (x_1,x_2,x_3), $[k = 1,2,3]$ directions, Figure 2. ρ is the density of condensate, P is the pressure, g the gravitational acceleration, t the time, $\bar{b}_k$ external forces and σ_{ik} the kinematic Stokes tensor described as,

$$\sigma_{ik} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \qquad i,j = 1,2,3 \qquad (3)$$

Difficulties involved in the solution of these equations can be resolved by simplifying these equations through a classic approach which assumes that the pressures are hydrostatic in third direction normal to the flow plane and that only shear stresses from horizontal velocity components are important. In addition to these assumptions the main aim now is to integrate equations (1) and (2) in the third direction. For this, one needs to use the Leibnitz' rule for partial differentiation of an integral between variable limits, i.e.

$$\frac{\partial}{\partial x_i} \int_{b(x_1,x_2)}^{h(x_1,x_2)} f(x_1,x_2,x_3)\ dx_3 = \int_b^h \frac{\partial f}{\partial x_i}\ dx_3 + f \Big|_h \frac{\partial h}{\partial x_i} - f \Big|_b \frac{\partial b}{\partial x_i}$$

$$i = 1,2 \qquad (4)$$

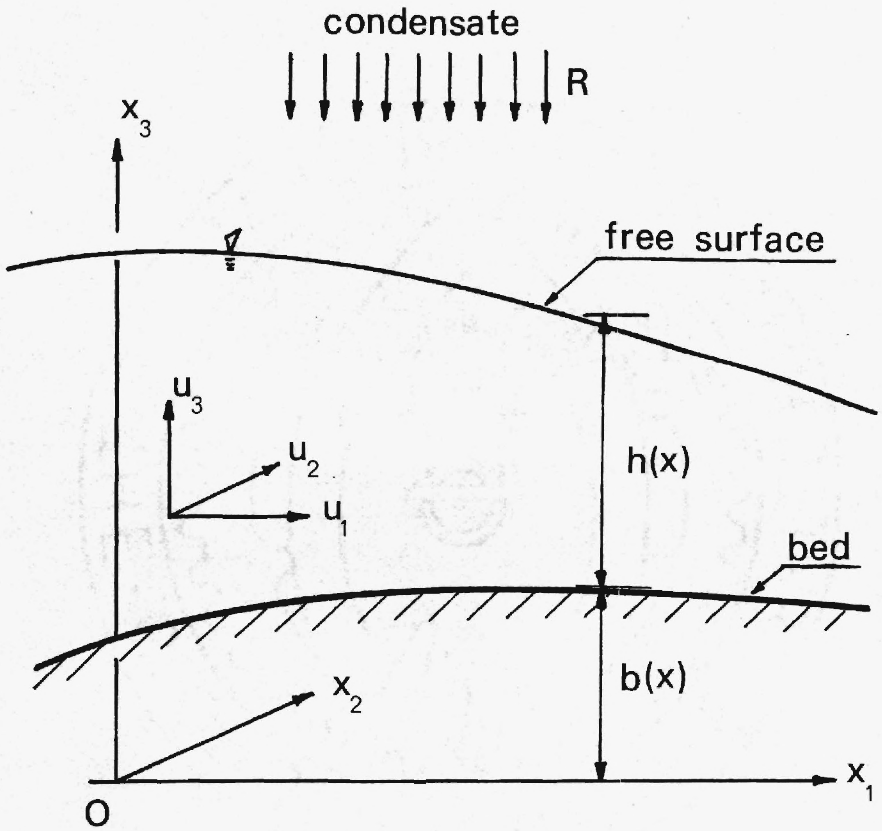

Figure 2. Flow Geometry and Coordinate System

where $h(x_1,x_2)$ is the depth of fluid above the plate bed elevation which is given as $b(x_1,x_2)$, Figure 2.

Upon integration over a section parallel to x_3-axis and using appropriate boundary conditions for an incompressible fluid equation (1) becomes;

$$\frac{\partial h}{\partial t} + h\frac{\partial u_i}{\partial x_i} + u_i\frac{\partial h}{\partial x_i} = R \qquad i = 1,2 \tag{5}$$

where u_i are spatially averaged velocities in the x_1- and x_2-directions respectively,

$$u_i\ (x_1,x_2,t) = \frac{1}{h}\int_b^h \bar{u}_i\ (x_1,x_2,x_3,t)\ dz \qquad i = 1,2 \tag{6}$$

and R is the rate of condensation over the plate.

Similarly, integration of the momentum equation in x_3 direction yields,

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} + g\frac{\partial h}{\partial x_i} - g\frac{\partial b}{\partial x_i} - \frac{1}{\rho}\frac{\partial \sigma_{ij}}{\partial x_j} + R\frac{u_i}{h} - \tau_i\Bigg|_b^h = 0 \qquad i,j = 1,2 \tag{7}$$

where $(\frac{\partial b}{\partial x_i})$ represents the slope of the plate in respective directions, $\tau_i(h)$ is the free surface resistance term which is assumed zero and $\tau_i(b)$ is the drag at the channel bed. As will be described later, drag at the channel bed can be represented via Darcy friction coefficient or by the use of Chezy equation. Although Darcy friction coefficient approach is Reynold's number dependent which may lead to complications depending on flow regime, both of these versions are included into the computer code developed to compute the channel drag terms.

Equations (5) and (7) now constitute the two-dimensional form of the equations governing the rimming condensate flow. Boundary conditions for such a flow can be described as follows; on segments of boundary where there is zero outward flux, components of the velocity vector normal to the boundary are zero. On other segments of the boundary the outflow flux might be specified which implies that the velocity field and depth of condensate at that segment is given. In addition to these two types of boundary conditions an initial distribution of the velocity field and condensate thickness is needed as an initial condition to start the solution.

II.b. One-Dimensional Model

Governing equations for (2-D) flow when reduced to (1-D) flow take the following form. The continuity equation;

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + h\frac{\partial u}{\partial x} = R \tag{8}$$

The momentum equation;

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} - g\frac{\partial b}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} - \tau\Big|_b^h + R\frac{u}{h} = 0 \tag{9}$$

in which the variables used are as described earlier but defined only in x-direction. The boundary and initial conditions of the problem are also as defined earlier.

II.c. Channel Bed Drag

It is possible to describe channel bed drag terms, $\tau_i(b)$, using two approaches; the first approach is the use of Darcy friction factor definition. Functionally, one can assume that

$$\tau_i(b) = F\ (\rho,\ V,\ \mu,\ d,\ \varepsilon) \tag{10}$$

where ε is the wall-roughness height, ρ the density, μ the viscosity, d a reference length parameter which may be chosen as condensate thickness and V is the velocity vector. Given equation (10), the dimensional analysis tells us that,

$$\frac{8\ \tau_w}{\rho V^2} = f = F\ (R_e,\ \frac{\varepsilon}{d}) \tag{11}$$

The dimensionless parameter f is called the Darcy friction factor. In laminar flow the Darcy friction coefficient is a function of the Reynolds number only and this relationship for a free surface flow can be given as,

$$f = \frac{24}{R_e} \tag{12}$$

The proof of this relationship was submitted to Beloit Corporation earlier thus it will not be repeated here. If the flow is turbulent however, the relationship between f, R_e and $\frac{\varepsilon}{d}$ is not well established for free surface flows especially in the transition zone. This poses problems for the two dimensional condensate flow problem studied here since it is expected and observed that the flow is either in transition or turbulent regime for a very small zone near the syphon. Nevertheless, equation (11) is implemented into the computer code generated in this study as an alternative that might be used by the user. However, the use of equation (11) to describe boundary drag terms should be avoided whenever Reynolds number reaches a (1500-2000) range at any point in the solution region. Here the Reynolds number is defined as

$$R_e = \frac{\rho V h}{\mu} \tag{13}$$

The second approach in defining the boundary drag terms is the use of the well known Chezy equation, Chow (1959). The representative forms for these stresses can be given as

$$\tau_i(b) = - \frac{g\, u_i}{C^2 h} (u_j\, u_j)^{1/2} \qquad i,j = 1,2 \tag{14}$$

where g is the gravitational acceleration, u_i is the velocity components in (x_1) and (x_2) directions and C is the Chezy coefficient which can be given as

$$C = \frac{1.49}{n} (h)^{1/6} \tag{15}$$

in British unit system. Here (n) is called Manning's roughness coefficient and is taken to be dimensionless. Table I below summarizes some values for this coefficient which are of interest in this study.

This definition of the boundary drag terms is also incorporated into the model developed in this study. It is suggested that the user should use this definition in analyzing problems with syphon effects.

TABLE I

Experimental Values of Manning's (n) Factor

Surface	n	Average Roughness Height (ft)
Glass	0.010 ± 0.002	0.0011
Brass	0.011 ± 0.002	0.0019
Steel, smooth	0.012 ± 0.002	0.0032
Steel, painted	0.014 ± 0.003	0.008
Steel, riveted	0.015 ± 0.002	0.012
Cast Iron	0.013 ± 0.003	0.0051
Cement, finished	0.012 ± 0.002	0.0032
Cement, unfinised	0.014 ± 0.002	0.008
Planed Wood	0.012 ± 0.002	0.0032
Corrugated Metal	0.022 ± 0.005	0.12

SECTION III

NUMERICAL MODEL

A finite element model is used to approximate the mathematical model developed in the previous sections. The first step in such a discretization process is the division of the solution region into a finite number of subregions which are called elements. This process is dictated by the need to find an alternative form of the equilibrium equations which will be easier to solve than the governing equations of the continuum. The modified conceptualization of the system results in a set of simultaneous algebraic equations rather than differential equations, thus simplifying the solution considerably. The size and distribution of the elements and the approximation used in each element are arbitrary. Given the one-dimensional and two-dimensional nature of the problem analyzed, two nodal one dimensional and three nodal two dimensional elements are used in the solution process for (1-D) and (2-D) numerical models respectively.

In developing the finite element matrix equations the Galerkin weighted residual process was adapted. According to this principle, the whole domain, denoted by A, is discretized into a number of elements, then the global assembly of all elemental contributions of weighted integral residuals is set to zero (Zienkiewicz, 1971; Gallagher, 1975).

$$\sum_{e=1}^{ne} \iint_{A^e} N_i \, [D \, (\phi_a)] \, d \, A^e = 0 \qquad i = 1,2,3,\ldots,n \quad (16)$$

Where N_i is the appropriate weighing function, D is the differential operator, and ϕ_a is an unknown function within the domain A. A^e is the area of element (e), (n) is the number of nodes in each element and (ne) is the total number of elements in A. Next step is the definition of the approximations used for the primary unknowns of the problem. For a Galerkin approach these take the form,

$$\bar{h}\ (x,y,t) = \sum_{i=1}^{n} N_i\ (x,y)\ h_i(t) \tag{17}$$

$$\bar{u}_j\ (x,y,t) = \sum_{i=1}^{n} N_i\ (x,y)\ u_{ji}(t) \qquad j = 1,2 \tag{18}$$

where N_i are the interpolating polynomials and (h_i) and (u_{ji}) are the nodal values of the unknown dependent functions of the problem. After substituting equations (17) and (18) into (16) and integrating each equation three matrix equations result.

$$[M]\ \{\dot{h}\} + \{F'\} = 0$$

$$[M]\ \{\dot{u}_1\} + \{F''\} = 0 \tag{19}$$

$$[M]\ \{\dot{u}_2\} + \{F'''\} = 0$$

where the dot notation represents time derivatives.

A final matrix assembly gives a coupled form

$$[\bar{M}]\ \{\dot{q}\} + \{\bar{F}\} = 0 \qquad (20)$$

where [M] is a bounded symmetric matrix having dimensions of (3n x 3n) where n is the total number of nodes in the domain. Equation (19) implies that there are three unknowns at each node

$$\dot{q}_i = \begin{Bmatrix} h_i \\ u_{1i} \\ u_{2i} \end{Bmatrix} \quad \text{and} \quad \bar{F}_i = \begin{Bmatrix} F_i' \\ F_i'' \\ F_i''' \end{Bmatrix} \qquad (21)$$

Finally, integrating equation (20) with respect to time it is possible to obtain numerical values for (h, u_1, u_2) starting from an initial distribution. Details of this process can be found in references given above. In what follows a more detailed account of these matrix equations will be developed for the two dimensional model. The matrix equations of the one dimensional model will not be given here since they are a subset of two dimensional forms.

V.a. Two-Dimensional Numerical Model

A finite element approximation to equations (5) and (7) can be obtained through a Galerkin approach. Over an element the residual, $\bar{R}$, for equations (5) and (7) can be given as,

$$\bar{R}_1\ (\bar{h}) = \frac{\partial \bar{h}}{\partial t} + \bar{h}\ \frac{\partial \bar{u}_i}{\partial x_i} + \bar{u}_i\ \frac{\partial \bar{h}}{\partial x_i} - R \qquad i = 1,2 \qquad (22)$$

and

$$\bar{R}_2(\bar{u}_i) = \frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + g \frac{\partial \bar{h}}{\partial x_i} - g \frac{\partial b}{\partial x_i} - \frac{1}{\rho} \frac{\partial \sigma_{ij}}{\partial x_j}$$

$$+ R \frac{\bar{u}_i}{\bar{h}} + \tau_i (b) \qquad i,j = 1,2 \qquad (23)$$

Substituting the approximate forms for ($\bar{h}$, $\bar{u}_1$, $\bar{u}_2$) into equation (22) and weighing the residual resulting from the continuity equation, with respect to a weighing function N_k yields

$$I^e = \iint_{A_e} N_k \left[\frac{\partial h_m}{\partial t} N_m + h_m N_m \frac{\partial N_j}{\partial x_i} u_{ij} + u_{im} N_m \frac{\partial N_j}{\partial x} h_j\right.$$

$$\left. - R\right] dA^e \quad (i = 1,2), (j,k,m = 1,2,3,\ldots,n) \qquad (24)$$

where repeated indices indicate summation, n is the number of nodes, N_m is the weighing function which is chosen as the finite element shape functions in a Galerkin formulation and (h_m, u_{im}) are the nodal values of condensate thickness and velocity components in an element. Equation (24) is written for a single element; however, it is understood that the same procedure is applied to the entire medium. For the details of the approximations used in three nodal elements, which in turn yields N_m, one should refer to any basic textbook on finite element method, Cook (1974).

Given approximation forms N_m, it is possible to integrate equation (24) to obtain the first set of matrix equations. The first term in equation (24) for example, will yield the mass matrix which can be given as

$$m_{km} = \iint_{A_e} N_k \; N_m \; dA^e \qquad k,m = 1,2,3,\ldots,n \quad (25)$$

For a three nodal element it can be shown that the matrix (m_{km}) yields the following coefficient matrix.

$$m_{km} = \frac{A^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \qquad (26)$$

Similarly other terms in equation (24) can be integrated which will yield the load vector defined as {F'}. Details of these derivations are omitted here but can be found in introductory finite element texts.

Simiarly, substituting the approximate forms for ($\bar{h}$, $\bar{u}_1$, $\bar{u}_2$) into equation (23) and weighing the residual resulting from the momentum equation, with respect to a weighing function N_k yields

$$I^e = \iint_{A^e} N_k \; [\frac{\partial u_{im}}{\partial t} \; N_m + u_{jm} \frac{\partial N_\ell}{\partial X_j} u_\ell + g \frac{\partial N_m}{\partial x_i} \; h_m - g \frac{\partial b}{\partial x_i}$$

$$- \frac{1}{\rho} \; \frac{\partial \sigma_{ij}}{\partial \, x_j} \; + \; R \frac{u_{im} N_m}{h_\ell N_\ell} \; + \tau_i \; (b)]$$

$$(i = 1,2), \; j,k,l,m = 1,2,\ldots,n \quad (27)$$

where repeated indices indicate summation. Proper integration of equation (27) term by term yields the remaining two ordinary differential equations shown in equation (19). At this point the non-linear partial differential equations governing the rimming condensate flow problem is reduced to non-linear ordinary differential equations. The next stage is the iterative solution of these equations which will yield the nodal values of the unknown dependent variables at each time step.

III.b. Time Integration and Solution of Non-Linear Equations

The equation system shown in equation (19) forms the basis for a recurrence scheme where the straightforward middifference trapezoidal time stepping technique is used to integrate the equations in time. This implies that both q and F are assumed to vary linearly within a typical small time interval, Δt, therefore at a time level $t_i = t + \Delta t$ the general implicit recurrence relationship is:

$$[\bar{M}] \{\dot{q}\}_{t+\frac{\Delta t}{2}} = \{\bar{F}\}_{t+\frac{\Delta t}{2}} \qquad (28)$$

or:

$$[\bar{M}] \{q\}_{t+\Delta t} = [\bar{M}] \{q\}_t + \frac{\Delta t}{2} (\{\bar{F}\}_{t+\Delta t} + \{\bar{F}\}_t) \qquad (29)$$

Obviously convergence of an iterative procedure for $\{F\}_{t+\Delta t}$ is necessary for solution. The convergence can be accelerated, however, by successive relaxation where an initial guess is corrected by a process of predicting a new solution q^{j+1} as a weighed function of the previous iterates q^j and q^{j-1} such that,

$$q^{j+1} = q^{j-1} + \omega \, (q^j - q^{j-1}) \qquad (30)$$

where the superscripts indicate the iteration level within a timestep. The relaxation factor (w) usually lies in the range $0.5 < \omega < 1.5$.

The complete sequence of steps therefore can be given as:

1) Predict $\{q\}_{t+\Delta t}$ from the explicit step given below

$$[\bar{M}] \; q^{j-1}_{t+\Delta t} = [\bar{M}] \; \{q\}_t + \Delta t \; \{\bar{F}\}_t \qquad (31)$$

2) Use this prediction to form the vector $\{F\}_{t+\Delta t}$ and solve equation (29) to obtain the first iterate $\{q^j\}_{t+\Delta t}$

3) Compare $|\{q^j\}_{t+\Delta t} - \{q^{j-1}\}_{t+\Delta t}|$ to be within some convergence tolerance ε. Here Eucledian norm is used to test this tolerance.

4) If the convergence criterion is satisfied proceed to the next time step, but if ε is exceeded obtain a further value $q^{j+1}_{t+\Delta t}$ from relaxation equation (30).

5) Repeat steps 1, 2, 3 and 4 until the convergence criteria is satisfied.

6) Repeat the above process for each time step.

The process described above yields linear rates of convergence. If a faster rate of convergence is desired a Newton iterative form should be utilized which yields a quadratic rate iterative algorithm.

should be utilized which yields a quadratic rate iterative algorithm. Such an alternative could be considered as a possible extension of the present study.

SECTION IV

THE COMPUTER CODE

In earlier sections of this report, an outline of the mathematical model and the finite element Galerkin formulation process used to approximate the governing partial differential equations are given. In this section, the main consideration will be the computer code generated and the description of input-output (I/O) statements necessary to implement the computer code. As mentioned earlier two separate computer codes are developed in this study, i.e. a (1-D) model and a (2-D) model. The (1-D) model computer code developed was sent to Beloit Corporation on June 1982. Since this code is a simpler version of the (2-D) model and since it is in operation at Beloit Corporation for more than a year it will not be referred to in this section for I/O considerations. This section is mainly a users manual for the (2-D) model computer code.

The "BEL2D" computer program presented in Appendix A is written in Fortran IV computer language. The program is divided into nine subprograms and a main program. To avoid making the present code too complicated, some features are built into it. These include the use of two dimensional three nodal linear elements, the linear variation of the time derivative between time steps and the one step iteration technique used to solve the nonlinear equations. The computer code also has several default data generation routines which may help the user in the data preparation phase of the study. At this stage of the study, no attempt is made to improve on these limitations which can be the basis of another research effort. As it stands, the "BEL2D" computer code is capable of analyzing time dependent, two dimensional, free surface rimming condensate problems with a variable syphon strength and location.

IV.a. Description of the Program

Various parts of the computer code and their specific functions are described below.

The "MAIN" Program: The Main program controls the flow of operations in the program and performs the time-space computations. Input-output subroutines, matrix generation subroutines, assembly subroutines and matrix solution subroutines are directly controlled from the main program.

Subroutine "ASSEM": Performs the assembly of the element matrices forming global stiffness and mass matrices and load vectors. With this information, control goes back to the "MAIN" program.

Subroutine "BOUND": This subroutine introduces the Dirichlet boundary conditions on velocities and on condensate thickness at the boundaries of the flow region. Typically condensate thickness and velocity components are specified around the syphon and on the rest of the boundary the velocity component normal to the boundary is assigned a zero value.

Subroutine "INPUT": All the input data for the problem to be analyzed is either generated or read in, in this subroutine. More specifically, nodal pattern, element pattern, element constants, time constants, initial condition, boundary conditions are either read in or generated and printed out in this subroutine. Details of the generation routines will be given later on in this section.

Subroutine "OUTPUT": Printout of the results obtained for the problem analyzed is organized in this subroutine.

Subroutine "REDUCE": This subroutine performs the first step reduction in a Gausian elimination solution process on a non-symmetric banded matrix, stored as a rectangular array. Coefficients of the variables (h, u_1, u_2) are stored in a single matrix assembled by the subroutine "ASSEM". The control is then directed to the subroutine "SOLVE" by the "MAIN" program for the backsubstitution process.

Subroutine "SOLVE": This subroutine completes the backsubstitution process on the reduced matrices obtained from subroutine "REDUCE". Results are stored as a vector and control goes back to the "MAIN" program.

Subroutine "MLTPLY": Performs the multiplication of a non-symmetric banded matrix, stored as a rectangular array, with a vector. The resultant vector is stored in a separate location, and the control goes back to the "MAIN" program.

Subroutine "ELEM": This subroutine forms the local element stiffness and mass matrices and load vectors for each element. These matrices and vectors are then assembled by the subroutine "ASSEM" to form the global rectangular matrices.

Subroutine "CONVRG": This subroutine computes the convergence parameters using an Euclidean norm form within each iteration cycle. These parameters are later on used in the main program to establish the convergence characteristics of each iteration cycle within each time step.

IV.b. Control Cards and Input Data

The first step in the analysis is to select a finite element representation for the region of interest. Elements and nodal points are then numbered in two numerical sequences, each starting with one. A typical idealization is shown in Figure 3. Nodal numbering sequence on this figure is shown for most of the reference nodes, nodes which are shown with dots. Due to space limitations nodes around the syphon are not numbered although they are also reference nodes. Numbers printed to the left or right of a node indicate the nodal numbering sequence chosen. Remaining nodal numbers are generated using these reference nodes by the computer code. Numbers enclosed in parentheses on the same figure indicate the element numbering sequence. This numbering sequence is also generated by the computer code given base elements such as (1), (15), etc. which are shown as shaded elements. Data set associated with this idealization along with the results was sent to Beloit Corporation under a separate cover in July 1983. Following this initial step, the following group of data cards are necessary to operate the computer code.

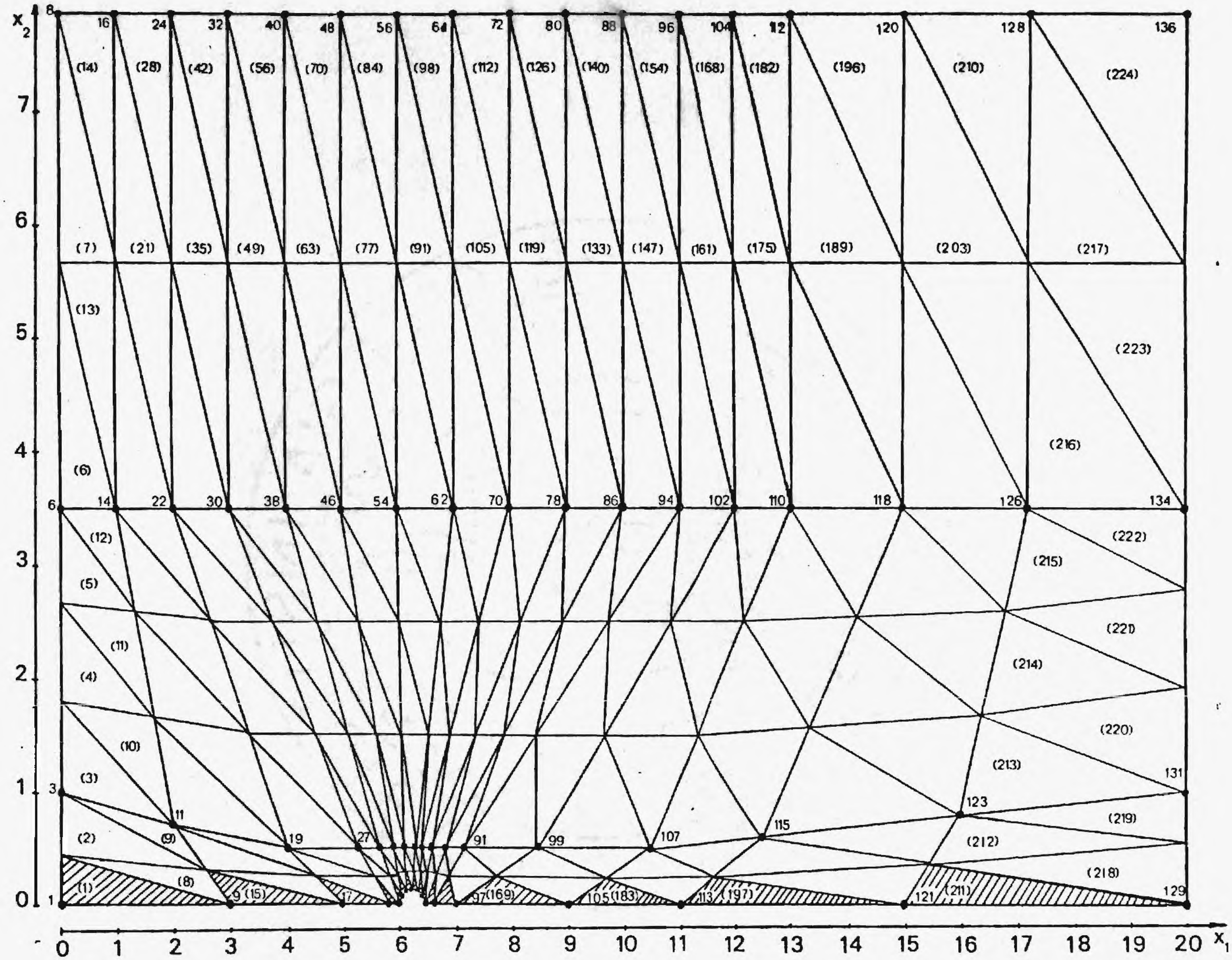

Figure 3. Finite Element Idealization of a Two-Dimensional Region

IDENTIFICATION CARD: (20A4)

Columns 1 to 80 of this card contain information to be printed as the title.

CONTROL CARD I: (6I5, 4F10.0)

Column (5)	(0)	Indicates the use of Darcy fraction coefficient to define boundary drag.
	(1)	Indicates that the choice is left to the code
	(2)	Indicates the use of Chezy equation to define boundary drag.
Columns (6-10)		Number of reference nodal points for which data will be read in. From this set the remainder of the nodal pattern data will be generated assuming equal spacing between nodes.
Columns (11-15)		Number of reference elements for which data will be read in. From this set the remainder of the element pattern data will be generated.
Columns (16-20)		Number of Dirichlet boundary nodes with condensate thickness specified.
Columns (21-25)		Number of Dirichlet boundary nodes with x-component velocity specified.
Columns (26-30)		Number of Dirichlet boundary nodes with y-component velocity specified.
Columns (31-40)		If a non-zero real number is specified here then a constant initial condensate thickness distribution equal to the assigned value is generated for all

nodes. If a zero is specified here than followup data in Control Card II is needed to specify an initial condensate thickness distribution.

Columns (41-50) If a non-zero real number is specified here then a constant initial x-component velocity distribution equal to the assigned value is generated for all nodes. If a zero is specified here then follow up data in Control Card II is needed to specify an initial x-component velocity distribution.

Columns (51-60) If a non-zero real number is specified here than a constant initial y-component velocity distribution equal to the assigned value is generated for all nodes. If a zero is specified here then follow up data in Control Card II is needed to specify an initial y-component velocity distribution.

Columns (61-70) This data should be read in as (1.00) always. This parameter is a relaxation parameter which accelerates the iterations performed at each time step. At this stage variations on this parameter are not fully implemented into the computer code.

CONTROL CARD SET II: (I10, 2F10.4, I10, 3F10.0)

In this data set, the number of data cards should be equal to the number of reference nodal points specified in Control Card I, columns (6-10).

Columns	(1-10)	Node number
	(11-20)	x-coordinate
	(21-30)	y-coordinate
	(31-40)	(0) Indicates node generation is not requested after this node
		(1) Indicates node generation is requested between this node and the next one
	(41-50)	(h), condensate thickness at node
	(51-60)	(u_1),x-component velocity at node
	(61-70)	(u_2),y-component velocity at node

If a zero is specified in columns (31-60) on Control Card I then a non-zero data should be given here for (h, u_1, u_2). Values of (h, u_1, u_2) for remaining nodes will be generated through linear interpolation between consecutive nodes. If a non-zero value is specified for either variable in Control Card I then corresponding data for (h, u_1, u_2) on this card set can be omitted.

CONTROL CARD SET III. (3I5, I10)

In this data set, the number of data cards should be equal to the number of reference elements specified in Control Card I, columns (11-15).

Columns (1-5)	First node number
Columns (6-10)	Second node number in counterclockwise direction in reference to first node above

Columns (11-15)	Third node number in counterclockwise direction in reference to second node above
Columns (16-25)	Number of elements on the same column after the reference element for which data generation is requested

CONTROL CARD IV. (5F8.0, 2E12.5, 2F6.0)

Columns (1-8)	Initial friction coefficient
Columns (9-16)	Gravitational acceleration
Columns (17-24)	Slope
Columns (25-32)	Condensation rate
Columns (33-40)	Error limit, usually set to 0.001
Columns (41-52)	Density
Columns (53-64)	Viscosity
Columns (65-70)	Should be set to (0.5). (Allows for the use of other time integration schemes, but not fully implemented at this stage).
Columns (71-76)	Mannings, n.

CONTROL CARD V. (4F10.0, I10)

Columns (1-10)	Initial time
(11-20)	Final time
(21-30)	Time step
(31-40)	Duration of condensation
(41-50)	Printout interval

CONTROL CARD VI. 8(I3, F7.0)

If Dirichlet boundary conditions on condensate thickness do not exist, then this set of cards should be omitted. Otherwise a data set equal to the number specified on Control Card I, columns (16-20) should follow.

8 (Columns(1-3), node number, columns (4-10), boundary condition)

CONTROL CARD VII. 8(I3, F7.0)

If Dirichlet boundary conditions on x-component velocity do not exist, then this set of cards should be omitted. Otherwise a data set equal to the number specified on Control Card I, columns (21-25) should follow.

8 (Columns(1-3), node number, columns (4-10), boundary condition)

CONTROL CARD VIII. 8(I3, F7.0)

If Dirichlet boundary conditions on y-component velocity do not exist, then this set of cards should be omitted. Otherwise a data set equal to the number specified on Control Card I, columns (26-30) should follow.

8 (Columns(1-3), node number, columns (4-10) boundary condition)

SECTION V

NUMERICAL RESULTS

Considerable effort and computer time was spent in solving several problems of interest to Beloit Corporation using the one dimensional model. Results of these computer runs are presented below. Completion of the two dimensional model on the other hand is more recent, thus only a few test runs were made using this code which will also be summarized at the end of this section. It is expected that further detailed use of the two dimensional code generated will be actualized at Beloit Corporation Laboratories.

V.a. One-Dimensional Model Test Runs

The data for the problem chosen to test the one dimensional computer code was obtained from Beloit Corporation. This data set corresponds to a drum with the following characteristic dimensions.

Internal Diameter, D_i = 5.0 ft

Length, L = 20.0 ft

Density of Condensate, ρ = 57.7 lb/ft^3

Viscosity of Condensate, μ = 1.29×10^{-4} lb/ft sec

Loading conditions ($\dot{m}$) and rotational speeds V_i considered in the thirty five computer runs made are summarized in Table II below. Results obtained were satisfactory and compared favourably with the earlier results obtained at Beloit Corporation.

Figure 4 given below shows a typical condensate thickness profile obtained in one of these computer runs. The legend summarizes the values of the specific constants used in this computer run. Figures 5

TABLE II

One Dimensional Model Data

$L = 20$ ft $\rho = 57.7$ lb/ft^2

$D_i = 5$ ft $\mu = 1.29 \times 10^{-4}$ lb/ft sec

Data No.	V_i (ft/min)	g(ft/sec^2)	$\dot{m}$(lb/hr)	R(ft/sec)	h_{min}(ft)	V_{SYPHON}(ft/sec)
1	1000	111.11	2000.0	3.07×10^{-5}	0.0022	0.279
2	1500	250.00	2000.0	3.07×10^{-5}	0.0016	0.383
3	2000	444.44	2000.0	3.07×10^{-5}	0.00135	0.454
4	2500	694.44	2000.0	3.07×10^{-5}	0.0012	0.511
5	3000	1000.00	2000.0	3.07×10^{-5}	0.0010	0.612
6	3500	1361.11	2000.0	3.07×10^{-5}	0.00093	0.659
7	4000	1777.77	2000.0	3.07×10^{-5}	0.00085	0.721
8	1000	111.11	2500.0	3.83×10^{-5}	0.0025	0.306
9	1500	250.00	2500.0	3.83×10^{-5}	0.0019	0.403
10	2000	444.44	2500.0	3.83×10^{-5}	0.0016	0.479
11	2500	694.44	2500.0	3.83×10^{-5}	0.00135	0.568
12	3000	1000.00	2500.0	3.83×10^{-5}	0.0012	0.639
13	3500	1361.11	2500.0	3.83×10^{-5}	0.0011	0.697
14	4000	1777.77	2500.0	3.83×10^{-5}	0.00099	0.774
15	1000	111.11	3000.0	4.6×10^{-5}	0.0028	0.328
16	1500	250.00	3000.0	4.6×10^{-5}	0.0022	0.418
17	2000	444.44	3000.0	4.6×10^{-5}	0.0018	0.511
18	2500	694.44	3000.0	4.6×10^{-5}	0.0015	0.613
19	3000	1000.00	3000.0	4.6×10^{-5}	0.00135	0.681
20	3500	1361.11	3000.0	4.6×10^{-5}	0.00122	0.754
21	4000	1777.77	3000.0	4.6×10^{-5}	0.00111	0.828
22	1000	111.11	3500.0	5.36×10^{-5}	0.0031	0.346
23	1500	250.00	3500.0	5.36×10^{-5}	0.0024	0.447
24	2000	444.44	3500.0	5.36×10^{-5}	0.00196	0.547
25	2500	694.44	3500.0	5.36×10^{-5}	0.00169	0.635
26	3000	1000.00	3500.0	5.36×10^{-5}	0.00150	0.715
27	3500	1361.11	3500.0	5.36×10^{-5}	0.00135	0.795
28	4000	1777.77	3500.0	5.36×10^{-5}	0.00124	0.865
29	1000	111.11	4000.0	6.13×10^{-5}	0.0034	0.361
30	1500	250.00	4000.0	6.13×10^{-5}	0.0026	0.472
31	2000	444.44	4000.0	6.13×10^{-5}	0.00215	0.570
32	2500	694.44	4000.0	6.13×10^{-5}	0.00185	0.663
33	3000	1000.00	4000.0	6.13×10^{-5}	0.00164	0.748
34	3500	1361.11	4000.0	6.13×10^{-5}	0.00147	0.834
35	4000	1777.77	4000.0	6.13×10^{-5}	0.00135	0.908

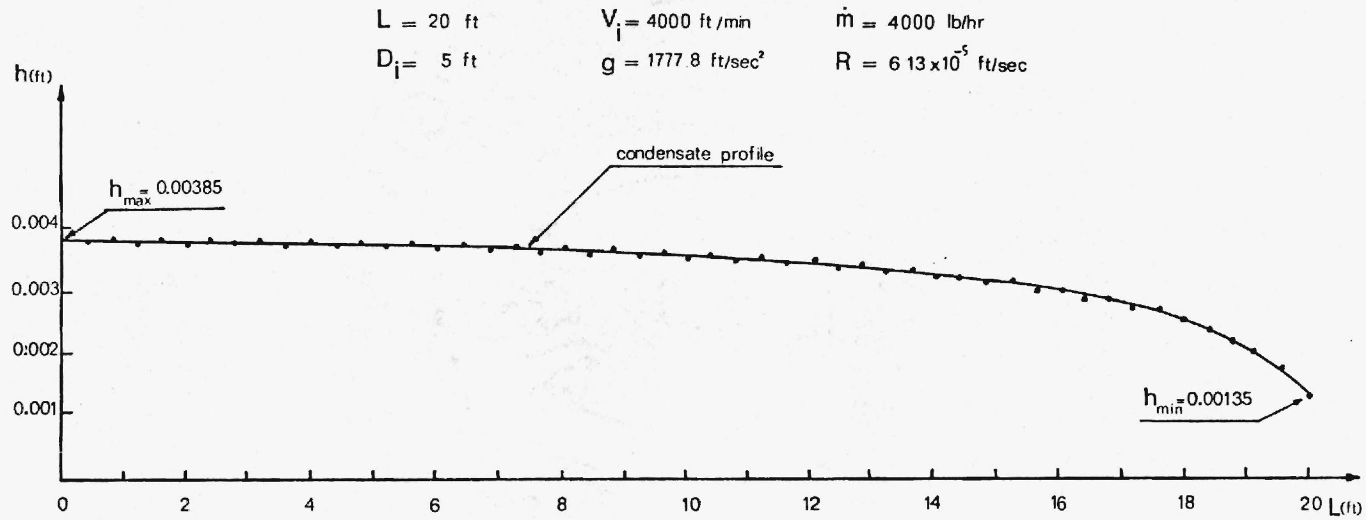

Figure 4. Condensate Profile for Data No.: 35

and 6 summarize the results obtained in terms of average depth h_{ave} and V_i and $\dot{m}$. These figures show expected trends between these variables and results obtained are in line with the earlier work done. Figures 7 and 8 contain plots of the same computer runs in terms of some nondimensional grouping of the parameters involved. In Figure 7 R_f is defined as the flow Reynolds number which is defined in terms of average flow velocity and average depth. On the vertical axis another dimensionless grouping is used which is again in terms of average flow depth.

$$R_f = \frac{\rho\, v_f\, h_{ave}}{\mu}, \quad N = \frac{\rho\, V_i\, h_{ave}^2}{\dot{m}}, \quad R_c = \frac{\rho\, V_i\, h_d}{M}$$

When results of the thirty five computer runs are plotted they form a very narrow band bounded by curves characterized by high and low rotational speeds. Given the organization of dimensionless parameters chosen in this plot it is possible to predict a functional relationship between average condensate thickness and average condensate velocity for various loading conditions and rotational speeds. However, this representation is not that useful since both axes contain parameters which are basically unknowns (u_j, h_{ave}). But if one defines a Reynold's number for the cylinder in terms of V_i and h_d where V_i is the rotational velocity of the drum and h_d is the downstream condensate thickness boundary condition which depends on syphon characteristics, then one can obtain h_{ave} from Figure 8 given $\dot{m}, \rho$ and V_i. Of course the results given in these figures are for a

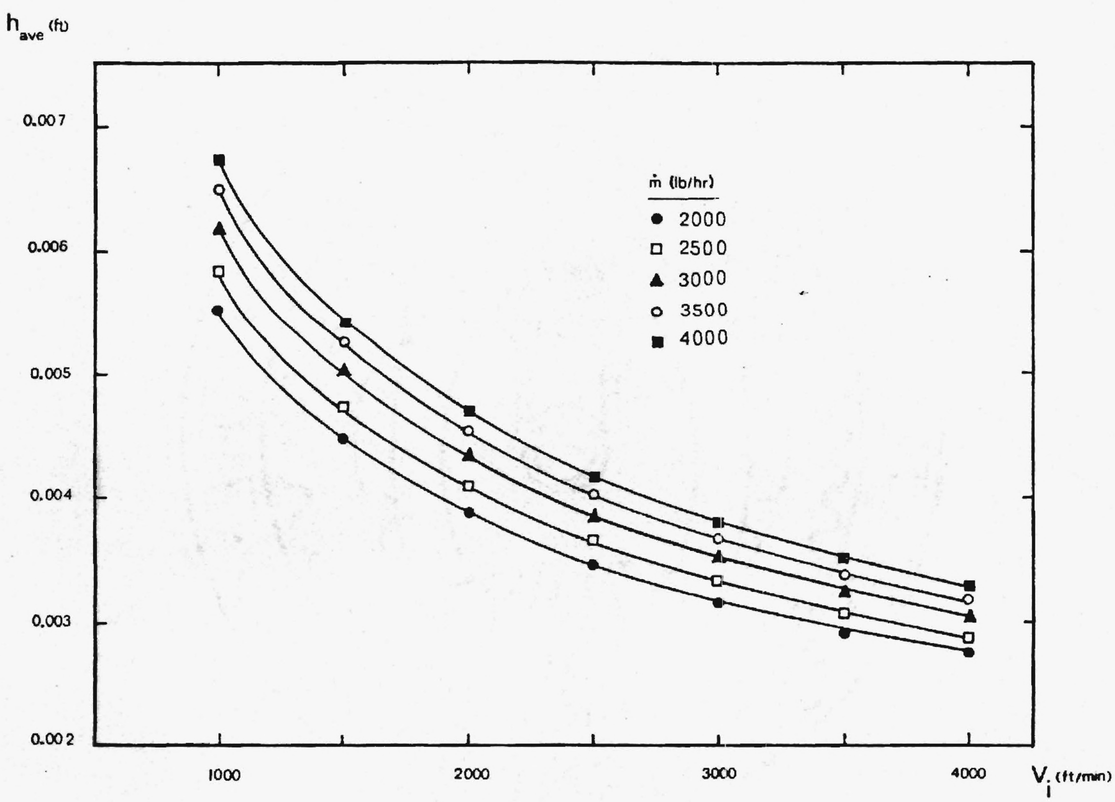

Figure 5. Average Depth vs. Rotation Velocity of the Drum

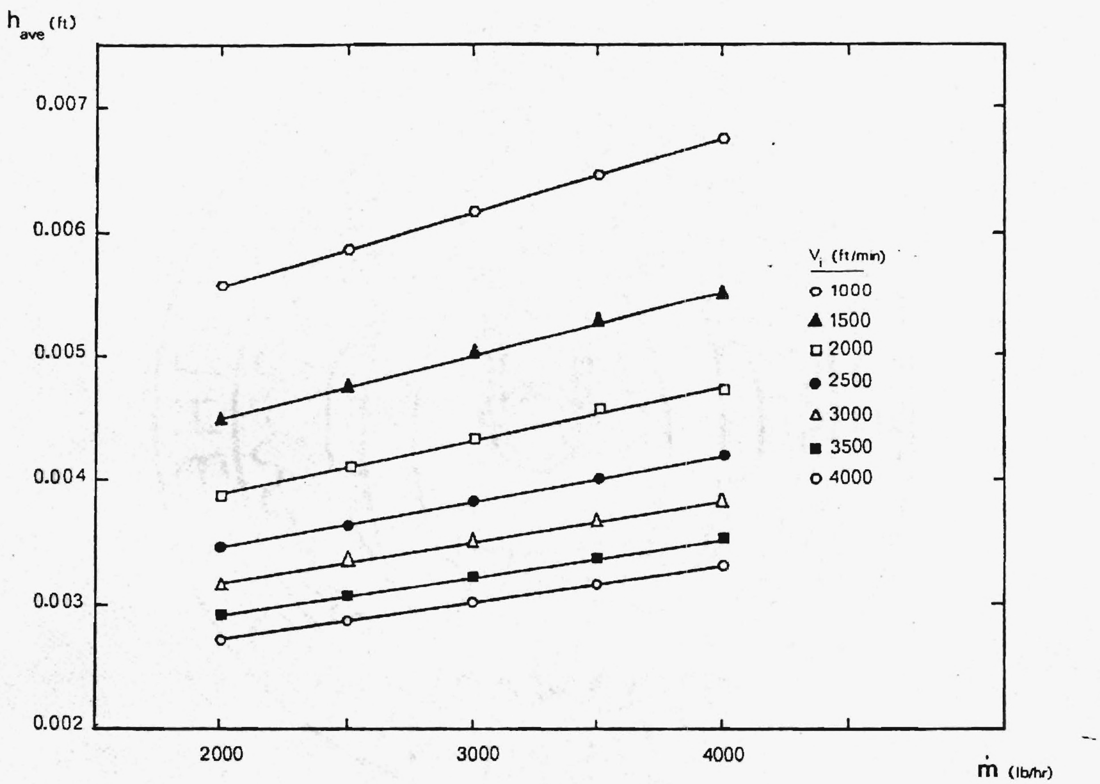

Figure 6. Average Depth vs. Condensate Loading

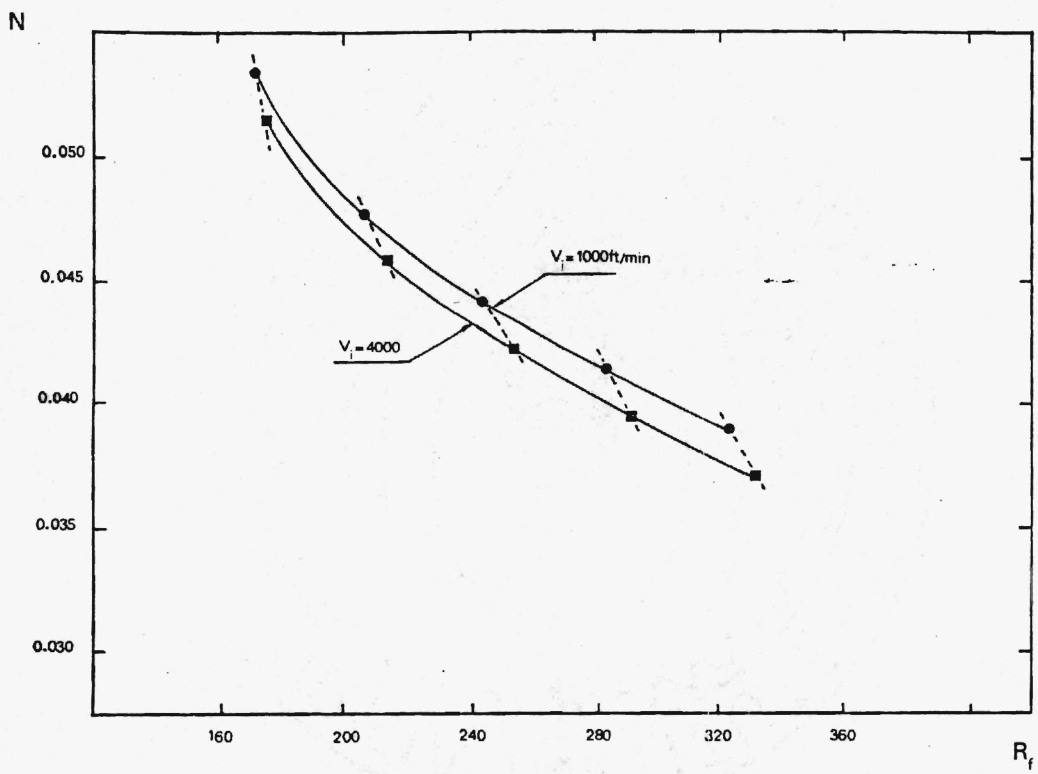

Figure 7. N vs. R_f

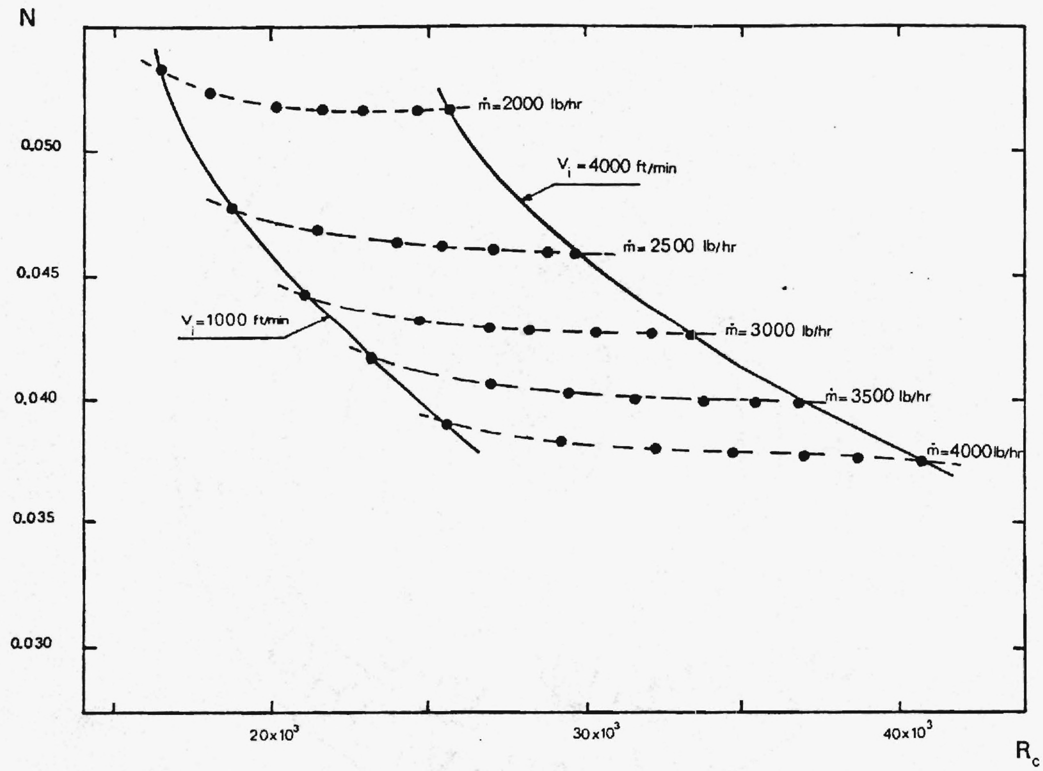

Figure 8. N vs. R_c

drum with dimensions specified in Table II. Other type curves can also be obtained from the finite element program developed. This line of activity is not pursued at this point which is in line with the proposed program of study.

V.b. Two-Dimensional Model Test Runs

Several test runs were also made with the two-dimensional computer code developed. Results obtained were also satisfactory and some of the printouts of these runs were sent to Beloit corporation on a separate cover earlier. The computer code was first tested to duplicate the results of the one dimensional computer runs. Data prepared to represent a flat plate with a line syphon at one end of the syphon yielded results similar to the ones presented earlier. Computer runs made with the two dimensional model with a non-symmetrically placed circular syphon (six feet away from the left boundary of the plate) showed clearly that the flow region at the immediate vicinity of the syphon is in either transition or turbulent flow regime. This fact necessitated the use of the Chezy coefficient description of boundary drag terms as opposed to Darcy friction definition which was satisfactorily used with the one dimensional code. Computer results obtained with this approach were also satisfactory and it is recommended that this version should always be used in analyzing problems with a circular syphon. The computer code also gives a mapping of elements indicating the specific flow regimes for each element in each time interval as a supplement. To be able to separate these flow regime regions properly it is suggested that smaller elements be used around the syphon compared to the element sizes chosen elsewhere on the plate.

SECTION VI

CONCLUSIONS AND RECOMMENDATIONS

This study was planned and carried out as an initial step in modeling rimming condensate flow problem in rotating drums of paper drying machinery. Throughout, the main aim was to develop a simple user oriented one-dimensional and two-dimensional numerical model which can be used in the analysis of such rimming condensate flow problems. The problem, as described in Section II of this report, is by no means a trivial problem and required considerable analytical and numerical expertise. The computer code generated is documented and steps involved in data preparation are summarized in detail in this report. To simplify the data preparation phase of the code several data generation routines are built in to the code. These are documented in Section IV of this report.

Several numerical experiments performed by the codes generated are also summarized in Section V of this report. When compared with earlier analytical and experimental results the model performs rather satisfactorily with a potential of analyzing problem types which were not possible to model utilizing the computer models available to Beloit Corporation at that time. As it stands now, the two-dimensional computer code generated is capable of analyzing rimming condensate flow problems for various loadings and syphon arrangements. With the present code it is also possible to treat problems with more than one syphon arbitrarily placed on the centerline of the plate. The efficiency of such a configuration might be of interest to Beloit Corporation.

Finally, an upgrading of the present two-dimensional model is possible and should be considered depending on the research needs of the Beloit Corporation. In terms of the accuracy and efficiency of the model the following can be incorporated into the model as a part of future work:

a) Inclusion of higher order approximations (higher order elements) into the finite element procedures.

b) Inclusion of a quadratic iteration process as opposed to the linear iteration process used in the model.

c) Improving the time integration processes used to higher order schemes.

All of these are related to the numerical aspects of the study. In terms of expanding the capabilities of the code generated, it is proposed that the code should be coupled with a finite element heat transfer analysis computer code so that after determining the condensate thickness distribution, the code should automatically generate the heat transfer characteristics of the plate on the same finite element mesh. Such a coupled algorithm should be accompanied with graphics capabilities allowing the designer to make changes on the design and see the effects of such a change comparatively and immediately.

REFERENCES

[1] Brit, K.W., Handbook of Pulp and Paper Technology, Van Nostrand, New York, (1970).

[2] Chow, V.T., Open Channel Hydraulics, McGraw-Hill, New York (1959).

[3] Chen, C.L., and Chow, V.T., "Formulation of Mathematical Watershed Flow Model", J. Eng. Mech. Div., ASCE, Vol. 97, (EM3), pp. 809-828, (1971).

[4] Cook, R.D., Concepts and Applications of Finite Element Analysis, John Wiley, New York (1974).

[5] Galagher, R.H., Finite Element Analysis Fundamentals, Prentice-Hall, New York, (1975).

[6] Gavelin, G., Drying of Paper and Board, Lockwood, New York, (1972).

[7] Warren, P.J., "An Analytical Study of Rimming Condensate Flow", M.S. Thesis presented to the School of Chemical Engineering, Georgia Institute of Technology (1980).

[8] Wedel, G.L., "Condensate Distribution and Its Effects on Heat Transfer in Steam-Heated Dryers", Beloit Corporation Report No. R79-026, Beloit Wisconsin (1979).

[9] __________, Beloit Corporation Research Report No. 166, Beloit, Wisconsin, (1975).

[10] Zienkiewicz, O.C., The Finite Element Method in Engineering Science, McGraw-Hill (1971).

APPENDIX A

The major arrays and symbols used in the BEL2D computer program are defined below. Some temporary storage variables are not defined here, but their definitions are evident from the context.

AREA = Area of triangular elements

BCDH = Dirichlet boundary condition on depth of condensate

BCDU = Dirichlet boundary condition on velocity, u_1

BCDV = Dirichlet boundary condition on velocity, u_2

CMN = Manning's (n)

ERR = Convergence error limit

F = Darcy friction coefficient

GR = Gravitational acceleration

H = Vector, nodal condensate thickness

HAVE = Average condensate thickness

HCONS = Constant condensate thickness, initial data

HIN = Vector, nodal condensate thickness, initial data

ICON = Element convectivity matrix

IUBW = Maximum upper band width

IFC = Flow control constant

IP = Printout interval

NDH = Number of Dirichlet boundary conditions on depth

NDU = Numer of Dirichlet boundary conditions on velocity, u_1

NDV = Number of Dirichlet boundary conditions on velocity, u_2

NNODE = Number of nodes

NELEM = Number of elements

NNPC = Number of reference nodes

NELEMC = Number of reference elements

NDBCNH = Node numbers with Dirichlet boundary condition on depth

NDBCNU = Node numbers with Dirichlet boundary condition on velocity, u_1

NDBCNV = Node numbers with Dirichlet boundary condition on velocity, u_2

NPMIS = Number of elements above reference element

NEWN = Local element connectivity matrix

P = Mass matrix

PHI = Vector, unknown variables

PHIN = Vector, initial values of unknown variables

R = Load vector

RHO = Density

RO = Condensation rate

SO = Slope

TCONT = Counter for time

TDUR = Condensate duration

TST = Time step

TI = Initial time

TF = Final time

U = Vector, nodal velocities (u_1)

UAVE = Average velocity

UIN = Vector, initial nodal velocities (u_1)

VIN = Vector, initial nodal velocities (u_2)

VISC = Viscosity

V = Vector, nodal velocities (u_2)

X = (x_1,x_2) global coordinates

XE = x_1, local coordinates

YE = x_2, local coordinates

```
      PROGRAM BEL2D(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

CC**********************************************************************
CC**  THIS IS A TWO DIMENSIONAL VERSION OF THE ORIGINAL
CC**  PROGRAM WRITTEN FOR BELOIT CORPORATION. FRICTION FACTOR IS
CC**  COMPUTED FOR EACH ELEMENT IN EACH ITERATION.
CC**  THE FRICTION TERMS ARE COMPUTED USING TWO APPROACHES :
CC**  IFC = 0 USES MOODY CHART, IFC = 2 USES CHEZY FORMULA WITH
CC**  MANNINGS COEFFICIENT.
CC**  ALL RUNS WITH THE SYPHON SHOULD USE IFC = 2 OR 1,
CC**  A SENSITIVITY ANALYSIS SHOULD BE MADE AT THIS STAGE.
CC**  IN ANY RUN IF THE ELEMENT REYNOLDS NUMBER GETS TO BE
CC**  GREATER THAN (1500) AGAIN IFC = 2 OR 1 SHOULD BE USED.
CC**  SENSITIVITY OF THE RESULTS SHOULD BE CHECKED FOR VARIATIONS
CC**  IN MANNINGS COEFFICIENT.
CC**  MAX NUMBER OF NODES FOR THIS CODE IS (150), MAX NUMBER OF ELEMENTS
CC**  FOR THIS CODE IS (300). THESE LIMITS CAN BE INCREASED OR DECREASED
CC**  BY ADJUSTING THE DIMENSION STATEMENTS.
CC**********************************************************************
      DIMENSION TITLE(20),P(450,20),R(450),X(2,150),
     2 PE1(3,3),RE1(3),U(150),V(150),HIN(150),UIN(150),
     3 VIN(150),PHI(450),H(150),PE2(3,3),RE2(3),PHIIN(450),
     4 RBN(450),NDBCNH(50),BCDH(50),NDBCNU(50),BCDU(50),
     5 NDBCNV(50),BCDV(50),RT(450),ICON(3,300),NEWN(3),
     6 U1(3),V1(3),H1(3),XE(3),YE(3),RE3(3),PE3(3,3),IEL(300)

      COMMON /MVEL/RE1,RE3,PE1,AREA,II,RE2,PE2,PE3,U1,V1,H1,UAVE,HAVE
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      COMMON /GMV/R,P,HIN,UIN,VIN,PHI,H,U,V
      COMMON /ELM/F,RO,GR,SO,VISC,RHO,CMN
      COMMON /TIME/TDUR,TITLE,TI,TF,TST,TCONT,IP
      COMMON /BC/NDBCNU,BCDU,NDBCNV,BCDV,NDBCNH,BCDH
      COMMON /LCL/NEWN,XE,YE,X,ICON
      COMMON/TURB/IEL

     CALL INPUT
     FP=F
     FIN = F
     NNODE3 = 3*NNODE
     ICONT = 0
130  TCONT = TI+TST
     ITIM = 0
     ITER = 0
128  CONTINUE
     DO 10 J=1,20
     DO 10 I=1,450
     P(I,J) = 0.
     RBN(I) = 0.
     R(I) = 0.
10   CONTINUE
     DO 15 I=1,NNODE
     PHIIN(I) = HIN(I)
     PHIIN(NNODE+I) = UIN(I)
     PHIIN((2*NNODE)+I) = VIN(I)
15   CONTINUE
     DO 20 II=1,NELEM
     DO 201 I=1,3
201  NEWN(I) = ICON(I,II)
     CALL SET

     DO 202 I=1,3
     U1(I) = UIN(NEWN(I))
     V1(I) = VIN(NEWN(I))
```

```
      H1(I) = HIN(NEWN(I))
      IF(ICONT.EQ.0)GO TO 21
      U1(I) = U(NEWN(I))
      V1(I) = V(NEWN(I))
      H1(I) = H(NEWN(I))
 21   CONTINUE
 202  CONTINUE
      SUMU = 0.0
      SUMH = 0.0
      DO 203 I=1,3
      UU = SQRT(U1(I)*U1(I)+V1(I)*V1(I))
      SUMU = SUMU+UU
      SUMH = SUMH+H1(I)
 203  CONTINUE
      UAVE = SUMU/3.0
      HAVE = SUMH/3.0
      IF(HAVE.EQ.0.) WRITE(6,57) ITER,II,TCONT
      IF(HAVE.LT.0.) WRITE(6,54) ITER,II,TCONT
   54 FORMAT(10X,"ERROR....HAVE IS LESS THAN ZERO",3X,"ITER =",I5,
     23X,"ELEMENT NO. =",I5,3X,"TCONT = ",F10.4)
      IF(UAVE.LE.0.) WRITE(6,58) ITER,II,TCONT
   57 FORMAT(10X,"ERROR...HAVE = 0.",3X,"ITER =",I5,3X,"ELEMENT NO.="
     2,I5,3X,"TCONT = ",F10.4)
   58 FORMAT(10X,"ERROR..UAVE= 0." ,3X,"ITER =",I5,3X,"ELEMENT NO. ="
     2,I5,3X,"TCONT = ",F10.4)
      IF(HAVE.LE.0..OR.UAVE.LE.0.) GO TO 4000
      F = 24.*VISC/(RHO*UAVE*HAVE)
      CALL ELEM
      CALL ASSEM
 20   CONTINUE
      IF(ICONT.EQ.1)GO TO 22
      DO 24 I=1,NNODE3
      RT(I) = R(I)
 24   CONTINUE
      CALL MLTPLY(P,PHIIN,RBN)
      DO 23 I=1,NNODE3
      RBN(I) = RBN(I)+TST*RT(I)
 23   CONTINUE
      CALL BOUND(P,RBN)
      CALL REDUCE(P,RBN)
      CALL SOLVE(P,RBN)
      DO 25 I=1,NNODE
      PHI(I) = RBN(I)
      PHI(NNODE+I) = RBN(NNODE+I)
      PHI((2*NNODE)+I) = RBN((2*NNODE)+I)
      H(I) = RBN(I)
      U(I) = RBN(NNODE+I)
      V(I) = RBN((2*NNODE)+I)
 25   CONTINUE
      ICONT = 1
      GO TO 128
 22   CONTINUE
      CALL MLTPLY(P,PHIIN,RBN)
      DO 40 I=1,NNODE3
      RBN(I) = RBN(I) + TST*(THETA*RT(I)+(1.-THETA)*R(I))
 40   CONTINUE
      CALL BOUND(P,RBN)
      CALL REDUCE(P,RBN)
      CALL SOLVE(P,RBN)
      CALL CONVRG(RBN,PHI,CONERR)
      ITER = ITER+1
      IF(ITER.GT.20)WRITE(6,11)
 11   FORMAT(10X,"CONVERGENCE PROBLEM CHECK")
      IF(ITER.GT.20)GO TO 4000
************RELAXATION STATEMENTS*******
      IF(CONERR.LE.ERR) GO TO 133
      DO 132 I=1,NNODE3
      RBN(I) = PHI(I) + WT*(RBN(I)-PHI(I))
```

```
  132 CONTINUE
  133 CONTINUE
******************** END OF RELAXATION**********
      DO 43 I=1,NNODE
      PHI(I) = RBN(I)
      PHI(NNODE+I) = RBN(NNODE+I)
      PHI((2*NNODE)+I) = RBN((2*NNODE)+I)
      H(I) = RBN(I)
      U(I) = RBN(NNODE+I)
      V(I) = RBN((2*NNODE)+I)
 43   CONTINUE
      IF(CONERR.GT.ERR)GO TO 128
      ITIM =ITIM+1
      F = FP
      IF(ITIM.NE.IP)GO TO 48
      CALL OUTPUT(ITER,F)
      ITIM = 0
 48   CONTINUE
      TCONT = TCONT+TST
      IF(TCONT.GT.TDUR)RO=0.
      IF(TCONT.GT.TF)GO TO 4000
      SUM1 = 0.0
      SUM2 = 0.0
      DO 50 I=1,NNODE
      HIN(I) = H(I)
      UIN(I) = U(I)
      VIN(I) = V(I)
      SUM1 = SUM1+H(I)
      SUM2 = SUM2+SQRT(U(I)*U(I)+V(I)*V(I))
 50   CONTINUE
      SUM1 = SUM1/NNODE
      SUM2 = SUM2/NNODE
      REY = SUM1*SUM2*RHO/VISC
      IF (REY.LE.0.) WRITE(6,55)
   55 FORMAT(10X,"ERROR....... REYNOLDS NO. = 0.")
      IF(REY.LE.0.) GO TO 4000
      FP = 24./REY
      F = FP
      ICONT = 0
      ITER = 0
      GO TO 128
 4000 CONTINUE
      STOP
      END


      SUBROUTINE CONVRG(RBN,PHI,CONERR)
      DIMENSION RBN(450),PHI(450)

      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC

      SUM1 = 0.0
      SUM2 = 0.0
      DO 50 I=1,NNODE3
      SUM1 = SUM1+(RBN(I)-PHI(I))**2.
      SUM2 = SUM2+PHI(I)**2.
 50   CONTINUE
      IF(SUM2.LE.0.)WRITE(6,11)
 11   FORMAT(10X,"ERROR IN CONVERGENCE ROUTINE")
      IF(SUM2.LE.0.) GO TO 55
      CONERR = SQRT(SUM1/SUM2)
   55 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BOUND(P,RBN)
      DIMENSION P(450,20),RBN(450),NDBCNU(50)
     2 ,NDBCNV(50),NDBCNH(50),BCDU(50),BCDV(50),
     3  BCDH(50)
      COMMON /BC/NDBCNU,BCDU,NDBCNV,BCDV,NDBCNH,BCDH
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      MHALF = (IUBW+1)/2
      IF(NDH.EQ.0)GO TO 10
      DO 1 I=1,NDH
      NK = NDBCNH(I)
      P(NK,MHALF) = P(NK,MHALF)*10.**25.
      RBN(NK) = RBN(NK)+P(NK,MHALF)*BCDH(I)
    1 CONTINUE
   10 CONTINUE
      IF(NDU.EQ.0)GO TO 20
      DO 2 I=1,NDU
      NK = NDBCNU(I)
      P((NNODE+NK),MHALF) = P((NNODE+NK),MHALF)*10.**25.
      RBN(NNODE+NK) = RBN(NNODE+NK)+P(NNODE+NK,MHALF)*BCDU(I)
    2 CONTINUE
   20 CONTINUE
      NNN = 2*NNODE
      IF(NDV.EQ.0)GO TO 30
      DO 3 I=1,NDV
      NK = NDBCNV(I)
      P(NNN+NK,MHALF) = P(NNN+NK,MHALF)*10.**25.0
      RBN(NNN+NK) = RBN(NNN+NK)+P(NNN+NK,MHALF)*BCDV(I)
    3 CONTINUE
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SET

      DIMENSION NEWN(3),XE(3),YE(3),X(2,150),ICON(3,300)
      COMMON /LCL/NEWN,XE,YE,X,ICON

      ORX = 0.
       ORY = 0.
      DO 1 I=1,3
      J =NEWN(I)
      ORX = ORX+X(1,J)
    1 ORY = ORY+X(2,J)
      ORX = ORX/3.0
      ORY = ORY/3.0
      DO 2 I=1,3
      J = NEWN(I)
      XE(I) =X(1,J)-ORX
    2 YE(I) = X(2,J)-ORY
      RETURN
      END
```

```
      SUBROUTINE ASSEM
      DIMENSION RE1(3),PE1(3,3),R(450),P(450,20),PE2(3,3),PE3(3,3),
     1RE2(3),RE3(3),U1(3),V1(3),H1(3),HIN(150),UIN(150),VIN(150),
     2PHI(450),H(150),U(150),V(150),NEWN(3),XE(3),YE(3),ICON(3,300)
     3,X(2,150)
      COMMON/MVEL/RE1,RE3,PE1,AREA,II,RE2,PE2,PE3,U1,V1,H1,UAVE,HAVE
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      COMMON /GMV/R,P,HIN,UIN,VIN,PHI,H,U,V
      COMMON /LCL/NEWN,XE,YE,X,ICON
      DO 1 I=1,3
      R(NEWN(I)) = R(NEWN(I))+RE1(I)
      R(NEWN(I)+NNODE) = R(NEWN(I)+NNODE)+RE2(I)
      R(NEWN(I)+(2*NNODE)) = R(NEWN(I)+(2*NNODE))+RE3(I)
    1 CONTINUE
      IB= (IUBW+1)/2

      DO 2 JJ=1,3
      DO 2 J=1,3
      NN = NEWN(JJ)
      KK = NEWN(J)
      LL = KK-NN
      P(NN,IB+LL) = P(NN,IB+LL)+PE1(JJ,J)
      P((NNODE+NN),(IB+LL))=P((NNODE+NN),(IB+LL))+PE2(JJ,J)
      P(((2*NNODE)+NN),(IB+LL))= P(((2*NNODE)+NN),(IB+LL))+PE3(JJ,J)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SOLVE(S,F)
      DIMENSION S(450,20),F(450)
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      MHALF = (IUBW+1)/2
      MHALF1 = MHALF+1
      F(NNODE3) = F(NNODE3)/S(NNODE3,MHALF)
      DO 1 M=2,NNODE3
      N = NNODE3+1-M
      DO 2 L=MHALF1,IUBW
      IF(S(N,L).EQ.0.)GO TO 2
      K = (N+L-MHALF)
      IF(K.GT.NNODE3)GO TO 1
      F(N) =F(N)-S(N,L)*F(K)
    2 CONTINUE
      F(N) = F(N)/S(N,MHALF)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REDUCE(S,F)
      DIMENSION S(450,20),F(450)
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      MHALF = (IUBW+1)/2
      MHALF1 = MHALF+1
      DO 1 N=1,NNODE3
      LL = MHALF
      DO 2 L=MHALF1,IUBW
      I = N+L-MHALF
      IF(I.GT.NNODE3)GOTO 1
      LL = LL-1
      IF(S(I,LL).EQ.0.)GO TO 2
      C = S(I,LL)/S(N,MHALF)
      J = LL
      DO 3 K=MHALF1,IUBW
      J = J+1
    3 S(I,J) =S(I,J)-C*S(N,K)
      F(I) = F(I)-F(N)*C
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MLTPLY(T,PHI,G)
      DIMENSION T(450,20),PHI(450),G(450)
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      MHALF = (IUBW+1)/2
      MHALF1 = MHALF+1
      DO 1 I=1,NNODE3
    1 G(I) =0.0
      K = MHALF1
      DO 2 I=1,MHALF
      K = K-1
      L = 1
      DO 2 J=K,IUBW
      G(I) = G(I)+T(I,J)*PHI(L)
      L = L+1
    2 CONTINUE
      K = 1
      MID = NNODE3-MHALF
      DO 3 I=MHALF1,MID
      K = K+1
      L = K
      DO 3 J=1,IUBW
      G(I) = G(I)+T(I,J)*PHI(L)
      L = L+1
    3 CONTINUE
      K = NNODE3-IUBW
      JJ = IUBW+1
      NREST = NNODE3-MHALF+1
      DO 4 I=NREST,NNODE3
      JJ = JJ-1
      K = K+1
      L = K
      DO 4 J=1,JJ
      G(I) = G(I)+T(I,J)*PHI(L)
      L = L+1
    4 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELEM
      DIMENSION NEWN(3),XE(3),YE(3),X(2,150),ICON(2,200),
     2 RE1(3),RE2(3),RE3(3),PE1(3,3),PE2(3,3),PE3(3,3),U1(3),
     3 V1(3),H1(3),A(3),B(3),C(3),IEL(300)

      COMMON /MVEL/RE1,RE3,PE1,AREA,II,RE2,PE2,PE3,U1,V1,H1,UAVE,HAVE
     COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      COMMON /ELM/F,RO,GR,SO,VISC,RHO,CMN
      COMMON /LCL/NEWN,XE,YE,X,ICON
     COMMON/TURB/IEL

     REY = UAVE*HAVE*RHO/VISC
     IEL(II) = 1
     IF(REY.LT.2000.) IEL(II) = 0
     IF(REY.GT.8000.) IEL(II) = 2
      AREA = ABS(((XE(2)*YE(3)-YE(2)*XE(3))-(XE(1)*YE(3)-
     2 XE(3)*YE(1))+(XE(1)*YE(2)-XE(2)*YE(1)))*0.5)

      A(1) = (XE(2)*YE(3)-XE(3)*YE(2))/(2.*AREA)
      A(2) = (XE(3)*YE(1)-XE(1)*YE(3))/(2.*AREA)
      A(3) = (XE(1)*YE(2)-XE(2)*YE(1))/(2.*AREA)
      B(1) = (YE(2)-YE(3))/(2.*AREA)
      B(2) = (YE(3)-YE(1))/(2.*AREA)
      B(3) = (YE(1)-YE(2))/(2.*AREA)
      C(1) = (XE(3)-XE(2))/(2.*AREA)
      C(2) = (XE(1)-XE(3))/(2.*AREA)
      C(3) = (XE(2)-XE(1))/(2.*AREA)
      PE1(1,1) = (AREA/12.)*2.
      PE1(1,2) = (AREA/12.)
      PE1(1,3) = (AREA/12.)

      PE1(2,1) = PE1(1,2)
      PE1(3,1) = PE1(1,3)
      PE1(2,2) = PE1(1,1)
      PE1(2,3) = PE1(1,3)
      PE1(3,2) = PE1(2,3)
      PE1(3,3) = PE1(2,2)

      DO 10 I=1,3
      DO 10 J=1,3
      PE2(I,J) = PE1(I,J)
      PE3(I,J) = PE1(I,J)
 10   CONTINUE
     AA = AREA/12.
     UU1 = (2.*U1(1)+U1(2)+U1(3))
     UU2 = (U1(1)+2.*U1(2)+U1(3))
     UU3 = (U1(1)+U1(2)+2.*U1(3))
     VV1 = (2.*V1(1)+V1(2)+V1(3))
     VV2 = (V1(1)+2.*V1(2)+V1(3))
     VV3 = (V1(1)+V1(2)+2.*V1(3))
     UC = (U1(1)*C(1)+U1(2)*C(2)+U1(3)*C(3))
     UB = (U1(1)*B(1)+U1(2)*B(2)+U1(3)*B(3))
     VC = (V1(1)*C(1)+V1(2)*C(2)+V1(3)*C(3))
     VB = (V1(1)*B(1)+V1(2)*B(2)+V1(3)*B(3))
     HB = H1(1)*B(1)+H1(2)*B(2)+H1(3)*B(3)
     HC = H1(1)*C(1)+H1(2)*C(2)+H1(3)*C(3)
     U11 = 6.*U1(1)+2.*U1(2)+2.*U1(3)
     U22 = 2.*U1(1)+6.*U1(2)+2.*U1(3)
     U33 = 2.*U1(1)+2.*U1(2)+6.*U1(3)
     U12 = 2.*U1(1)+2.*U1(2)+U1(3)
     U13 = 2.*U1(1)+U1(2)+2.*U1(3)
     U23 = U1(1)+2.*U1(2)+2.*U1(3)
     V11 = 6.*V1(1)+2.*V1(2)+2.*V1(3)
     V22 = 2.*V1(1)+6.*V1(2)+2.*V1(3)
     V33 = 2.*V1(1)+2.*V1(2)+6.*V1(3)
     V12 = 2.*V1(1)+2.*V1(2)+V1(3)
     V13 = 2.*V1(1)+V1(2)+2.*V1(3)
     V23 = V1(1)+2.*V1(2)+2.*V1(3)
```

```
      RE1(1) = ((RO*AREA)/3.)-AA*((HB*UU1)+((2.*H1(1)+H1(2)+H1(3))
     2*UB)+(HC*VV1)+((2.*H1(1)+H1(2)+H1(3))*VC))
      RE1(2) = ((RO*AREA)/3.)-AA*((HB*UU2)+((H1(1)+2.*H1(2)+H1(3))
     2*UB)+(HC*VV2)+((H1(1)+2.*H1(2)+H1(3))*VC))
      RE1(3) = ((RO*AREA)/3.)-AA*((HB*UU3)+((H1(1)+H1(2)+2.*H1(3))
     2*UB)+(HC*VV3)+((H1(1)+H1(2)+2.*H1(3))*VC))
      RBA = (RO*AREA)/(12.*HAVE)
      RE2(1) =-(RBA*UU1)+(GR*SO*AREA/3.)-((GR*AREA/3.)*HB)
     3-(AA*UB*UU1)-(AA*UC*VV1)
      RE2(2) =-(RBA*UU2)+(GR*SO*AREA/3.)-((GR*AREA/3.)*HB)
     3-(AA*UB*UU2)-(AA*UC*VV2)
      RE2(3) =-(RBA*UU3)+(GR*SO*AREA/3.)-((GR*AREA/3.)*HB)
     3-(AA*UB*UU3)-(AA*UC*VV3)
      RE3(1) =-(RBA*VV1)+(GR*SO*AREA/3.)-((GR*AREA/3.)*HC)
     3-(AA*VB*UU1)-(AA*VC*VV1)
      RE3(2) =-(RBA*VV2)+(GR*SO*AREA/3.)-((GR*AREA/3.)*HC)
     3-(AA*VB*UU2)-(AA*VC*VV2)
      RE3(3) =-(RBA*VV3)+(GR*SO*AREA/3.)-((GR*AREA/3.)*HC)
     3-(AA*VB*UU3)-(AA*VC*VV3)
      DO 150 JM=1,3
      SUMNU=0.
      SUMNV=0.
      DO 160 JL=1,3
      SUMNU=SUMNU+AREA*(B(JL)*B(JM)+C(JL)*C(JM))*U1(JL)
      SUMNV=SUMNV+AREA*(B(JL)*B(JM)+C(JL)*C(JM))*V1(JL)
  160 CONTINUE
      RE2(JM) = RE2(JM) + SUMNU*VISC/RHO
      RE3(JM) = RE3(JM) + SUMNV*VISC/RHO
  150 CONTINUE
      IF(IFC.EQ.0) GO TO 500
      IF(IFC.EQ.2) GO TO 400
      IF(IEL(II).EQ.1) GO TO 400
      IF(IEL(II).EQ.2) GO TO 400
      IF(IEL(II).EQ.0) GO TO 500
  400 CONTINUE
      CK = (1.49/CMN)*(HAVE**(1./6.))
      FBA = (32.2*UAVE)/(HAVE*CK*CK)
      FBA = FBA*AREA/12.
      RE2(1) = RE2(1) - FBA*UU1
      RE2(2) = RE2(2) - FBA*UU2
      RE2(3) = RE2(3) - FBA*UU3
      RE3(1) = RE3(1) - FBA*VV1
      RE3(2) = RE3(2) - FBA*VV2
      RE3(3) = RE3(3) - FBA*VV3
      GO TO 700
  500 CONTINUE
      FBA = (F*AREA)/(480.*HAVE)
      RE2(1) = RE2(1) - FBA*(U1(1)*U11+U1(2)*U12+U1(3)*U13)
      RE2(2) = RE2(2) - FBA*(U1(1)*U12+U1(2)*U22+U1(3)*U23)
      RE2(3) = RE2(3) - FBA*(U1(1)*U13+U1(2)*U23+U1(3)*U33)
      RE3(1) = RE3(1) - FBA*(V1(1)*V11+V1(2)*V12+V1(3)*V13)
      RE3(2) = RE3(2) - FBA*(V1(1)*V12+V1(2)*V22+V1(3)*V23)
      RE3(3) = RE3(3) - FBA*(V1(1)*V13+V1(2)*V23+V1(3)*V33)
  700 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE INPUT
      DIMENSION NDBCNU(50),BCDU(50),NDBCNV(50),BCDV(50),NDBCNH(50)
     2 ,BCDH(50),TITLE(20),NEWN(3),XE(3),YE(3),X(2,150),ICON(3,300)
     3 ,HIN(150),UIN(150),VIN(150),P(450,20),R(450),PHI(450)
     4 ,H(150),U(150),V(150)
      COMMON /BC/NDBCNU,BCDU,NDBCNV,BCDV,NDBCNH,BCDH
      COMMON /TIME/TDUR,TITLE,TI,TF,TST,TCONT,IP
      COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      COMMON /LCL/NEWN,XE,YE,X,ICON
      COMMON /ELM/F,RO,GR,SO,VISC,RHO,CMN
      COMMON /GMV/R,P,HIN,UIN,VIN,PHI,H,U,V
      WRITE(6,1)
1    FORMAT(//,10X,"THIS PROGRAM WAS PREPARED AND SUBMITTED IN PARTIAL
     1FULFILLMENT OF PROJECT NO: E20-613",/,13X,"BETWEEN BELOIT CORPORAT
     2ION AND GEORGIA INSTITUTE OF TECHNOLOGY, ATLANTA GEORGIA",//,
     314X,"THE PROGRAM IS PREPARED BY DR. MUSTAFA M. ARAL OF SCHOOL OF
     4CIVIL ENGINEERING",/,36X," GEORGIA INSTITUTE OF TECHNOLOGY",///)
      WRITE(6,2)
2     FORMAT(//,10X,"THE PROGRAM IS LAST UPDATED ON JUNE.3.1983",//)
      READ (5,3) TITLE
3     FORMAT(20A4)
      WRITE(6,4)TITLE
4     FORMAT(1H1,10X,20A4,//,10X,"GENERATED NODAL PATTERN AND DATA",/)
      READ(5,16) IFC,NNPC,NELEMC,NDH,NDU,NDV,HCONS,UCONS,VCONS,WT
   16 FORMAT(6I5,4F10.0)
      WRITE(6,17) NNPC,NELEMC,NDH,NDU,NDV,HCONS,UCONS,VCONS,WT,IFC
17    FORMAT(10X,"NNPC =",I4,3X,"NELEMC =",I4,3X,"NDH =",I4,
     1 3X,"NDU =",I4,3X,"NDV =",I4,3X,"HCONS = ",F10.5,3X,
     2"UCONS = ",F10.5,3X,"VCONS = ",F10.5//10X,"WT = ",F10.4,3X,
     3"IFC = ",I3,"****","0 = USES MOODY CHART, 2 = USES CHEZY FORMULA,
     4 1 = LEFT TO THE PROGRAM",///)
      WRITE(6,58)
   58 FORMAT(10X,"IF (HCONS,UCONS,VCONS) ARE ASSIGNED A ZERO VALUE"
     2" THEN A DISTRIBUTION OF THESE VARIABLES MUST BE GIVEN"
     3" AS INPUT DATA."/10X,"IF A CONSTANT OTHER THEN ZERO IS ASSIGNED"
     4" THEN THAT VALUE IS AUTOMATICALLY ASSIGNED TO ALL NODES BY THE "
     5 "CODE"///)
      NCC = 0
180   NCC = NCC+1
      IF(NCC.GT.NNPC) GO TO 215
      READ(5,190) N,X(1,N),X(2,N),NPMIS,HIN(N),UIN(N),VIN(N)
190   FORMAT(I10,2F10.4,I10,3F10.0)
      WRITE(6,200) N,X(1,N),X(2,N),HIN(N),UIN(N),VIN(N)
200   FORMAT(1H,13X,4HNODE,I4,5X,3HX= ,F10.4,5X,3HY= ,F10.4,5X,
     15HHIN= ,F10.8,5X,5HUIN= ,F10.6,3X,5HVIN= ,F10.6)
      NI = N
      IF(NPMIS.NE.0) GO TO 210
      GO TO 180
210   NCC = NCC+1
      IF(NCC.GT.NNPC) GO TO 215
      READ(5,190) N,X(1,N),X(2,N),NPMIS,HIN(N),UIN(N),VIN(N)
      NE = N
      NPG = (NE-NI)
      DX = (X(1,NE)-X(1,NI))/FLOAT(NPG)
      DY = (X(2,NE)-X(2,NI))/FLOAT(NPG)
```

```
      DS1=SQRT(DX*DX+DY*DY)
      DS2=SQRT((X(1,NE)-X(1,NI))**2.+(X(2,NE)-X(2,NI))**2.)
      DHH=HIN(NE)-HIN(NI)
      DUU=UIN(NE)-UIN(NI)
      DVV = VIN(NE)-VIN(NI)
      DU2=DUU*DS1/DS2
      DH2=DHH*DS1/DS2
      DV2 = DVV*DS1/DS2
       DO 214 IJ = 1,NPG
       I = IJ
       NG = NI+I
       X(1,NG) = X(1,NI)+FLOAT(I)*DX
       X(2,NG) = X(2,NI)+FLOAT(I)*DY
      HIN(NG)=HIN(NI)+FLOAT(I)*DH2
      UIN(NG)=UIN(NI)+FLOAT(I)*DU2
      VIN(NG) = VIN(NI) + FLOAT(I)*DV2
       WRITE(6,200) NG,X(1,NG),X(2,NG),HIN(NG),UIN(NG),VIN(NG)
 214   CONTINUE
       IF(NPMIS.EQ.0)GO TO 180
       NI = N
       GO TO 210
 215   NNP = N
       NNODE = NNP
       IF(NNP.LE.150)GO TO 219
       WRITE(6,216)
 216   FORMAT(1H1,10X,"ERROR...NUMBER OF NODES IS GREATER THAN 150")
       GO TO 1000
 219   N = 0
       DO 222 M=1,NELEMC
       N = N+1
       READ (5,220) (ICON(IJ,N),IJ=1,3),NMIS
 220   FORMAT(3I5,I10)
       IF(NMIS.EQ.0)GO TO 222
      DO 201 LL=1,2
      IF(LL.EQ.1) GOTO 213
      N=N+1
      ICON(1,N)= ICON(2,(N-(NMIS+1)))
      ICON(2,N) = ICON(1,N)+1
      ICON(3,N) = ICON(1,(N-(NMIS+1)))+1
  213 CONTINUE
       DO 221 K=1,NMIS
       N = N+1
       DO 221 J=1,3
       ICON(J,N) = ICON(J,N-1)+1
 221   CONTINUE
  201 CONTINUE
 222   CONTINUE
       NELEM = N
       IF(NELEM.LE.300)GO TO 224
       WRITE(6,223)
 223   FORMAT(1H1,10X,"ERROR.....NUMBER OF ELEMENTS GREATER THAN 300")
       GO TO 1000
 224   WRITE(6,225)NNP,NELEM
 225   FORMAT(1H1,10X,"NUMBER OF NODES=",I4,3X,"NUMBER OF ELEMENTS=",
     2 I4,///,8X,4(8HELE. NO.,3X,12HNODAL POINTS,8X),//)
*********************
      ICON(2,(NELEM-NMIS-1)) =ICON(2,NELEM)
      ICON(3,NELEM) = ICON(1,(NELEM-NMIS-1))
      ICON(2,1) = ICON(2,(NMIS+2))
      ICON(3,NMIS+2) = ICON(1,1)
*********************
```

```
229   WRITE(6,230)(N,(ICON(IJ,N),IJ=1,3),N=1,NELEM)
230   FORMAT(4(8X,I5,3X,3I5))

      READ(5,6) F,GR,SO,RO,ERR,RHO,VISC,THETA,CMN
  6   FORMAT(5F8.0,2E12.5,2F6.0)
      WRITE(6,12)F,GR,SO,RO,ERR,RHO,VISC,THETA,CMN
 12    FORMAT(//,3X,"FRICTION COEF =",E10.5,7X,"GRAV.ACC. ="
    2 ,E12.4,6X,"SLOPE =",E11.4,8X,"CONDENSATION RATE ="
    3 ,E11.5,//,3X,"ERR =",F10.5,17X,"DENSITY =",E10.5,
    410X,"VISCOSITY =",E10.5,5X,"THETA =",F6.3//,3X,"MANNING N = "
    5,F10.5/)

      READ(5,19) TI,TF,TST,TDUR,IP
      WRITE(6,18) TI,TF,TST,TDUR,IP
 19   FORMAT(4F10.0,I10)
 18   FORMAT(//3X,"TI = ",3X,E12.4,3X,"TF = ",3X,E12.4,3X,"TST = ",
    2 3X,E12.4,3X,"TDUR = ",3X,E12.4,3X,"PRINTOUT INTERVAL = ",3X,I5//)
      WRITE(6,21)
 21   FORMAT(10X,"BOUNDARY CONDITION DATA"//)
     IF(NDH.NE.0) WRITE(6,62)
      IF(NDH.NE.0)READ(5,22) ((NDBCNH(I),BCDH(I)),I=1,NDH)
      IF(NDH.NE.0)WRITE(6,23) ((NDBCNH(I),BCDH(I)),I=1,NDH)
     IF(NDU.NE.0) WRITE(6,63)
      IF(NDU.NE.0) READ(5,22) ((NDBCNU(I),BCDU(I)),I=1,NDU)
      IF(NDU.NE.0) WRITE(6,23) ((NDBCNU(I),BCDU(I)),I=1,NDU)
     IF(NDV.NE.0) WRITE(6,64)
      IF(NDV.NE.0) READ(5,22) ((NDBCNV(I),BCDV(I)),I=1,NDV)
      IF(NDV.NE.0) WRITE(6,23) ((NDBCNV(I),BCDV(I)),I=1,NDV)
 62 FORMAT(/5X,"BOUNDARY CONDITIONS ON DEPTH"//2(6X,"NODE",6X,
    2"DEPTH")/)
 63 FORMAT(/5X,"BOUNDARY CONDITIONS ON X-COMPONENT VELOCITY"//
    2,2(6X,"NODE",3X,"VELOCITY")/)
 64 FORMAT(/5X,"BOUNDARY CONDITIONS ON Y-COMPONENT VELOCITY"//
    2,2(6X,"NODE",3X,"VELOCITY")/)

 22   FORMAT(8(I3,F7.0))
 23   FORMAT(2(I10,1X,F10.4))
     DO 500 I=1,NNODE
     IF(HCONS.EQ.0.0) GO TO 36
     HIN(I) = HCONS
  36 CONTINUE
     IF(UCONS.EQ.0.) GO TO 37
     UIN(I) = UCONS
  37 CONTINUE
     IF(VCONS.EQ.0.) GO TO 38
     VIN(I) = VCONS
  38 CONTINUE
 500 CONTINUE
     IF(NDH.EQ.0) GO TO 510
     DO 509 I=1,NDH
     MM = NDBCNH(I)
     HIN(MM) = BCDH(I)
 509 CONTINUE
 510 CONTINUE
     IF(NDU.EQ.0) GO TO 520
     DO 519 I=1,NDU
     MM = NDBCNU(I)
     UIN(MM) = BCDU(I)
 519 CONTINUE
 520 CONTINUE
```

```
      IF(NDV.EQ.0) GO TO 530
      DO 529 I=1,NDV
      MM = NDBCNV(I)
      VIN(MM) = BCDV(I)
  529 CONTINUE
  530 CONTINUE
       IUBW = 0
      DO 602 N=1,NELEM
       DO 600 J=1,3
       JJ = J+1
       IF(JJ.GT.3) JJ=1
       IUBWS = IABS(ICON(J,N)-ICON(JJ,N))
       IF(IUBWS.GT.IUBW) IUBW=IUBWS
 600   CONTINUE
  602 CONTINUE
       IUBW = 2*IUBW+1
       WRITE(6,601) IUBW
 601   FORMAT(///10X,"IUBW = ",I10//)
 1000  CONTINUE
       RETURN
       END

       SUBROUTINE OUTPUT (ITER,F)
       DIMENSION R(450),P(450,20),HIN(150),UIN(150),VIN(150)
     2 ,PHI(450),H(150),U(150),V(150),TITLE(20),IEL(300)
      COMMON/TURB/IEL
       COMMON /GMV/R,P,HIN,UIN,VIN,PHI,H,U,V
       COMMON/CHAR/NNODE,NELEM,NDH,NDU,NDV,ERR,NNODE3,IUBW,THETA,WT,IFC
      COMMON/TIME/TDUR,TITLE,TI,TF,TST,TCONT,IP
       WRITE(6,1) TCONT,ITER
   1   FORMAT(//10X,"RESULTS OF THE PROBLEM AT TIME =",F15.8,
     2 5X,"NUMBER OF ITERATIONS =",I4)
       WRITE(6,2)
   2   FORMAT(/4(5X,"NODE",4X,"THICKNESS, H(I)"))
       CONTINUE
       WRITE(6,3) ((I,PHI(I)),I=1,NNODE)
   3   FORMAT(4(5X,I3,4X,E16.9))
       WRITE(6,4)
   4   FORMAT(/4(5X,"NODE",4X,"VELOCITY , U(I)"))
       NNODE2 = NNODE*2
       J = NNODE + 1
       WRITE(6,3) ((I-NNODE),PHI(I),I=J,NNODE2)
        WRITE(6,5)
   5   FORMAT(/4(5X,"NODE",4X,"VELOCITY , V(I)"))
       J = NNODE2+1
       WRITE(6,3) (((I-NNODE2),PHI(I)),I =J,NNODE3)
       SUM = 0.
       DO 9 J=1,NNODE
   9   SUM = SUM+PHI(J)
       SUM = SUM/FLOAT(NNODE)
       WRITE(6,6) SUM
   6   FORMAT(/5X,"AVERAGE THICKNESS FOR THIS TIME IS = "
     2 ,E16.9/)
       SUM = 0.
       L = NNODE+1
       DO 7 J=L,NNODE2
   7   SUM = SUM+SQRT(PHI(J)**2.+PHI(J+NNODE)**2.)
       SUM = SUM/FLOAT(NNODE)
       REY = 24./F
       WRITE(6,8) SUM,F,REY
   8   FORMAT(5X,"AVERAGE VELOCITY FOR THIS TIME IS = "
     2 ,E16.9,3X,"FRICTION COEFF. = ",F10.4,3X,"REYNOLDS NO. = ",
     3 E10.5/)
      WRITE(6,20)
   20 FORMAT(/3X,"CODE=0, LAMINAR - CODE=1. TRANS. - CODE=2, TURB. FL"
     2,//,7(3X,"ELEM NO.",3X,"CODE"))
      WRITE(6,21) ((I,IEL(I)),I=1,NELEM)
   21 FORMAT(7(5X,I3,6X,I4))
       RETURN
       END
```