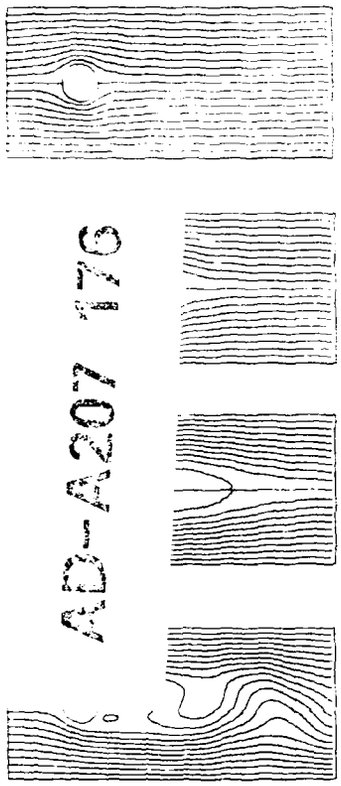
DTIC FILE COPY

AD-A207 176

US Army Corps of Engineers



REPAIR, EVALUATION, MAINTENANCE, AND REHABILITATION RESEARCH PROGRAM

TECHNICAL REPORT REMR-HY-5

EXPLICIT NUMERICAL ALGORITHM FOR MODELING INCOMPRESSIBLE APPROACH FLOW

by

Robert S. Bernard

Hydraulics Laboratory

DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
PO Box 631, Vicksburg, Mississippi 39181-0631

March 1989
Final Report

Approved For Public Release; Distribution Unlimited

DTIC
ELECTE
APR 28 1989
S H D

Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000
Under Civil Works Research Work Unit 32319

131

The following two letters used as part of the number designating technical reports of research published under the Repair, Evaluation, Maintenance, and Rehabilitation (REMR) Research Program identify the problem area under which the report was prepared

	Problem Area		Problem Area
CS	Concrete and Steel Structures	EM	Electrical and Mechanical
GT	Geotechnical	EI	Environmental Impacts
HY	Hydraulics	OM	Operations Management
CO	Coastal		

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

This program is furnished by the Government and is accepted and used by the recipient with the express understanding that the United States Government makes no warranties, expressed or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof. The program belongs to the Government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent this program to anyone as other than a Government program.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

COVER PHOTOS:

Computer drawings of streamlines for flow past a circular cylinder

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY Unclassified	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Technical Report REMR-HY-5	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION USAEWES Hydraulics Laboratory 6b. OFFICE SYMBOL (If applicable) CEWES-HS-R	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) PO Box 631 Vicksburg, MS 39181-0631	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION US Army Corps of Engineers 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Washington, DC 20314-1000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Explicit Numerical Algorithm for Modeling Incompressible Approach Flow

12. PERSONAL AUTHOR(S)
Bernard, Robert S.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final report	FROM _____ TO _____	March 1989	72

16. SUPPLEMENTARY NOTATION A report of the Hydraulics Problem Area of the Repair, Evaluation, Maintenance, and Rehabilitation (REMR) Research Program. Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Approach Channels, Mathematical Models, Hydraulic Models, Shallow Water, Incompressible Flow, Subcritical Flow

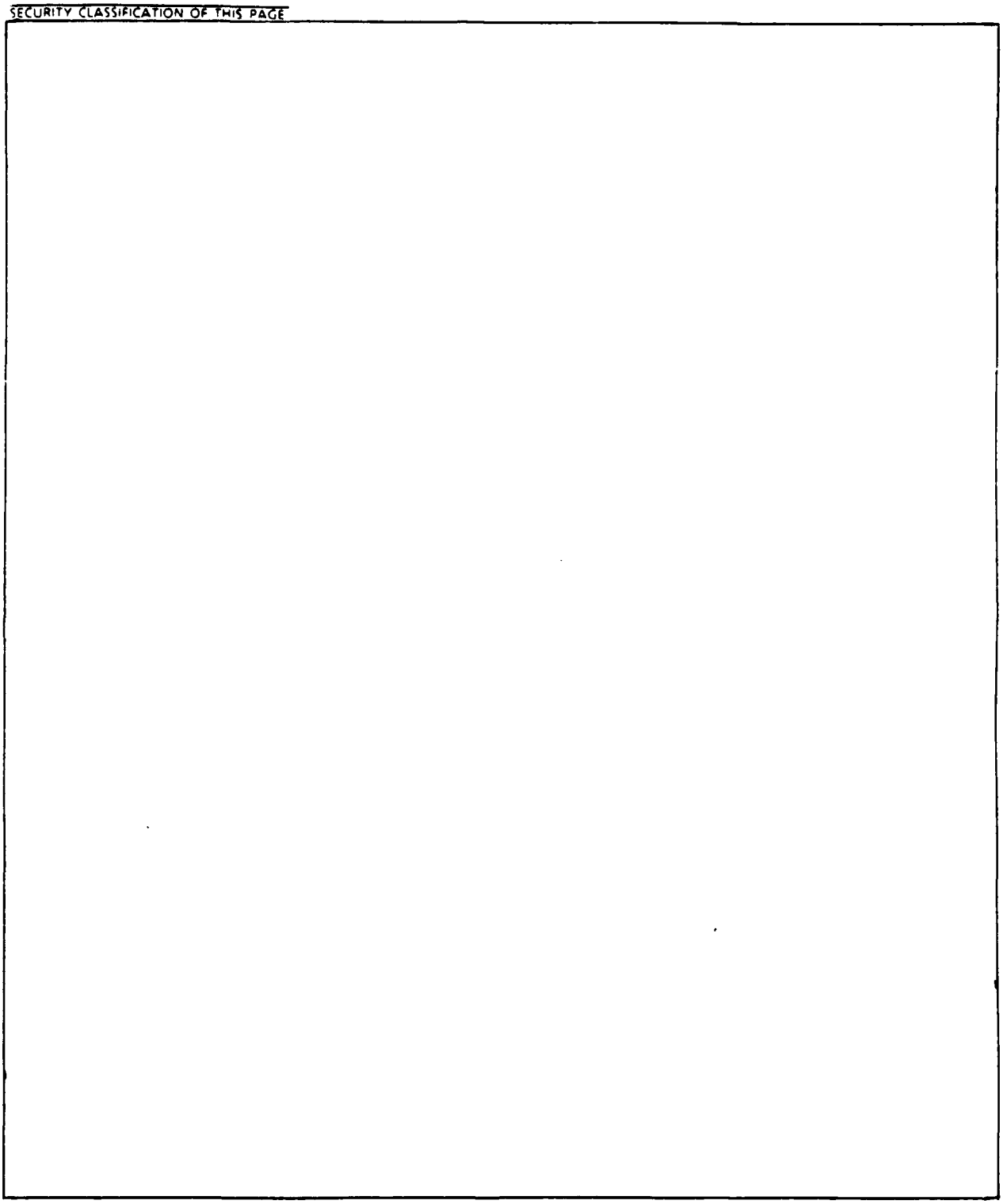
19. ABSTRACT (Continue on reverse if necessary and identify by block number)

An explicit finite-difference algorithm is presented which is to be used for numerical modeling of incompressible approach flow near hydraulic structures. The existing formulation is a predictor-corrector scheme derived from MacCormack's method. As such, it is applicable for transient flow at low Froude number, and for steady-state subcritical flow at moderate Froude number. The algorithm has been incorporated in the STREMR computer program, which will allow engineers and scientists to simulate the effects of structural modifications and repair measures prior to physical testing. The results of essential benchmark calculations are included herein for verification.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

Unclassified



PREFACE

This study was conducted from October 1984 through September 1987 by personnel of the Reservoir Water Quality Branch (RWQB), Hydraulic Structures Division (HSD), Hydraulics Laboratory (HL), US Army Engineer Waterways Experiment Station (WES), under the Repair, Evaluation, Maintenance, and Rehabilitation (REMR) research program sponsored by the Headquarters, US Army Corps of Engineers (USACE). Funding was provided under Civil Works Research Work Unit 32319, "Predictive Techniques for Approach Flow to Spillways and Other Structures."

The work was conducted by Dr. Robert S. Bernard, RWQB, under the general supervision of Messrs. F. A. Herrmann, Jr., Chief, HL; and Richard A. Sager, Assistant Chief, HL; Glenn A. Pickering, Chief, HSD: and Jeffery P. Holland, Chief, RWQB. Messrs. Michael L. Schneider and Rutherford C. Berger, RWQB, provided technical advice and counsel. Dr. Hartmut Kapitza, Mississippi State University (MSU), Mississippi State, Mississippi, developed a discrete Poisson solver under the supervision of Dr. Joe F. Thompson, Professor of Aerospace Engineering, MSU.

The REMR Directorate of Research and Development Coordinator in USACE was Mr. Jesse A. Pfeiffer, Jr., and members of the REMR Overview Committee, USACE, were Mr. James E. Crews, Chairman, and Dr. Tony C. Liu. The REMR Technical Monitor for Hydraulics was Mr. Glen Drummond, USACE. The REMR Program Manager at WES was Mr. William F. McCleese, Structures Laboratory, and the Problem Area Leader was Mr. Pickering. This report was written by Dr. Bernard and edited by Mrs. Marsha C. Gay, Information Technology Laboratory, WES.

COL Dwayne G. Lee, EN, is the Commander and Director of WES. Dr. Robert W. Whalin is the Technical Director.

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

CONTENTS

CONVERSION FACTORS, NON-SI TO SI (METRIC) UNITS OF MEASUREMENT

Non-SI units of measurement used in this report can be converted to SI (metric) units as follows:

Multiply	by	To Obtain
cubic feet	0.02832	cubic metres
degrees	0.01745	radians
feet	0.3048	metres
inches	2.54	centimetres
square feet	0.09290304	square metres

EXPLICIT NUMERICAL ALGORITHM FOR MODELING INCOMPRESSIBLE APPROACH FLOW

PART I: INTRODUCTION

Background

1. Adverse approach flow can increase the frequency of repair and the cost of maintenance while reducing the working life of a hydraulic structure. In many cases it is less expensive to modify the approach than to pay the long-term costs that would result from continued operation under existing flow conditions. Accordingly, the modifications themselves must be chosen to achieve the greatest benefit with the simplest and least expensive remedial measures.

2. In the past, each measure under consideration usually had to be tested in the laboratory with a physical scale model. While such testing is essential for the ultimate renovation, repair, and maintenance of a structure, it is costly and time-consuming in the early stages of evaluation; i.e., when there are many candidate measures from which the best are to be selected. In addition, laboratory modeling may be of limited value in some cases, due to scale differences between models and full-scale structures. Thus, it is expedient to have alternative means available that can help to eliminate errors introduced by scale effects.

3. A new computer program, STREMR, is under development to aid scientists, mathematicians, and engineers in simulating complex approach flows. The STREMR code is a finite-difference numerical model for two-dimensional (2-D) depth- or width-averaged flow with boundaries and obstacles of arbitrary shape. By changing the input and the finite-difference grid, which is generated by a separate computer program, WESCOR (Thompson 1983), the user can investigate the relative effects of such parameters and constraints as

a. Shape of boundaries (bank lines)
b. Vertical or lateral topography
c. Friction (Manning's n)
d. Dikes and training walls
e. Piers and piles

f. Vanes and berms

g. Inlets and outlets

h. Islands

i. Submerged structures

4. The ultimate purpose of STREMR is to provide guidance and reinforcement for physical models, but not to replace them. In fact, the greatest benefit can be achieved at the least expense by using the two types of models interactively rather than separately. Thus, the following sequence of operations is recommended:

a. Use STREMR for screening and preliminary evaluation; that is, to study the effects of as many repair measures and approach modifications as time allows.

b. Select the best of these to be tested in the laboratory with physical scale models.

c. If additional refinements are needed, use STREMR to study those refinements before continuing with physical testing.

d. Select final specifications for modifications and repair measures (final design), based on refinements in the laboratory.

5. The existing STREMR code uses an explicit finite-difference scheme to solve the depth- or width-averaged equations of motion for 2-D incompressible flow. The algorithm in its present form is limited to calculations at low Froude number (less than 0.3) for transient flow, but it is applicable at somewhat higher Froude numbers (perhaps up to 0.8) for subcritical steady-state flow. In any case, the scheme is definitely inappropriate for critical and supercritical flow, because it cannot accommodate sharp changes in the elevation of the water surface. This does not seem a great limitation, since many approach flows are subcritical flows at moderate Froude number (less than 0.5). Even if the flow does become critical, STREMR can still model the subcritical upstream region.

Purpose and Scope

6. The purpose of this report is to document and demonstrate the numerical algorithm used in STREMR. Part II offers a brief description of the methods employed, with the mathematical details given in appendices. Part III enumerates the results of benchmark tests used to verify the algorithm and assess its limitations. Part IV discusses the strengths and weaknesses of STREMR, and Part V outlines plans for continued development and improvement of

the algorithm and the STREMR code. Appendix A presents the governing equations, and Appendix B describes the predictor-corrector scheme. Appendix C discusses the boundary conditions, and Appendix D is the user's manual for the preliminary STREMR code.

PART II: GENERAL DESCRIPTION OF ALGORITHM

7. The fundamental assumption in the STREMR code is that the flow upstream of a hydraulic structure can be described by the depth- or width-averaged equations of motion for an incompressible fluid. In addition, it is assumed that the displacement of the free surface from its initial elevation is small, which means the time derivative of the free-surface elevation is disregarded. With these constraints, the computational free surface becomes a rigid lid, and the pressure existing there is approximately equal to the displacement of the actual free surface. This approximation is exact at zero Froude number (Fr = 0),* but it gradually becomes inexact as the Froude number increases toward unity (Fr = 1). At small Froude numbers (Fr < 0.3), the rigid-lid approximation is adequate even for transient flow, because pressure changes (changes in the free surface) propagate much faster than changes in circulation (vorticity). At moderate Froude numbers (Fr < 0.5), vorticity changes may propagate with speeds comparable to those for pressure changes, but the steady-state velocities will be essentially the same for the rigid-lid flow and the free-surface flow. The approximation breaks down completely as the flow approaches critical (Fr = 1).

8. The governing equations for incompressible flow are the equations for conservation of momentum and mass, which are discussed in Appendix A. These equations are discretized on a curvilinear grid using a finite-volume approximation for first derivatives and a finite-difference approximation for second derivatives. The discrete momentum equations are solved with a predictor-corrector scheme (Bernard 1986) derived from MacCormack's method (MacCormack 1969). Each time-step consists of a predictor phase and a corrector phase, which are combined to advance the developing flow by one time increment. The discrete equation for conservation of mass (the continuity equation) is satisfied in each predictor and corrector phase by introducing a pressure gradient, which is obtained in turn from the solution to a Poisson equation for pressure. The discrete Poisson equation is solved by means of an iterative procedure based on the conjugate-gradient technique (Kapitza and Eppel 1985). The predictor-corrector scheme is applied sequentially, time-

---

* The Froude number Fr is the ratio of flow velocity to free-surface wave velocity.

step after time-step, to march the evolving flow through time toward the steady state. For some conditions there may be no steady state, in which case the computation simply arrives at a periodic solution, such as vortex shedding. Details of the entire algorithm are given in Appendix B.

9. The STREMR code can make calculations in either the horizontal or the vertical plane, depending on the particular application. For computations in the horizontal, bottom friction (as represented by Manning's equation) is the dominant resisting force. Additional resistance may arise from shear stress (due to the no-slip condition) along the edges of the flow field and obstacles therein, but this will be far outweighed by the bottom friction when the lateral and longitudinal dimensions are much greater than the vertical depth. For calculations in the vertical plane, bottom friction gives way to sidewall friction, which is not important unless the lateral dimension of the flow (normal to the plane of motion) is small compared to the longitudinal and vertical dimensions. In such cases, the no-slip condition along edges and obstacles is likely to generate most of the resistance to the flow.

10. The effects of turbulence have not yet been incorporated into the STREMR code, and much effort will certainly be exerted toward that purpose in the near future. While STREMR is admittedly incomplete without a turbulence model, however, a turbulence model misapplied can be worse than no turbulence model at all. Moreover, the complexity of some approach flows may be enough to confound the most sophisticated turbulence models even in the most experienced hands. Faced with this kind of uncertainty, the sensible solution is to adopt a model that allows the user to set reasonable bounds on the influence of turbulence. In the existing version of STREMR, that can be accomplished by changing the fluid viscosity; but this demands judgment on the part of the user, and it gives rather dubious results at best. A turbulence algorithm will soon be incorporated that automatically accounts for the gross effects without expending too much computer time on the details.

11. The proper formulation of boundary conditions is crucial to fluid flow calculations, and much care has been given to this facet of algorithm development in STREMR. The existing code accommodates four types of boundaries in the plane of the flow:

a. Flux boundaries

b. Open boundaries

c. Slip boundaries

d. No-slip boundaries

12. A flux boundary has a fixed distribution of mass flux normal thereto; e.g., solid boundaries (zero flux) or inlets and outlets with a specified flux distribution. Open boundaries are like flux boundaries, except that the distribution of flux through them may change in the course of time, depending on the evolution of the computational flow field. A slip boundary is a frictionless solid boundary, which offers no tangential resistance to the flow. A no-slip boundary is a solid boundary that exerts shear stress on the flow due to the no-slip condition (zero tangential velocity) along the boundary.

13. The evolution of mass flux through open boundaries with time is found by means of a discrete radiation condition (Orlanski 1976), which allows changes inside the flow field to be transmitted outward, but not vice versa. Thus, a disturbance (such as a moving eddy) created inside the computational domain can leave the domain through an open boundary, but it cannot reenter. The discrete radiation condition allows calculations to be made with fewer grid cells (i.e., less storage and computer time) than might otherwise be necessary with fixed flux conditions. Most computations will entail both flux boundaries and open boundaries, with flux boundaries usually occurring at inlets and open boundaries at outlets.

14. The flux boundary conditions are specified independent of the pressure and are adjusted to maintain a fixed flow rate, which can be specified by the user or computed by STREMR. On rectangular grids and orthogonal curvilinear grids, the pressure is adjusted in a straightforward manner on the boundaries to accommodate the independently specified changes in mass flux. Nonorthogonal grids require additional measures for the pressure boundary condition (Bernard 1987).

15. The treatment of all boundary conditions is handled automatically by the STREMR code, and the user need only specify the boundary types (flux, open, slip, or no-slip). Appendix C offers a detailed mathematical discussion of the formulation and resolution of the boundary conditions, including the no-slip condition.

PART III: BENCHMARK CALCULATIONS

16. Presented in this part are the results of four sets of computer calculations, each of which represents a simple flow that might occur upstream of a hydraulic structure. The purpose here is to demonstrate both the assets and the shortcomings of the existing STREMR code, with an eye toward improving the shortcomings in the near future. In any case, to avoid wasted effort and erroneous computation, it is important that prospective users have a keen awareness of the code's limitations.

17. The first set of calculations involves a straight channel with a nonuniform semitrapezoidal cross section, showing the coupled effects of bathymetry and bottom friction. The second set entails a bendway in a channel with a uniform trapezoidal cross section, indicating the need for additional resistance forces (possibly arising from secondary flow) in the mathematical model. The third set follows the time evolution of the flow about a circular cylinder, demonstrating the code's ability to model transient flow at low Froude number. The fourth set of calculations, intended as a practical engineering example, shows the effect of reshaping an abutment in a hypothetical approach flow. All linear dimensions are given in feet, and all velocities in feet per second. The initial condition for all configurations is potential flow.

Flow in a Straight Channel

18. Figure 1 (Ruff et al. 1987) shows the schematic drawing of a tilting flume used to study riprap stability in flood-control channels. In the following discussion, the lateral position y refers to distance measured from the right side of the flume as the observer looks downstream. The longitudinal position x refers to distance measured from a station 60 ft* upstream of the transition section shown in Figure 1; e.g., the longitudinal position $x = 120$ coincides with sta 120 in the schematic.

19. A number of physical tests were conducted for several tilt angles and water depths in the flume (Ruff et al. 1987), but only one of those tests

---

* A table of factors for converting non-SI units of measurement to SI (metric) units is given on page 3.

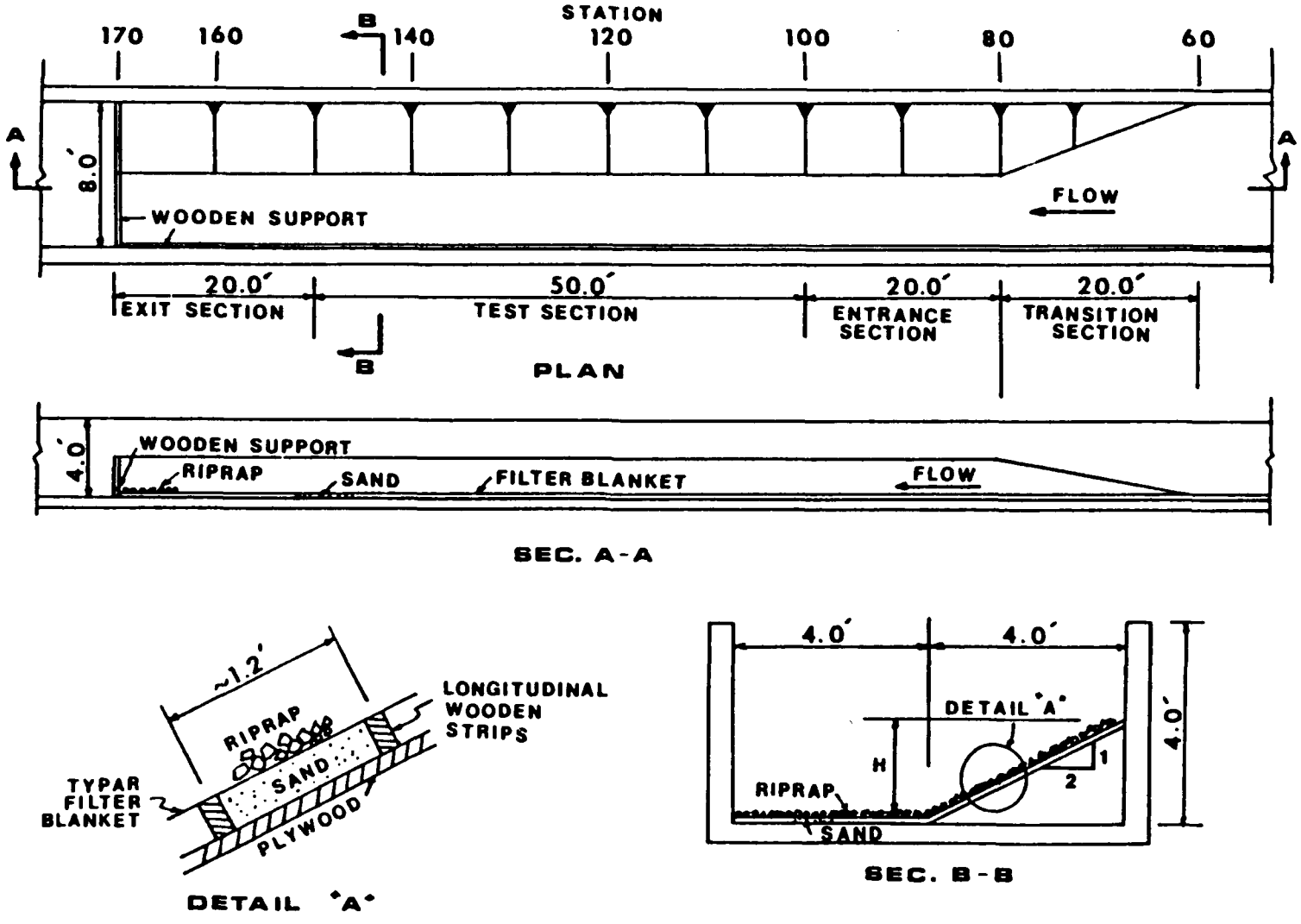


Figure 1. Schematic drawing for tilting flume (from Ruff et al. 1987)

will be used herein for comparison with computed results. Specifically, the test designated as number 20 by Ruff et al. (1987) was conducted with nearly identical bed and water-surface slopes of 0.00510 and 0.00515, respectively. The average rock diameter was 1 in., the riprap thickness was 2 in., and the flow rate was 40 cfs. The maximum water depths were 1.54, 1.56, and 1.48 ft at sta 120, 130, and 140, respectively. The estimated value for Manning's coefficient was 0.024.

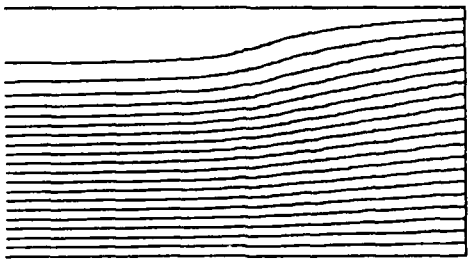
20. The STREMR computation simulating this test was executed on a 46 × 29 grid.* In the transition section of the flume (sta 60 to 80), the longitudinal grid spacing was 1.0 ft. This was gradually increased from 1.0 to 4.0 ft in the entrance section (sta 80 to sta 100), and was held fixed at 4.0 ft in the test and exit sections (sta 100 to sta 160). The lateral grid spacing was 0.25 ft at all stations. Figure 2 shows the grid and the steady-state velocity vectors and streamlines in the transition and entrance sections, computed with Manning's coefficient set at 0.03. Note that the lateral dimension on these plots has been magnified for the sake of clarity.

21. Since the purpose of this set of calculations was to examine the computed effects of bathymetry and bottom friction, the lateral boundaries were idealized as slip boundaries and the kinematic viscosity was taken to be 0.1 ft^2/sec. The distribution of inflow was held fixed (flux boundary), and the distribution of outflow was computed using the discrete radiation condition discussed for open boundaries in Appendix C. The maximum depth was set at 1.5 ft for all stations, and the banked portion of the bed was idealized by stairstepping the mean cell depth from cell to cell, using the topography option in STREMR (Appendix D). The Froude number based on the maximum depth was $Fr = 0.8$, and the Reynolds number based on the channel width was $Re = 400$. Note that for a maximum depth of 1.5 ft, the wetted width of the flume was 7.0 ft, and the water's edge lay at $y = 1.0$ ft.

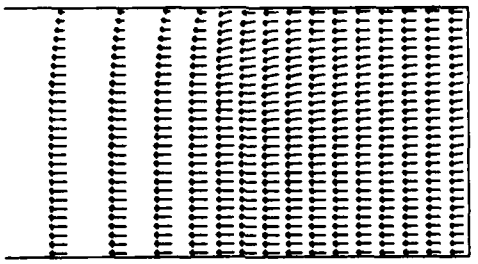
22. The STREMR code was executed with a time-step of 0.1 sec, and 200 time-steps were sufficient to establish a steady state in the test section. The computed depth-averaged velocities were nearly constant with x between sta 120 and 140, and these are plotted against lateral position y in Figure 3 for three different values of Manning's coefficient: $n = 0.02$,

---

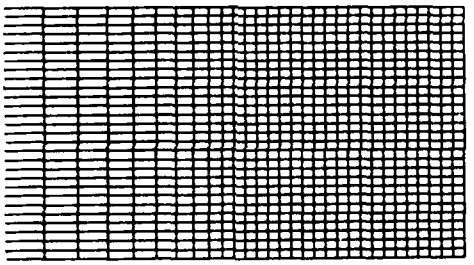
* A 46 × 29 grid has 46 nodes in the horizontal direction and 29 nodes in the vertical direction.



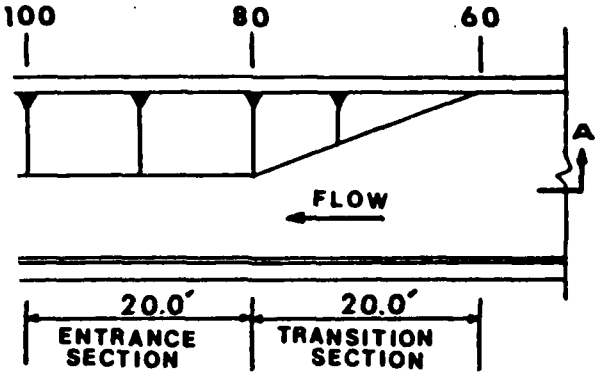
a. Computed streamlines



b. Velocity vectors



c. Grid



d. Entrance and transition sections

Figure 2. Computed streamlines, velocity vectors, and grid for transition and entrance sections of flume

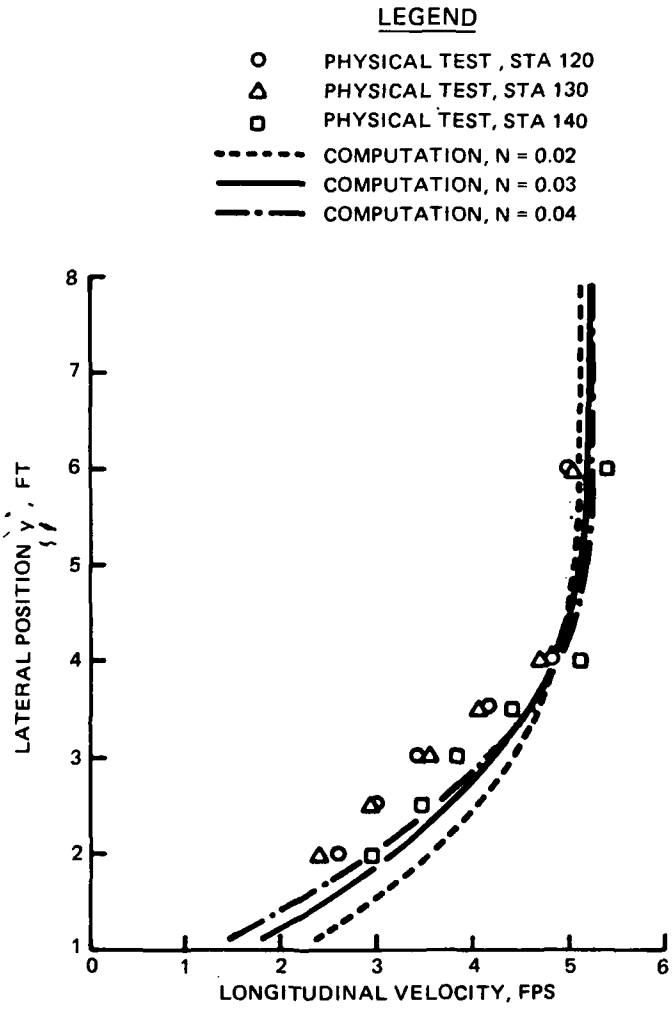


Figure 3. Comparison of computed and measured lateral distributions of depth-averaged longitudinal velocity for tilting flume

0.03 and 0.04. Measured depth-averaged velocities from sta 120, 130, and 140 are plotted for comparison. At first glance, agreement between computation and experiment appears quite good for the coefficients $n = 0.03$ and 0.04, especially for sta 140. Closer inspection, however, reveals that the measured velocity distribution curve has a slightly different shape from that of the computed distribution.

23. The STREMR code operates on the supposition that the flow is essentially two-dimensional; that is, that the velocity is uniform in the third dimension. While this seems a valid simplification in the flat portion of the channel cross section, it appears somewhat less so on the banked portion, especially near the water's edge ($y = 1$). This is borne out by comparison of computed (vertically uniform) velocities with measured vertical distributions (Figure 4). Here it is quite clear that some correction is needed for the nonuniform vertical distribution of longitudinal velocity near the water's edge.

24. The value estimated by Ruff et al. (1987) for Manning's coefficient ($n = 0.024$) was based upon the hydraulic radius of the flume. There was some question as to whether such a value is appropriate for the STREMR code, since the latter uses local depth instead of hydraulic radius to compute local resistance due to friction. In Figure 5, however, the plotted curve of STREMR-computed water-surface slope versus Manning's coefficient indicates that the correct slope (0.00515) is obtained for $n = 0.027$. This exceeds the estimated value of n by less than 13 percent.

Flow in a Bendway

25. Figure 6 shows cross-section and plan views for a channel consisting of a 15-ft straight section, a 100-deg circular bendway, and another 15-ft straight section. The channel has a uniform lateral width of 8.6 ft (water surface), with inner and outer radii of 17.7 and 26.3 ft, respectively, in the bendway. The cross section is trapezoidal with a maximum depth of 0.455 ft and a lateral width of 7.0 ft across the bottom. This configuration represents a partially filled, truncated segment of an existing physical model containing two bendways (Maynord 1988). The estimated value for Manning's coefficient is $n = 0.02$.

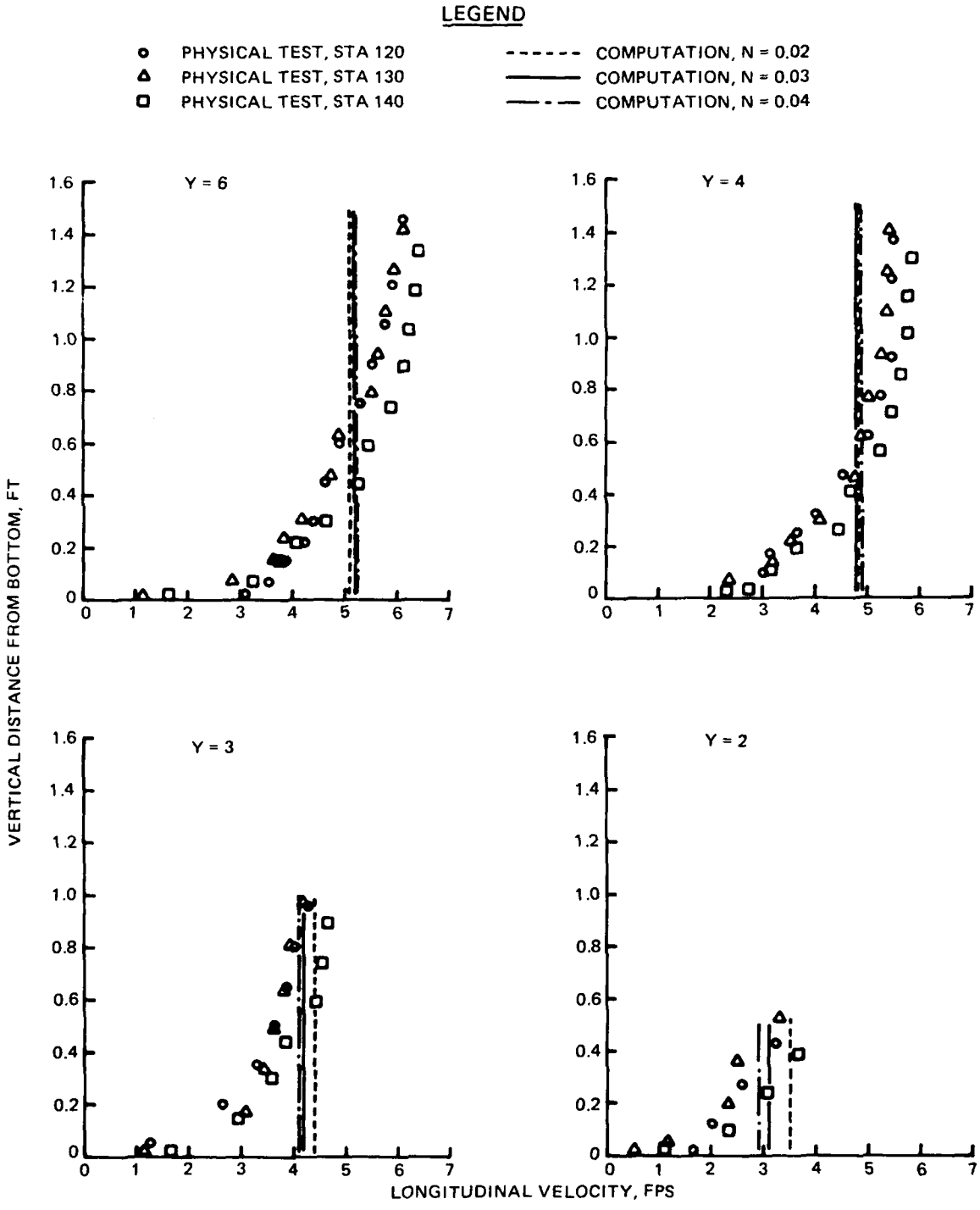


Figure 4. Comparison of computed and measured vertical distributions of longitudinal velocity for tilting flume

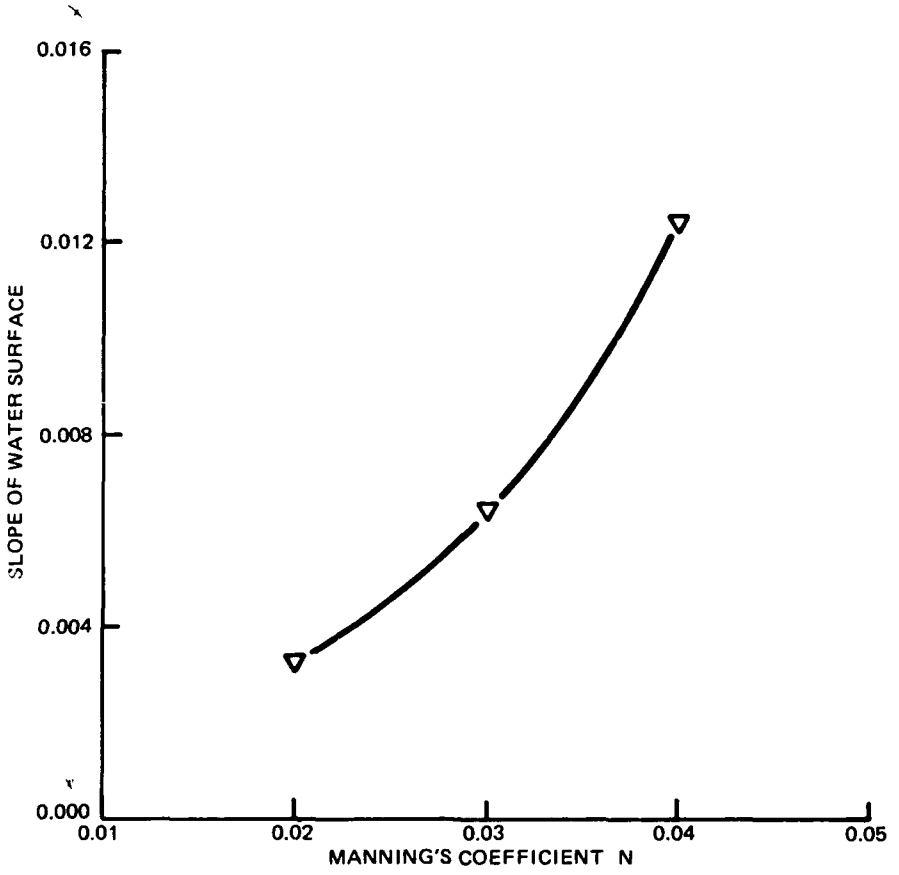
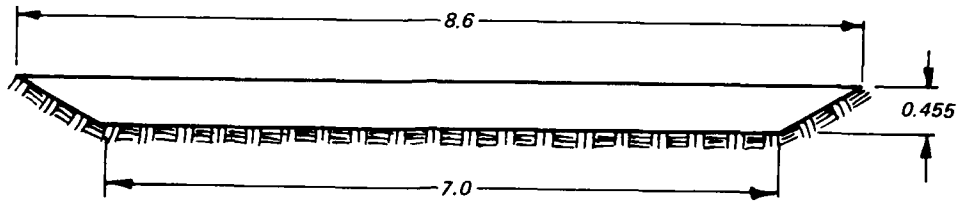
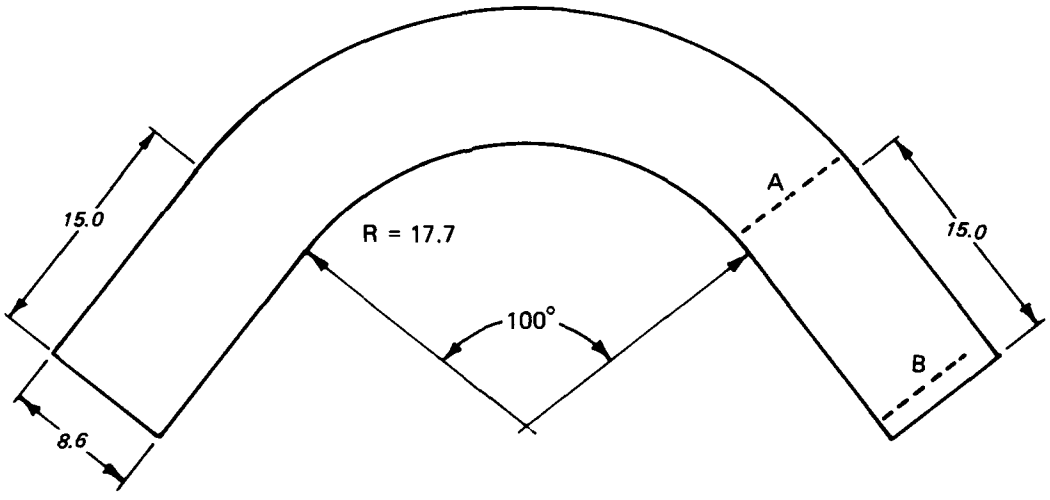


Figure 5. Computed slope of water surface versus Manning's coefficient



a. Channel cross section



b. Channel plan view

Figure 6. Schematic drawing for channel with bendway

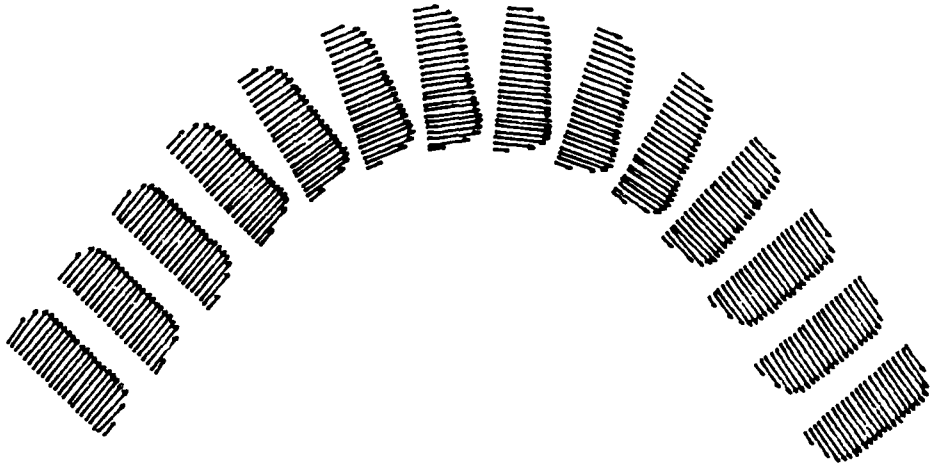
26. A 69 × 47 grid was used to simulate a flow rate of 6.75 cfs in the channel. Figure 7 shows the grid and the computed steady-state velocity vectors at various longitudinal stations. All of the 69 lateral (radial) grid lines appear in the figure, but only 24 of the 47 longitudinal grid lines are shown. The flow is from left to right (clockwise), with a fixed inflow distribution (directly proportional to depth) through the left (flux) boundary. The discrete radiation condition was used to compute the outflow distribution through the right (open) boundary. The no-slip condition was imposed on the inner and outer lateral boundaries, and the kinematic viscosity was taken to be 0.01 ft^2/sec. The Froude number based on the maximum depth was $Fr = 0.5$, and the Reynolds number based on the channel width was $Re = 1600$.

27. The STREMR code was executed with a time-step of 0.2 sec, and 200 time-steps were sufficient to establish steady-state flow at sta A and B (located as shown in Figure 7). Comparison of computed results with physical test results* is presented for sta A and B in Figures 8 and 9, respectively. Longitudinal velocity is plotted against lateral (radial) position measured from the inner boundary. As before, the computed velocities are depth-averaged values, but the test data represent measurements taken at vertical distances from the water surface of 20, 60, and 80 percent of the total depth.

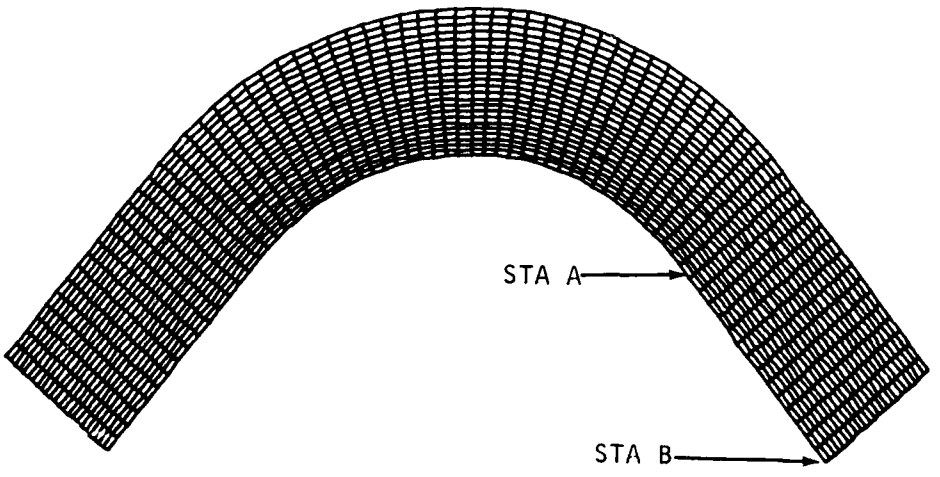
28. At first glance, agreement between the computed and measured velocity distributions looks only fair to poor, but upon closer inspection it looks even worse. The measured velocities for all three depths increase monotonically across the channel, with the increase becoming more dramatic from sta A to sta B. The computed velocities show just the opposite trend at sta A, with a slight improvement at sta B. These trends are only slightly affected by uniformly changing the viscosity or the Manning coefficient, or by varying the local depth with the local pressure (free-surface displacement). The poor agreement indicates that something is still missing from the mathematical model for the flow resistance. Some mechanism is needed that automatically varies the viscosity or Manning coefficient, or perhaps both, with position. In the context of the STREMR code, that is the only way that the computed trend can be dramatically reversed for the bendway.

---

* Unpublished test data provided by S. T. Maynord, Research Hydraulic Engineer, August 1987, US Army Engineer Waterways Experiment Station, Vicksburg, MS.



a. Velocity vectors



b. Grid

Figure 7. Computed velocity vectors and grid for channel with bendway

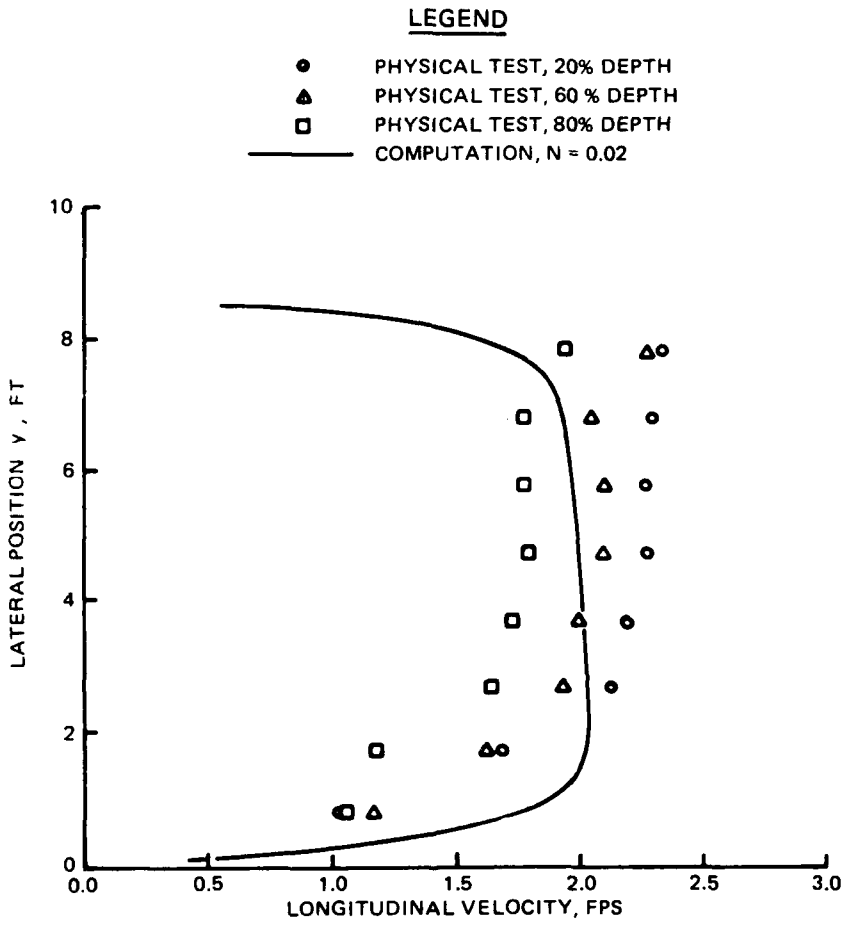


Figure 8. Comparison of computed and measured lateral distributions of longitudinal velocity for channel with bendway, sta A

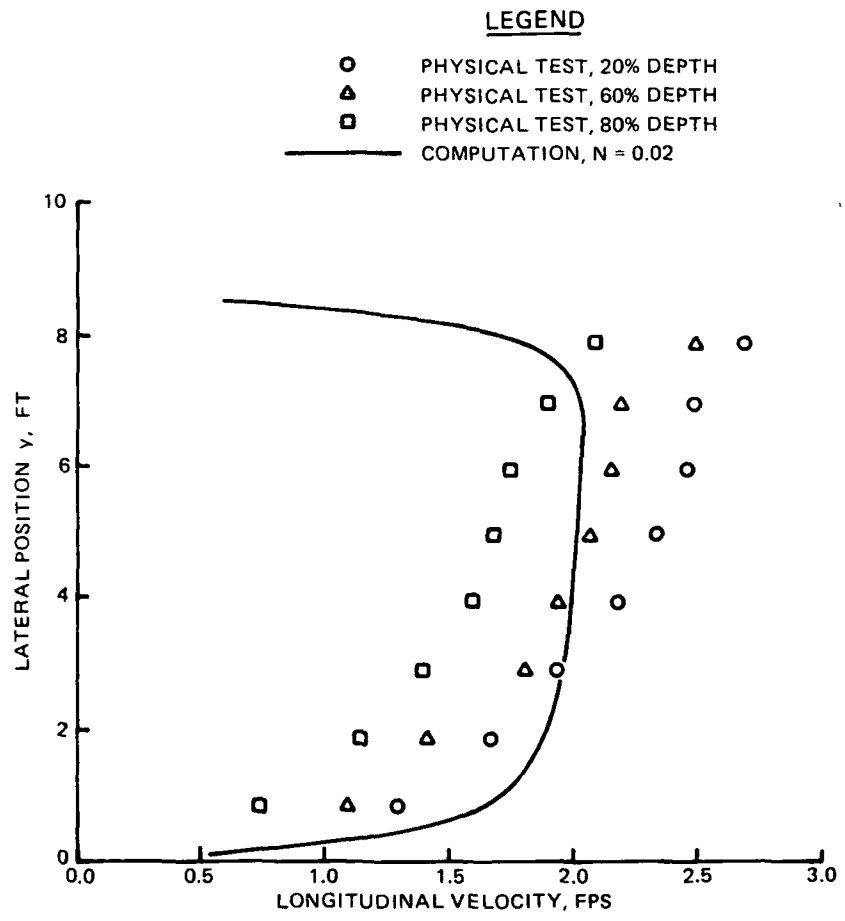


Figure 9. Comparison of computed and measured lateral distributions of longitudinal velocity for channel with bendway, sta B

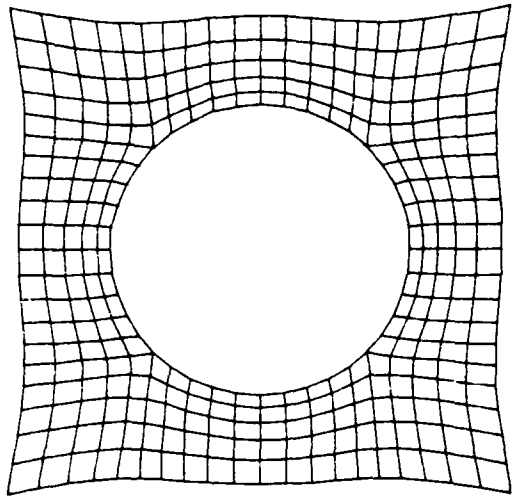
Flow Past a Circular Cylinder

29. Figure 10 shows a 91 x 41 grid that represents a channel containing a circular cylinder. The channel is 18 ft long, 8 ft wide, and 1 ft deep, and the cylinder is 2 ft in diameter. A STREMR calculation was performed for this configuration with a flow rate of 8.0 cfs and a kinematic viscosity of 0.02 ft^2/sec, making the Froude number $Fr = 0.176$ based on the depth, and the Reynolds number $Re = 100$ based on the diameter of the cylinder. Manning's coefficient was set to zero, so there was no resistance due to bottom friction.

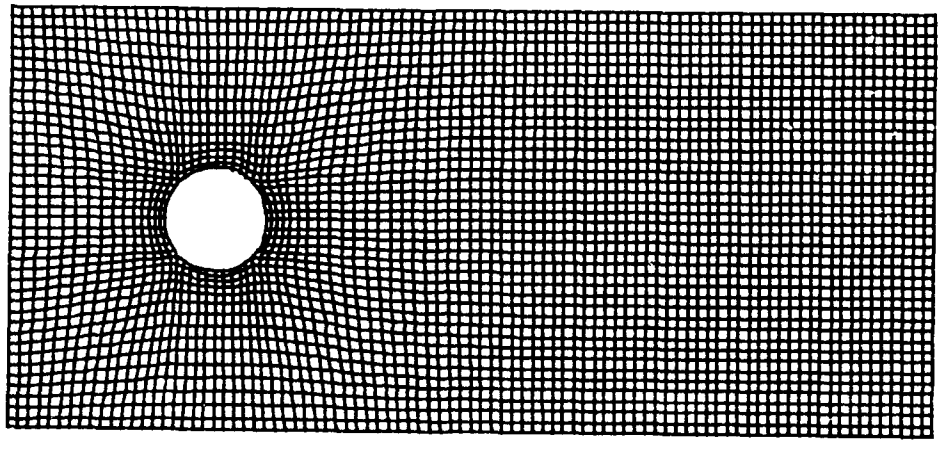
30. The inflow distribution through the left (flux) boundary was fixed and uniform for the duration of the run, and the outflow distribution through the right (open) boundary was allowed to change in accordance with the discrete radiation condition. The no-slip condition was imposed on the surface of the cylinder, the channel sides were taken to be slip boundaries, and the time-step was set at 0.05 sec. The code was allowed to run for 1,520 time-steps, and the resulting streamlines are shown in Figures 11 and 12.

31. Figure 11 shows the evolution of the flow field at intervals of 5 sec for the first 55 sec. The initial condition (at 0 sec) is potential flow, which gradually gives way to a long, symmetric wake (20 to 25 sec). The symmetric wake becomes unstable (30 to 40 sec) and degenerates to a periodic wake (45 to 55 sec) with a period of 9 to 10 sec. After 65 sec the periodic wake is well developed, as is shown by the streamline plots at intervals of 1 sec in Figure 12.

32. The flow pattern generated by the STREMR code is the well-known Karman vortex street, and the computed period (about 9.5 sec) is somewhat shorter than the observed period for unbounded flow, due to the proximity of the channel sides. The Strouhal number (frequency times diameter divided by velocity) resulting from the computation is about 0.21, while the measured value (Berger and Wille 1972) is about 0.17 for $Re = 100$. This set of calculations demonstrates that the code can accommodate transient flow at low Froude number and moderate Reynolds number. Moreover, the periodic wake evolves naturally in the presence of uniform steady inflow, without the need for any sort of externally added perturbation or forcing function.



a. Detail



b. Entire grid

Figure 10. Computational grid for channel with circular cylinder

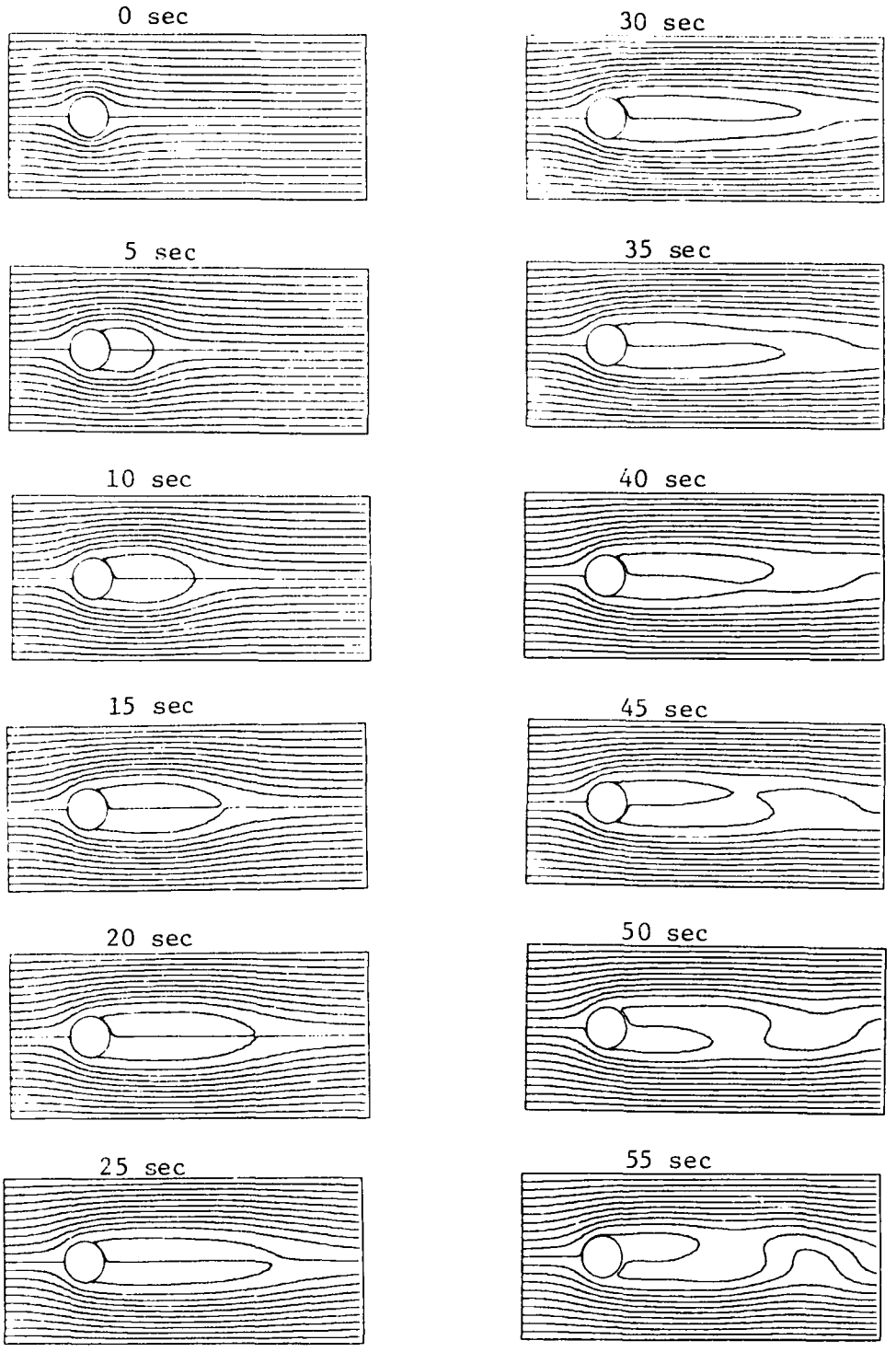


Figure 11. Computed streamlines for developing flow past circular cylinder

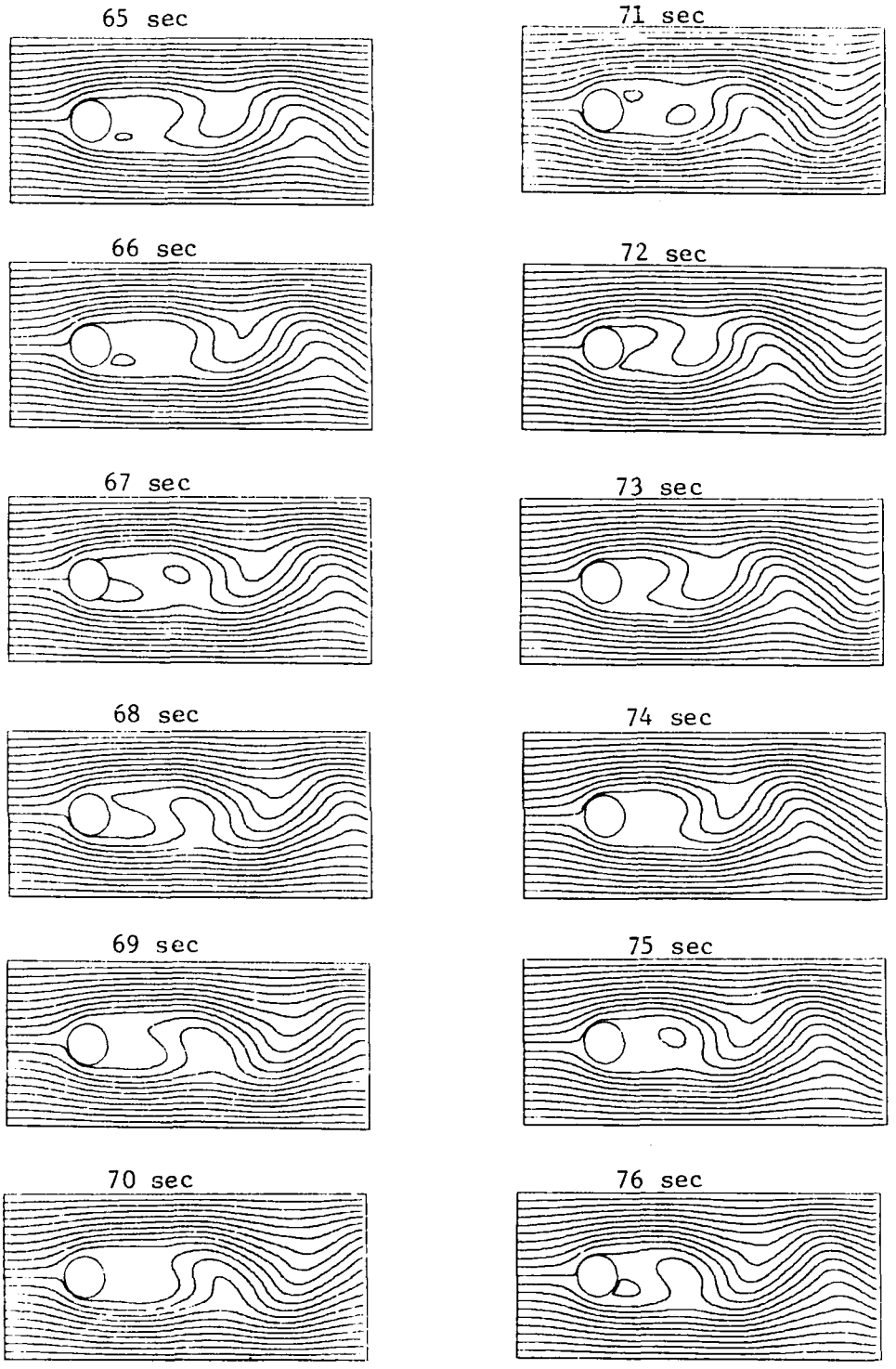


Figure 12. Computed streamlines for fully developed periodic flow past circular cylinder

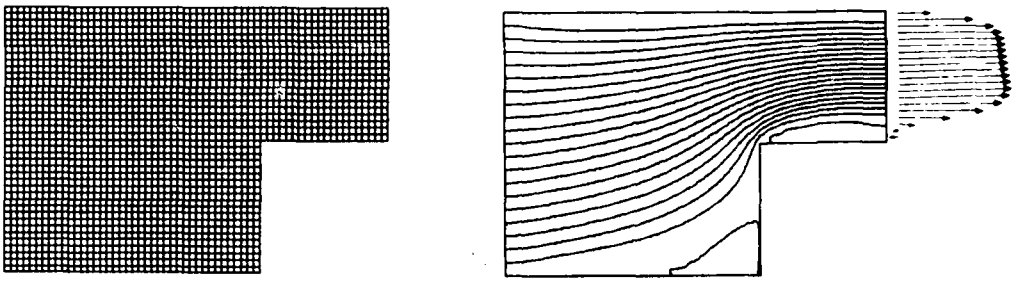
Flow Past an Abutment

33. Prior to physical modeling of a structure, the STREMR code may help reduce the number of alternatives for modification and repair of a given approach flow. As a simple hypothetical example, consider a channel 12 ft long, 8 ft wide, and 1 ft deep, with a square abutment 4 ft long and 4 ft wide (as shown in the topmost configuration of Figure 13). For this case the code was executed with a flow rate of 8.0 cfs, a kinematic viscosity of 0.02 ft^2/sec, and a Manning coefficient of 0.0 (no bottom friction). The Froude number was 0.176 to 0.352 based on the depth, and the Reynolds number was 400 based on the channel width. The 61 × 41 grid for the channel was bounded by a 21 × 21 abutment.

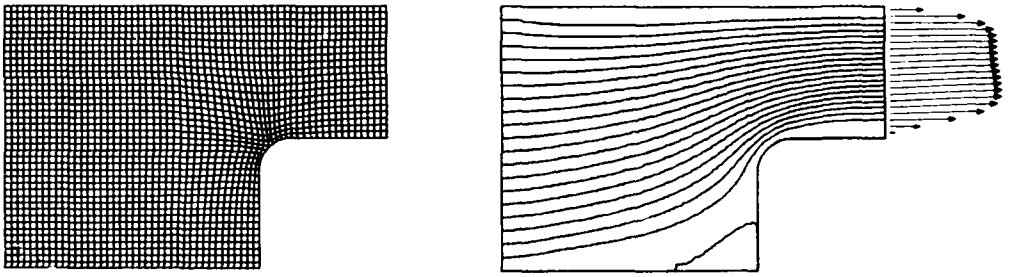
34. The inflow distribution was uniform and fixed at the left (flux) boundary, and the outflow distribution was allowed to vary according to the discrete radiation condition at the right (open) boundary. The no-slip condition was imposed on the channel sides and on the abutment. The time-step was set at 0.03 sec, and 600 steps were sufficient to establish a steady state near the outflow.

35. The STREMR code was run for four different abutment shapes, starting with the sharp-cornered configuration shown in Figure 13a. Thereafter the corner radius was increased to 1.0, 2.0, and 3.0 ft to investigate the effect of abutment shape upon outflow distribution. The resulting streamlines and velocity vectors are shown for the steady state in each case. (The velocities were taken 1 ft upstream of the outflow boundary.)

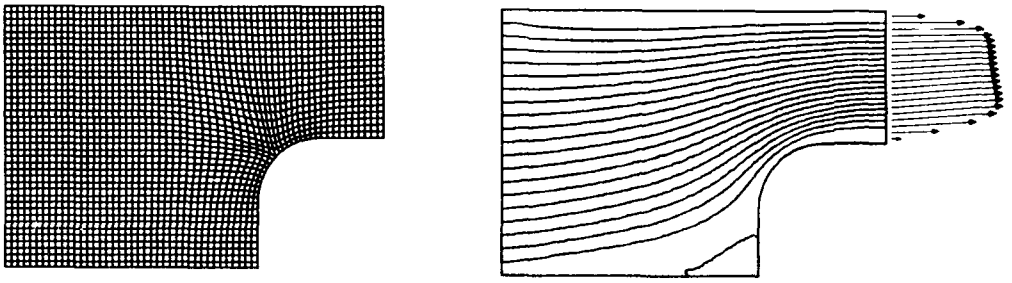
36. The sharp-cornered abutment produced backflow in the vicinity of the outflow boundary, and this would be considered an adverse flow if a symmetric distribution were the desired condition. Increasing the corner radius to 1.0 ft eliminated the backflow, but still left markedly different velocity distributions on opposite sides of the channel. Further increase of the corner radius improved the symmetry of the outflow.



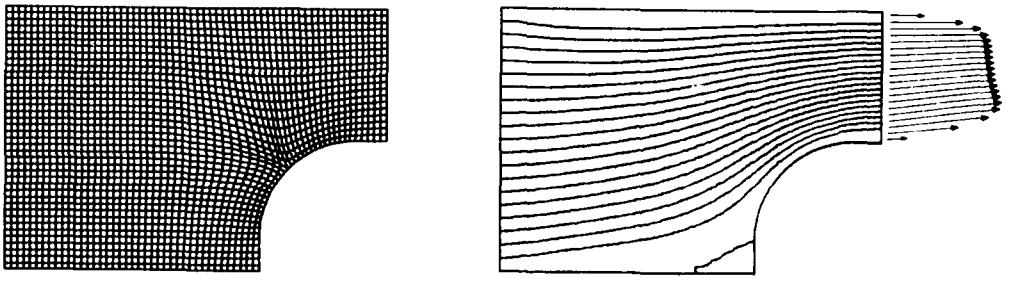
a. Sharp-cornered abutment



b. 1.0-ft corner radius



c. 2.0-ft corner radius



d. 3.0-ft corner radius

Figure 13. Computational grids, streamlines, and outflow velocity vectors for flow past an abutment

PART IV: DISCUSSION

37. The results presented in Part III warrant some measure of optimism, if not complacency. The preliminary version of the STREMR code (Appendix D) appears to satisfy the fundamental requirements of a 2-D approach flow model: it conserves mass and momentum, and it adequately resolves the development and transport of vorticity (circulation) provided the grid spacing is sufficiently fine. In general, at least 5 cells are needed to represent the smallest eddies in the flow field, so at least 10 cells should be used along each vorticity-producing boundary segment. This was the case for the flow about the circular cylinder, whose perimeter was divided into four arc lengths bounded by 10 cells each (Figure 10). Three or four times that many cells would have been necessary to resolve the fine details of the flow, such as separation points and secondary vortices, close to the cylinder itself. The grid used was fine enough to resolve only the large-scale periodic behavior in the wake, but that should be adequate for many approach flow calculations in which only the gross effects on some downstream hydraulic structure are needed.

38. The predictor-corrector scheme (Appendix B) is time-centered and has little or no damping, which might otherwise suppress the development of transient and periodic flow patterns. Damping reduces the amplitudes of transients in individual grid cells, and it is not to be confused with diffusion, which reduces the amplitude in one cell while increasing it in neighboring cells. While damping may be undesirable in transient flow calculations, some damping is helpful in steady-state calculations. Without damping, the amplitudes in a few cells (not necessarily in the region of interest) may oscillate back and forth forever, never reaching a steady state even though most of the flow field does converge. In practice the computation should be terminated when convergence is achieved in the region of interest. If convergence is not achieved when the flow is not truly periodic, then the grid should be refined and the placement of open and flux boundaries reexamined.

39. A word of caution is in order with regard to the placement of convex corners on the grid boundaries in an adverse (rising) pressure gradient. (See, for example, the detail for the grid about the circular cylinder in Figure 10.) Unless the grid has one set of grid lines well

aligned with streamlines, there may be enough false transverse diffusion (due to numerical error) to allow the flow to separate prematurely, or even incorrectly. Therefore, whenever the position of separation points is important, convex grid corners should be placed well downstream of anticipated separation-point locations. Otherwise flow separation may occur even for slip boundaries (which is not necessarily a problem if convex corners are coincident with desired points of separation). The problem of false transverse diffusion exists to some degree for all primitive-variable (pressure-velocity) formulations of the governing equations, since no discrete algorithms exist that are absolutely free of numerical error. The only way to avoid this kind of behavior completely is to reformulate the equations so that vorticity itself is one of the dependent variables. In most cases, however, it should be enough simply to recognize the possibility of grid-induced separation and to act accordingly in generating grids and setting boundary conditions.

40. The poor agreement between computation and experiment for the bendway indicates that an important mechanism is lacking in the mathematical model; i.e., in the 2-D equations of motion as presented in Appendix A. Perhaps a better formulation of the resistance due to bottom friction and vertical mixing is needed, acknowledging the vertical distribution of the horizontal velocity components, if not the vertical component itself. Likewise, a simple 2-D turbulence model might also yield better results by adjusting the local viscosity in proportion to lateral mixing. Whatever the solution, the deficiency appears to be with the equations of motion and not with the numerical algorithm. Since the existing code incorporates the simplest of friction formulas (Manning's equation), it would be somewhat naive to expect accurate flow predictions in general without further development. Even the apparent agreement between computation and experiment for the straight channel (tilting flume) is overshadowed by the subtle difference in shape of the lateral distributions of depth-averaged velocity, which may hint at the deficiency made quite plain by the results for the bendway.

41. As important as further model development is the need for prospective users to recognize the limitations imposed by the 2-D nature of the code. While most flow problems of interest entail some motion in three dimensions, many of them (especially those involving shallow water) can be reduced to quasi-2-D problems for engineering purposes. Nevertheless, the meaningful reduction of a problem from three dimensions to two is largely a

matter of judgment, and some experience in shallow-water hydraulics is beneficial.

42. With regard to computer resources, at least one million words of memory were needed for computing the flow about the cylinder, which used a 91 × 41 grid. In general the STREMR code takes 4 to 5 msec per grid cell per time-step on a VAX 11/750 minicomputer, but it runs at least 60 times faster on a CYBER 205. The complete flow history for the cylinder (1,520 time-steps) took about 6.5 hr on the VAX, but it would have taken only about 6.5 min on the CYBER. All of the calculations employed three to four iterations of the conjugate-gradient Poisson solver in each half of each time-step, and this was sufficient to keep errors below 1 percent for conservation of mass.

43. The flow calculation for the cylinder was relatively lengthy because the code had to execute many time-steps to follow the slow development of the flow in real (physical) time. By contrast, the calculation for the straight channel (200 time-steps to steady state on a 47 × 29 grid) took only 20 min on the VAX, and it would have taken only 20 sec on the CYBER. In general, steady-state calculations will be 10 to 100 times cheaper than transient/periodic calculations.

PART V: CONCLUSION

44. About 3 years' work has now gone into the development of a practical computer model for 2-D approach flow. The original version of the STREMR code was an outgrowth of the WESSEL code (Thompson and Bernard 1985) and used vorticity and stream function as the dependent variables instead of pressure and velocity. That version was strictly limited, however, to steady-state problems, and it has now been replaced by the current version, which is applicable for transient/periodic flow and incorporates recent developments in numerical methods.

45. The fundamental components of the STREMR code are now installed and working. The code conserves mass and momentum, and it resolves the development and transport of vorticity in real (physical) time on properly constructed computational grids with sufficiently fine grid spacing. The greatest remaining deficiency appears to be the lack of a turbulence model to account for the spatial variation of lateral mixing (diffusion) and vertical mixing (bottom friction). The remaining year or so of work will be devoted to finding and appending one or more turbulence models to the 2-D equations of motion. The ultimate goal will be to make reliable flow predictions within useful bounds, with as few judgment calls as possible on the part of the user.

46. Efforts will also continue toward making STREMR easier and less expensive to use. While these considerations are justifiably subordinate to reliability, they are nonetheless important if the code is to be of any help to potential users. It is anticipated that many prospective users will have access to mini- and microcomputers, and much of the software improvement will be oriented toward facilities of this kind.

47. A future report will include a complete user's manual for the final version of the STREMR approach-flow code. In the meantime, the existing code (Appendix D) will be made available to prospective users upon request.

REFERENCES

Berger, E., and Wille, R. 1972. "Periodic Flow Phenomena," Annual Review of Fluid Mechanics, Vol 4, p 313.

Bernard, R. S. 1986. "Discrete Solution of the Anelastic Equations for Mesoscale Modelling," Report GKSS 86/E/51, GKSS-Forschungszentrum, Postfach 1160, 2054 Geesthacht, West Germany.

Bernard, R. S. 1987. "Skew Grids and Irrotational Flow," Transactions of the Fifth Army Conference on Applied Mathematics and Computing, ARO Report 88-1, pp 631-641, US Army Research Office, Research Triangle Park, NC.

Kapitza, H., and Eppel, D. 1985. "A 3-D Poisson Solver Based on a Conjugate Gradient Algorithm," Report GKSS 85/E/23, GKSS-Forschungszentrum, Postfach 1160, 2054 Geesthacht, West Germany.

MacCormack, R. W. 1969. "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, American Institute of Aeronautics and Astronautics, Cincinnati, OH.

Maynord, S. T. 1988 (Mar). "Stable Riprap Size for Open Channel Flows," Technical Report HL-88-4, US Army Engineer Waterways Experiment Station, Vicksburg, MS; also Ph.D. Dissertation, Colorado State University, Fort Collins, CO.

Orlanski, I. 1976. "A Simple Boundary Condition for Unbounded Hyperbolic Flows," Journal of Computational Physics, Vol 21, pp 251-269.

Ruff, J. F., Shaikh, A., Abt, S. R., and Richardson, E. V. 1987 (Mar). "Riprap Stability in Side Sloped Channels," prepared under Contract No. DACW39-83-C-0045, by Colorado State University, Fort Collins, CO, for US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Thompson, J. F. 1983 (Mar). "A Boundary-Fitted Coordinate Code for General Two-Dimensional Regions with Obstacles and Boundary Intrusions," Technical Report E-83-8, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

Thompson, J. F., and Bernard, R. S. 1985 (Aug). "WESSEL: Code for Numerical Simulation of Two-Dimensional Time-Dependent Width-Averaged Flows with Arbitrary Boundaries," Technical Report E-85-8, US Army Engineer Waterways Experiment Station, Vicksburg, MS.

APPENDIX A: GOVERNING EQUATIONS

1. Consider an incompressible fluid in two dimensions (x,y) with density ρ and depth h normal to the plane of motion. Assuming that h is a gently varying function of position only, and that the kinematic viscosity ν is constant, then the depth-averaged equations of motion are

$$u_t + uu_x + vu_y = h^{-1}\nu \text{ div } (h \text{ grad } u) - \rho^{-1}p_x - X(u,v,h) \tag{A1}$$

$$v_t + uv_x + vv_y = h^{-1}\nu \text{ div } (h \text{ grad } v) - \rho^{-1}p_y - Y(u,v,h) \tag{A2}$$

$$(uh)_x + (vh)_y = 0 \tag{A3}$$

where

u,v = x- and y-components of velocity, respectively
t = time
div = divergence operator
grad = gradient operator
p = pressure
X,Y = x- and y-components of body force

and the subscripts x, y, and t denote partial derivatives. The body force in this case is the friction force, the components of which are functions of depth and velocity. Equations A1 and A2 are transport equations for the x- and y-components of momentum, and Equation A3 (the continuity equation) maintains conservation of mass. The pressure p represents the deviation from the local hydrostatic value that would exist in the absence of velocity gradients.

2. Cartesian coordinates are not well suited to flow configurations with nonrectangular boundaries, and it is expedient to rewrite the governing equations in general curvilinear coordinates

$$\xi = \xi(x,y) \tag{A4}$$

$$\eta = \eta(x,y) \tag{A5}$$

Under this coordinate transformation, Equations A1 through A3 become

$$hJu_t + Uu_\xi + Vu_\eta = \nu J \text{ div } (h \text{ grad } u) - \rho^{-1} hJp_x - hJX(u,v,h) \tag{A6}$$

$$hJv_t + Uv_\xi + Vv_\eta = \nu J \text{ div } (h \text{ grad } v) - \rho^{-1} hJp_y - hJY(u,v,h) \tag{A7}$$

$$U_\xi + V_\eta = 0 \tag{A8}$$

The quantities U and V are volumetric flux components, the subscripts ξ and η denote partial derivatives, and J is the Jacobian of the transformation, given by

$$U = y_\eta uh - x_\eta vh \tag{A9}$$

$$V = x_\xi vh - y_\xi uh \tag{A10}$$

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{A11}$$

The cartesian velocity components may also be expressed in terms of the flux components, using the relations,

$$u = h^{-1}J^{-1}(x_\xi U + x_\eta V) \tag{A12}$$

$$v = h^{-1}J^{-1}(y_\xi U + y_\eta V) \tag{A13}$$

The components of the gradient of any function f are

$$f_x = J^{-1}(y_\eta f_\xi - y_\xi f_\eta) \tag{A14}$$

$$f_y = J^{-1}(x_\xi f_\eta - x_\eta f_\xi) \tag{A15}$$

which may in turn be used in the expression for the divergence of the gradient,

$$J \text{ div } (h \text{ grad } f) = \frac{\partial}{\partial \xi}(hy_\eta f_x - hx_\eta f_y) + \frac{\partial}{\partial \eta}(hx_\xi f_y - hy_\xi f_x) \qquad (A16)$$

3. Although the spatial derivatives have been rewritten in terms of the curvilinear coordinates, the cartesian velocity components have been retained for convenience in the transport equations. The body-force components X and Y are unaltered by the change of coordinates because they do not involve spatial derivatives.

4. The body force due to friction is assumed to obey Manning's equation, with the depth h replacing the hydraulic radius:

$$X(u,v,h) = Kn^2 u(u^2 + v^2)^{1/2} h^{-4/3} \qquad (A17)$$

$$Y(u,v,h) = Kn^2 v(u^2 + v^2)^{1/2} h^{-4/3} \qquad (A18)$$

The quantity n is Manning's coefficient, and the factor of proportionality K is 1.4.5 for non-SI units (slugs, feet, seconds) and 9.81 for SI units (kilograms, metres, seconds).

APPENDIX B: PREDICTOR-CORRECTOR SCHEME

1. To represent incompressible approach flow with a computer (numerical) model, it is first necessary to discretize the governing equations (Appendix A, Equations A6 through A8) and then solve them with an appropriate algorithm. This is achieved in the manner discussed in the following paragraphs.

2. The flow field is divided into a finite number of discrete cells, and the boundaries may be permeable (inlets and outlets) or impermeable (solid walls). The arrangement of cells therein constitutes a finite-volume grid which may be curvilinear in the physical (x,y) plane (Figure B1), but rectangular in the computational (ξ,η) plane (Figure B2). The transformation of coordinates from (x,y) to (ξ,η) carries all information that pertains to grid spacing in the physical plane, and the choice of grid spacing in the computational plane has no effect on anything except convenience. Thus, a unit spacing is chosen in the ξ,η plane:

$$\Delta\xi = \Delta\eta = 1 \tag{B1}$$

This makes the grid cells perfectly square in the computational plane, but not necessarily so in the physical plane. Since the choice of planes for solving the governing equations is also arbitrary, the discrete formulation of the solution algorithm is carried out entirely in the ξ,η plane.

3. The cartesian velocity components (u, v) are defined at the center of each grid cell (Figure B2), where the depth h and pressure p are also computed. The volumetric flux component U is defined at the midpoint of the right (east) face, and the flux V at the midpoint of the upper (north) face. Each cell is identified with integer indices (i,j) so that the cell center occurs at position (ξ,η) given by

$$\xi = i - 1/2 \tag{B2}$$

$$\eta = j - 1/2 \tag{B3}$$

The values of x and y are specified at the cell corners, with $x(i,j)$ and $y(i,j)$ defined at the lower left (southwest) corner of cell (i,j).

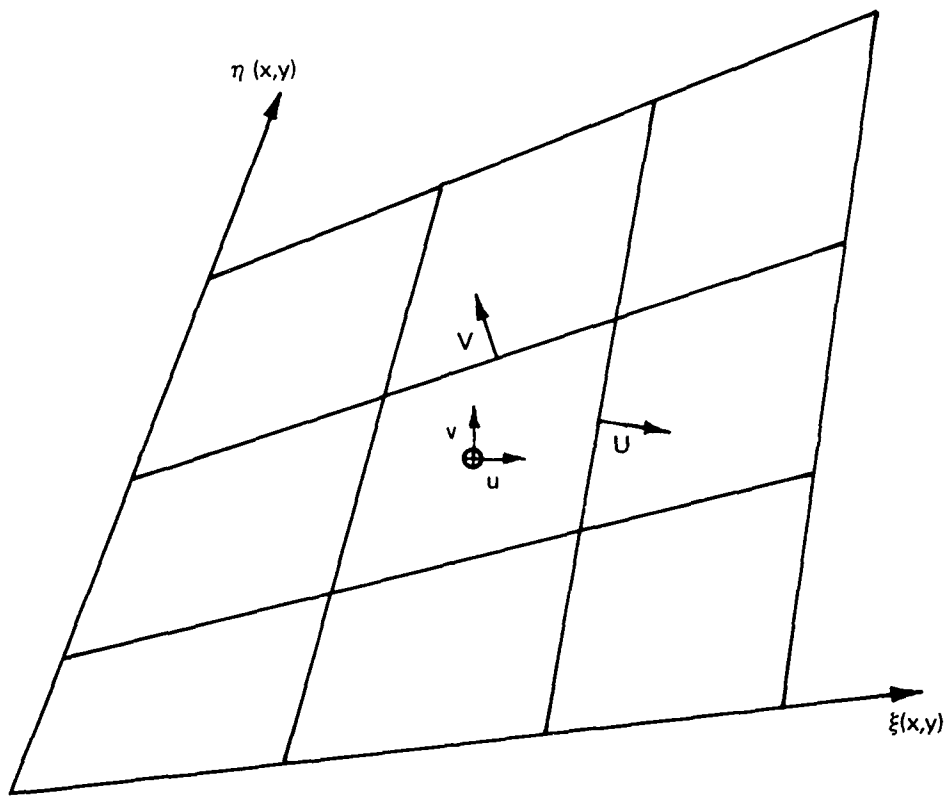


Figure B1. Computational grid in physical (x,y) plane

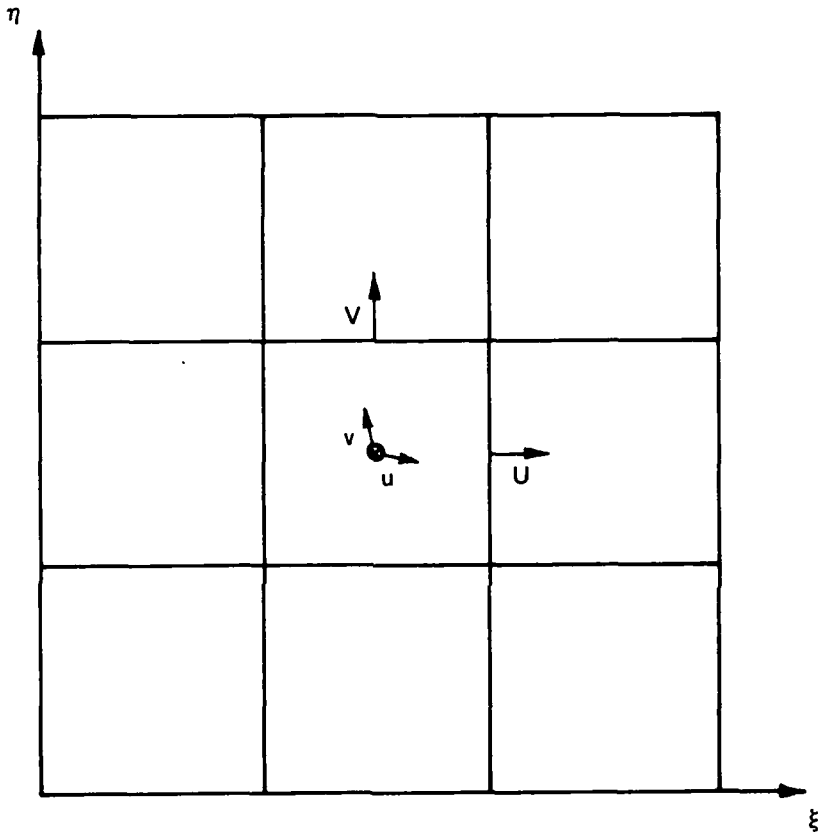


Figure B2. Computational grid in computational (ξ,η) plane

Letting subscripts and superscripts e,w,n,s denote the evaluation of quantities on east, west, north, and south faces of the cell, with c indicating the center, the derivatives of the x-coordinate are given by

$$x_\xi^n = x(i+1,j+1) - x(i,j+1) \tag{B4}$$

$$x_\eta^e = x(i+1,j+1) - x(i+1,j) \tag{B5}$$

$$x_\xi^e = \frac{1}{4}\,[x(i+2,j+1) - x(i,j+1) + x(i+2,j) - x(i,j)] \tag{B6}$$

$$x_\eta^n = \frac{1}{4}\,[x(i+1,j+2) - x(i+1,j) + x(i,j+2) - x(i,j)] \tag{B7}$$

$$x_\xi^c = \frac{1}{2}\,[x(i+1,j+1) - x(i,j+1) + x(i+1,j) - x(i,j)] \tag{B8}$$

$$x_\eta^c = \frac{1}{2}\,[x(i+1,j+1) - x(i+1,j) + x(i,j+1) - x(i,j)] \tag{B9}$$

and similar equations apply for the y-coordinate. Since (u,v) and (U,V) are specified at different locations on each cell, it is expedient to introduce the shift indices (r,s), which can be used to relate velocity and flux components on the staggered grid cell shown in Figure B2. The shift indices are pairs of integers with values of zero or unity, in any of four possible combinations; e.g.,

$$(r,s) = (1,0) \tag{B10}$$

The convention is adopted that <u>all</u> quantities have indices (i,j) <u>unless</u> specified otherwise. Note also that the fluxes through the east, west, north and south faces are U(i,j) , U(i-1,j) , V(i,j), and V(i,j-1), respectively. Combining the shift indices with Equations A12 and A13, the following relations now exist between the cell-centered velocity components and the face-centered flux components:

$$h_c J_c\, u(i,j) = x_\xi^c(i,j)\, U(i-r,j) + x_\eta^c(i,j)\, V(i,j-s) \tag{B11}$$

$$h_c J_c\, v(i,j) = y_\xi^c(i,j)\, U(i-r,j) + y_\eta^c(i,j)\, V(i,j-s) \tag{B12}$$

where J is the Jacobian of the curvilinear coordinate transformation. The r- and s-indices thus dictate whether U will be taken from the east or west face, and V from the north or south face, in computing the cell-centered velocity components $u(i,j)$ and $v(i,j)$.

4. In the algorithm at hand, only the face-centered flux components $U(i,j)$ and $V(i,j)$ will be kept from one time-step (or predictor-corrector phase) to the next. The cell-centered velocities $u(i,j)$ and $v(i,j)$ will always be computed from existing fluxes using the space-shifted relations given by Equations B11 and B12. These velocity components will be used to find cell-centered velocity increments (in time). The velocity increments will then be used to calculate flux increments on the cell faces by reversing the shift operation in Equations A9 and A10:

$$\Delta U(i-r,j) = h_c \left[y_\eta^c \Delta u(i,j) - x_\eta^c \Delta v(i,j) \right] \tag{B13}$$

$$\Delta V(i,j-s) = h_c \left[x_\xi^c \Delta v(i,j) - y_\xi^c \Delta u(i,j) \right] \tag{B14}$$

Before enumerating the details of the predictor-corrector scheme, it is helpful to introduce the difference operators

$$D_\xi f(i,j) = f(i,j) - f(i-1,j) \tag{B15}$$

$$D_\eta f(i,j) = f(i,j) - f(i,j-1) \tag{B16}$$

with the understanding that f can represent a single function or the product of two or more functions. Omitting the pressure and body-force terms from Equations A6 and A7, and employing the shift indices (r,s) <u>only</u> in the advective terms, each of the transport equations for momentum takes the same discrete form:

$$\frac{\Delta f}{\Delta \tau} + U(i-1+r,j) D_\xi f(i+r,j) + V(i,j-1+s) D_\eta f(i,j+s)$$

$$= J_c \nu_c \, \text{div} \, (h \, \text{grad} \, f) \tag{B17}$$

where

$$\Delta\tau = h_c^{-1} J_c^{-1} \Delta t \tag{B18}$$

and

ν = kinematic viscosity

div = divergence operator

grad = gradient operator

Δt = time increment

5. Using the difference operators given by Equations B15 and B16, the continuity equation (A8) takes the discrete form

$$D_\xi U(i,j) + D_\eta V(i,j) = 0 \tag{B19}$$

Given the shift indices (r,s), the procedure for advancing U and V by one time increment Δt will be the following:

a. Use Equations B11 and B12 to compute u and v from existing values of U and V.

b. Use Equation B17 to compute Δu and Δv, adding the body-force components as indicated by Equations A6 and A7.

c. Calculate the flux increments ΔU and ΔV from Equations B13 and B14.

d. Find new flux values U and V by adding the increments ΔU and ΔV to the existing values.

e. Adjust U and V for conservation of mass by adding a pressure gradient such that Equation B19 is satisfied.

In order to accomplish step e, it is convenient to introduce the scalar potential,

$$\phi = \frac{p\Delta t}{\rho} \tag{B20}$$

where ρ is density. Let U' and V' be the unadjusted flux components. The mass-conserving flux components are then found from the relations,

$$U = U' - h_e \left(y_\eta^e \phi_x^e - x_\eta^e \phi_y^e \right) \tag{B21}$$

$$V = V' - h_n \left(x_\xi^n \phi_y^n - y_\xi^n \phi_x^n \right) \tag{B22}$$

with the superscripts n and e indicating derivatives taken on the north and east faces, respectively. When the derivatives ϕ_x and ϕ_y are evaluated by the chain rule (Equations A14 and A15), the result is

$$\phi_x = J^{-1}(y_\eta \phi_\xi - y_\xi \phi_\eta) \tag{B23}$$

$$\phi_y = J^{-1}(x_\xi \phi_\eta - x_\eta \phi_\xi) \tag{B24}$$

For U and V to satisfy the continuity equation (B19), the scalar potential must satisfy the Poisson equation,

$$J \text{ div } (h \text{ grad } \phi) = U'_\xi + V'_\eta \tag{B25}$$

The evaluation of the left-hand side of Equation B25 is the same as that for the viscous terms in the transport equations, and it will be discussed later.

6. The five-step time-marching procedure is suitable for implementation in a two-phase predictor-corrector scheme like that of MacCormack (1969).* In the predictor phase, the five steps are followed as prescribed, using the existing values U^m and V^m from time-step m to calculate the provisional time-advanced values with the provisional increments ΔU^m and ΔV^m :

$$U^* = U^m + \Delta U^m - h(y_\eta \phi^*_x - x_\eta \phi^*_y) \tag{B26}$$

$$V^* = V^m + \Delta V^m - h(x_\xi \phi^*_y - y_\xi \phi^*_x) \tag{B27}$$

The superscript m indicates the time-step number, and the asterisk $*$ indicates the provisionally advanced time,

$$t^* = t^m + \Delta t \tag{B28}$$

Before starting the corrector phase, the shift indices (r,s) are replaced with the indices (r^*,s^*) given by

---

* References cited in this Appendix are included in the References at the end of the main text.

$$r^* = 1 - r \tag{B29}$$

$$s^* = 1 - s \tag{B30}$$

7. In the corrector phase, the first four steps of the time-marching procedure are executed exactly as in the predictor phase, using U^* and V^* to compute the continuity-violating increments ΔU^* and ΔV^*. These increments are used in step e of the corrector to calculate the mass-conserving flux components that will exist at time-step m+1 :

$$U^{m+1} = \frac{1}{2}(U^m + U^* + \Delta U^*) - h(y_\eta \phi^{**}_x - x_\eta \phi^{**}_y) \tag{B31}$$

$$V^{m+1} = \frac{1}{2}(V^m + V^* + \Delta V^*) - h(x_\xi \phi^{**}_y - y_\xi \phi^{**}_x) \tag{B32}$$

The average pressure during the time-step is

$$p^{m+1/2} = \frac{\rho}{2\Delta t}(\phi^* + 2\phi^{**}) \tag{B33}$$

The Poisson equation for ϕ^* in step e of the predictor phase is

$$J \text{ div } (h \text{ grad } \phi^*) = D_\xi(U^m + \Delta U^m) + D_\eta(V^m + \Delta V^m) \tag{B34}$$

and the Poisson equation for ϕ^{**} in step e of the corrector phase is

$$J \text{ div } (h \text{ grad } \phi^{**}) = \frac{1}{2}\left[D_\xi(U^m + \Delta U^m + \Delta U^*) + D_\eta(V^m + \Delta V^m + \Delta V^*)\right] \tag{B35}$$

8. The factor 2 appears in Equation B33 because $\phi^* = p^*\Delta t/\rho$ in the predictor phase, while $\phi^{**} = p^{**}\Delta t/2\rho$ in the corrector phase. Equation B35 removes not only those continuity violations generated in the first four steps of the corrector phase, but also any residual violations left over from the predictor phase.

9. As the predictor-corrector scheme marches the fluxes (U,V) through time, the shift indices (r,s) must be cycled so that each of four possible combinations is used with equal frequency. This is necessary to avoid the accumulation of error due to asymmetry (directional bias). As previously stated, the cell-centered velocities (u,v) need not be kept from one time-step to the next, nor even from predictor phase to corrector phase. They are

simply intermediate quantities needed to compute the velocity increments $(\Delta u, \Delta v)$, which are themselves intermediate quantities. In the evolution of the flow field, only the fluxes (U,V) and the pressure p are really important.

10. It now remains to clarify the discrete representation of the operation div (h grad f), which appears in the viscous terms of the transport equations (A6 and A7), and in the left-hand side of the Poisson equation (B25). If h is independent of position, the operation simply reduces to div grad f, which is the Laplacian of the function f. The cartesian form of the operation is quite simple, but the scheme outlined in this Appendix demands a general curvilinear form.

11. The fundamental approach taken herein is to evaluate the gradient operator (grad) via the chain rule (Equations B23 and B24), and the divergence operator (div) via the Gauss Divergence Theorem:

$$J \text{ div } (h \text{ grad } f) = E_\xi + F_\eta \tag{B36}$$

where

$$E = h(y_\eta f_x - x_\eta f_y) \tag{B37}$$

$$F = h(x_\xi f_y - y_\xi f_x) \tag{B38}$$

Once again using subscripts and superscripts (e,w,n,s) to indicate quantities evaluated on east, west, north, and south cell faces, it is helpful to write the discrete form of Equation B36 as

$$J \text{ div } (h \text{ grad } f) = E_e - E_w + F_n - F_s \tag{B39}$$

12. The fluxes E and F are proportional to normal derivatives of the function f on the cell faces. When used with the scalar potential ϕ, they represent corrections to the volumetric flux components (U,V); when used in the viscous terms, they represent components of momentum flux due to shear stress. Now introducing the double subscripts (cc,ee,ww,nn,ss,ne,nw,se,sw) to indicate coefficients and values of cell-centered quantities in adjacent cells (central, east, west, etc.) as shown in Figure B3, derivatives of the cell-

centered function f are approximated with the following finite-difference expressions:

$$f^e_\xi = f_{ee} - f_{cc} \tag{B40}$$

$$f^n_\eta = f_{nn} - f_{cc} \tag{B41}$$

$$f^e_\eta = \frac{1}{4}\,(f_{ne} - f_{se} + f_{nn} - f_{ss}) \tag{B42}$$

$$f^n_\xi = \frac{1}{4}\,(f_{ne} - f_{nw} + f_{ee} - f_{ww}) \tag{B43}$$

Incorporating Equations B40 through B43 into Equation B39, the latter takes the final form,

$$\begin{aligned} J\ \mathrm{div}\ (h\ \mathrm{grad}\ f) = {} & A_{ee}\, f_{ee} + A_{ww}\, f_{ww} + A_{nn}\, f_{nn} + A_{ss}\, f_{ss} \\ & + A_{cc}\, f_{cc} + A_{ne}\, f_{ne} + A_{nw}\, f_{nw} + A_{se}\, f_{se} + A_{sw}\, f_{sw} \end{aligned} \tag{B44}$$

and the coefficients are given by

$$A_{cc} = -\alpha_e - \alpha_w - \beta_n - \beta_s \tag{B45}$$

$$A_{ee} = \alpha_e - \gamma_n + \gamma_s \tag{B46}$$

$$A_{ww} = \alpha_w + \gamma_n - \gamma_s \tag{B47}$$

$$A_{nn} = \beta_n - \gamma_e + \gamma_w \tag{B48}$$

$$A_{ss} = \beta_s + \gamma_e - \gamma_w \tag{B49}$$

$$A_{ne} = -\gamma_e - \gamma_n \tag{B50}$$

$$A_{nw} = \gamma_w + \gamma_n \tag{B51}$$

$$A_{se} = \gamma_e + \gamma_s \tag{B52}$$

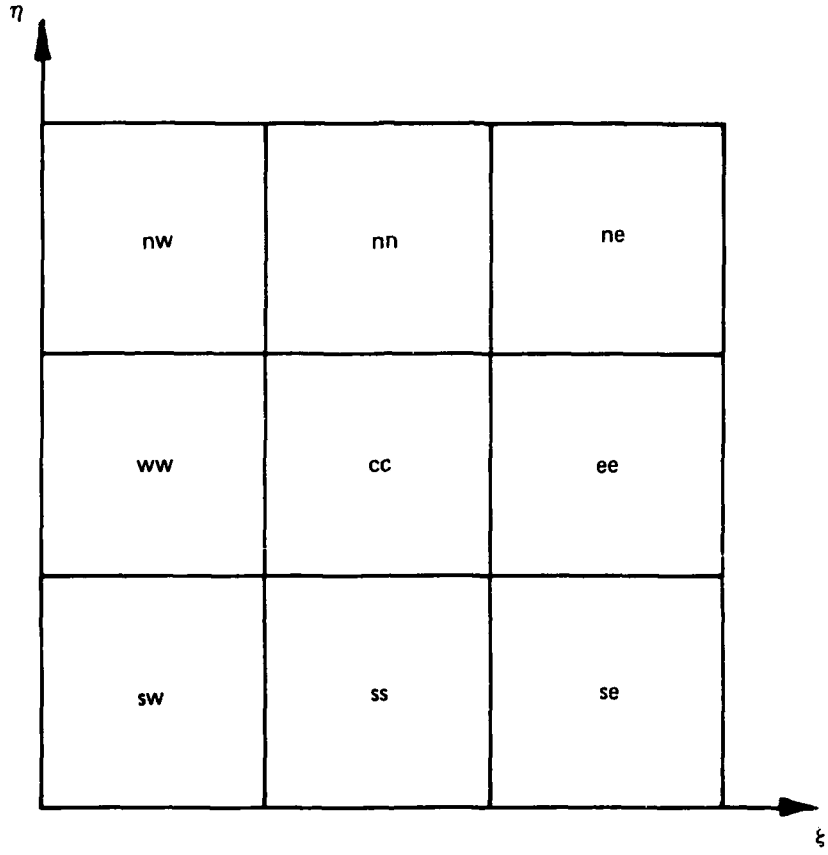


Figure B3. Labeling system for cell-centered quantities in neighboring grid cells

$$A_{sw} = -\gamma_w - \gamma_s \tag{B53}$$

where

$$\alpha = hJ^{-1}\left(x_\eta^2 + y_\eta^2\right) \tag{B54}$$

$$\beta = hJ^{-1}\left(x_\xi^2 + y_\xi^2\right) \tag{B55}$$

$$\gamma = \frac{1}{4} hJ^{-1}\left(x_\xi x_\eta + y_\xi y_\eta\right) \tag{B56}$$

13. The expressions for the coefficients (Equations B45 through B53) are valid for cells not adjacent to boundaries. The incorporation of boundary conditions is discussed at length in Appendix C. At this point it is sufficient merely to recognize that the boundary constraints can be absorbed directly into the coefficients of the function f for every cell in the flow field, including those adjacent to the boundaries.

14. Subject to the discretization procedure outlined in the previous paragraphs, the Poisson equation reduces to a set of N equations in N unknowns, where N is the total number of cells in the computational flow field. The N equations are linear and the matrix of coefficients is sparse, which means that each row or column has only a few nonzero entries (a maximum of nine). Large sparse systems of linear equations are best solved by iterative methods, and the specific method chosen may depend somewhat on the flow problem to be solved. A preliminary version of the STREMR code (Appendix D) offers the user a choice of one of two successive over-relaxation (SOR) methods or an idealized generalized conjugate-gradient (IGCG) method.

APPENDIX C: BOUNDARY CONDITIONS

1. The treatment of boundary conditions is just as important as the treatment of the governing equations. The latter serve to propagate the flow variables through space and time, but only in the manner permitted by the boundaries. Poorly formulated boundary conditions will always generate poor results, no matter how accurate the computational method. For discussion, boundary conditions may be divided into two categories:

a. Definite boundary conditions arise from some known (or assumed) physical constraint.

b. Indefinite boundary conditions arise only because the computational flow field is smaller than the physical flow field.

Boundary conditions for pressure generally fall into the first category, but those for velocity and volumetric flux may fall into either.

2. Wherever the normal component of volumetric flux (either U or V) can be specified on the boundary at each instant, there exists a simple Neumann condition for the scalar potential ϕ defined in Appendix B. That is, if either the east or west face of a cell is coincident with a boundary, and the flux U is known for that segment, then $U' = U$ and there is no correction needed for that flux component. This reduces Equation B21 to

$$h(y_\eta \phi_x - x_\eta \phi_y) = 0 \tag{C1}$$

where h is the depth. Likewise, if either the north or south face coincides with a boundary segment where V is known, then $V' = V$ and Equation B22 reduces to

$$h(x_\xi \phi_y - y_\xi \phi_x) = 0 \tag{C2}$$

Equations C1 and C2 apply on solid boundaries (zero flux) and on inlets and outlets (nonzero flux). In the treatment used herein, these constraints are absorbed directly into the Poisson equation for boundary-adjacent cells. For example, if the east cell face lies on a boundary, then only the north, south and west faces have nonzero normal derivatives of ϕ, and Equation B39 becomes

$$J \text{ div } (h \text{ grad } \phi) = -E_w + F_n - F_s \tag{C3}$$

where

J = Jacobian of the curvlinear coordinate transformation

div = divergence operator

grad = gradient operator

E,F = fluxes proportional to normal derivatives of the function f on the cell faces

e,w,n,s,c = subscripts denoting evaluation of quantities on east, west, north, and south faces of grid cell with c indicating the center

Likewise, if both the east and north faces lie on boundaries (i.e., the cell lies in a concave corner), then Equation B39 reduces to

$$J \text{ div } (h \text{ grad } \phi) = -E_w - F_s \tag{C4}$$

For orthogonal grids, no further considerations are necessary in computing the scalar potential (and hence the pressure) and its derivatives on boundary-adjacent cells. In general, however, curvilinear grids are not orthogonal, and additional logic is needed for cell faces with only one end touching a boundary.

3. Suppose that the east cell face coincides with a boundary, but the north face does not. In this case the northeast corner of the cell lies on the boundary, and the northwest corner lies in the field. In other words, the north face has only one end touching a boundary. Using Equation B41, it is straightforward to compute the η-derivative of ϕ on the north face:

$$\phi^n_\eta = \phi_{nn} - \phi_{cc} \tag{C5}$$

On the contrary, a problem arises if Equation B43 is used to calculate the ξ-derivative of ϕ on the north face:

$$\phi^n_\xi = \frac{1}{4} (\phi_{ne} - \phi_{nw} + \phi_{ee} - \phi_{ww}) \tag{C6}$$

Equation C6 requires information outside the flow field, and it is incomplete until the outside pressures are specified. Previous experience (Bernard 1987)* has shown that, rather than using Equation C6, it is better to use the first-order approximation:

$$\phi_{\xi}^{n} = \frac{1}{2} (\phi_{nn} - \phi_{nw} + \phi_{cc} - \phi_{ww}) \tag{C7}$$

4. Similar reasoning applies for all boundary-adjacent cells, and the equations for the coefficients of ϕ in the discrete Poisson equation (analogous to Equations B45 through B53) must accommodate every possibility directly, including concave and convex corners. The preliminary version of the STREMR code takes care of this automatically.

5. This same logic is used for the evaluation of viscous terms in cells adjacent to slip boundaries, inlets, and outlets; that is, the normal component of momentum flux due to shear stress is assumed to be zero on these boundaries.

6. For cells adjacent to no-slip boundaries, the normal component of momentum flux through the boundary is computed subject to the constraint that both velocity components are zero <u>on</u> the boundary. Suppose, for example, that the lower (south) cell face coincides with a no-slip boundary. If the cell indices are (i,j), then the cell-centered velocity components (u,v) in cell (i,j-1) are equal but opposite to those in cell (i,j):

$$u(i,j-1) = -u(i,j) \tag{C8}$$

$$v(i,j-1) = -v(i,j) \tag{C9}$$

Actually, cell (i,j-1) is a fictitious cell that lies across the boundary from cell (i,j), but the imposition of Equations C8 and C9 forces the velocity to be zero precisely on the boundary. Since the velocity is uniformly zero for all cell faces tangent to the boundary, the ξ-derivatives of u and v are zero on the south face, and the shear-stress momentum flux through the south face takes a simple form. For example, the momentum flux νF_s, obtained by

---

* References cited in this Appendix are included in the References at the end of the main text.

replacing ϕ by u in Equation C3, reduces to

$$\nu F_s = 2\nu\beta_s u(i,j) \tag{C10}$$

where ν is kinematic viscosity and β is defined by Equation B55. Similar arguments and equations hold for u and v in other cells adjacent to no-slip boundaries.

7. It may sometimes happen that the computational grid is too coarse to resolve the viscous forces arising from the no-slip condition; that is, the amount of viscosity required for numerical stability may be greater than the actual fluid viscosity, making the force imposed by Equation C10 unrealistically large. When this happens, it may be advisable to use a smaller value of ν in Equation C10 than in the other viscous terms. Accordingly, the preliminary version of STREMR allows the user to specify one coefficient of viscosity to be used only for imposing the no-slip condition, and another to be used for the remaining viscous terms.

8. All of these boundary conditions fall into the category of definite boundary conditions, because they impose known or assumed constraints on the pressure and velocity. It still remains, however, to set conditions for the components of velocity (u,v) and volumetric flux (U,V) at inlets and outlets, and these may be either definite or indefinite.

9. Wherever the volumetric flux is a specified function of time, normal to an inlet or outlet, then the boundary condition for U or V is definite. On the other hand, if this flux cannot be specified in advance, the boundary condition is indefinite. The latter situation arises when the computational grid does not cover the entire physical flow field (e.g., a truncated river or reservoir), so that some inlets or outlets may simply represent pseudo-boundaries along which the field is truncated to reduce computer storage and time requirements. On these boundaries it is expedient to use a discrete radiation condition (Orlanski 1976), which allows outward-moving disturbances to leave the computational region instead of bouncing around like echoes.

10. The preliminary version of the STREMR code makes a distinction between boundaries and pseudo-boundaries with respect to the normal component of volumetric flux. If the flux is fixed by the user for the duration of the computation, the cell adjacent thereto is designated a FLUX cell; but if the

flux is calculated from a discrete radiation condition, this cell is designated an OPEN cell. For the predictor-corrector scheme used herein, the radiation condition has been adapted in the following manner.

11. If the east or west face of a cell lies on an OPEN (radiation) boundary, then the volumetric flux normal to that face is assumed to obey the simple advection equation,

$$U_t + \hat{c}U_\xi = 0 \tag{C11}$$

where t denotes time and $\hat{c}$ is the discrete phase velocity. Likewise, if the north or south face lies on an OPEN boundary, then the flux normal to that face obeys a similar equation,

$$V_t + \hat{c}V_\eta = 0 \tag{C12}$$

Suppose that the east face of cell (i,j) coincides with an open boundary, so that $U(i,j)$ obeys Equation C11. The discrete phase velocity is taken to be

$$\hat{c} = -\left(\frac{U_t}{U_\xi}\right)^{m-1/2}_{i-3/2} \tag{C13}$$

where the superscript $m-1/2$ denotes quantities evaluated at the intermediate time between time-steps m and $m-1$, while the subscript $i-3/2$ indicates quantities taken halfway between the east and west faces of cell $(i-1,j)$. In other words, the phase velocity is to be computed from information at the previous time-step (or predictor phase) in the only neighboring field cell that shares a common face with the boundary-adjacent cell (i,j). Using a central-difference approximation in both space and time, it follows that

$$\frac{\hat{c}\Delta t}{\Delta\xi} = \frac{U^m(i-1,j) - U^{m-1}(i-1,j) + U^m(i-2,j) - U^{m-1}(i-2,j)}{U^m(i-2,j) + U^{m-1}(i-2,j) - U^m(i-1,j) - U^{m-1}(i-1,j)} \tag{C14}$$

The left-hand side of Equation C14 is the radiation Courant number C, related to the discrete phase velocity by

$$C = \frac{\hat{c}\Delta t}{\Delta\xi} \tag{C15}$$

Since the object here is for the east face of cell (i,j) to transmit information outward (to the right) but not inward, the value of C is set to zero if the right-hand side (RHS) of Equation C14 is negative; i.e.,

$$C = \text{MAX}[0, \text{RHS(C14)}] \tag{C16}$$

Furthermore, to avoid possible instability from Courant numbers greater than unity, Equation C16 is supplemented with a second constraint,

$$C = \text{MIN}[1, \text{RHS(C16)}] \tag{C17}$$

This keeps the flux Courant number in the range $0 \leq C \leq 1$, and the value for U(i,j) at the end of the predictor phase of time-step m now becomes

$$U^*(i,j) = (1 - C)\, U^m(i,j) + CU^m(i-1,j) \tag{C18}$$

where * denotes provisionally advanced time. Similar arguments apply for boundary-normal flux components on other cells. If the boundary coincides with a west or south face, however, the flux Courant number is constrained to lie in the range $-1 \leq C \leq 0$ to satisfy the requirement that changes be transmitted outward from the flow field.

12. The discrete radiation condition is implemented in each predictor and corrector phase, using radiation Courant numbers calculated at the end of the preceding predictor phase or time-step. Equation C14 is used to obtain C for the predictor phase of time-step m. In the corrector phase, however, the superscripts m and m-1 are replaced by the superscripts * and m, respectively, in Equation C14. This value of C is then used to compute the boundary value for U(i,j) at time-step m+1 :

$$U^{m+1}(i,j) = \frac{1}{2}[U^m(i,j) + (1 - C)U^*(i,j) + CU^*(i-1,j)] \tag{C19}$$

13. The transport equations for momentum require the specification of velocity components outside the flow field for cells adjacent to FLUX and OPEN boundaries. In the preliminary version of STREMR, this is done by two-point extrapolation. For example, if the east face of cell (i,j) lies on an OPEN or

FLUX boundary, the values of (u,v) outside the flow field are taken to be

$$u(i+1,j) = 2u(i,j) - u(i-1,j) \qquad \text{(C20)}$$

$$v(i+1,j) = 2v(i,j) - v(i-1,j) \qquad \text{(C21)}$$

Similar equations apply for cells adjacent to other FLUX and OPEN boundaries.

APPENDIX D: USER'S MANUAL FOR PRELIMINARY STREMR CODE

Background

1. The VAXPVC code is a preliminary version of the two-dimensional STREMR code using primitive variables (pressure and velocity). It uses an explicit predictor-corrector scheme (Bernard 1986)* to solve the equations of motion for incompressible flow on boundary-fitted grids constructed by the WESCOR code (Thompson 1983). From the standpoint of the user, however, the flow field can be thought of as a rectangular domain in the computational plane, containing indentions (such as dikes) and obstacles (such as islands). Thus, the boundaries are all straight lines in the computational (i,j) plane, but they may be curved in the physical (x,y) plane.

2. In the computational plane, all quantities except the components of the volumetric-flux vector are defined at the centers of the grid cells; that is, the pressure p and the depth h are computed at the cell centers, along with the cartesian velocity components u and v. The volumetric-flux components U and V are computed at the midpoints of the cell faces, but this is done automatically by the code, along with the calculation of pressure. The user may specify only u, v, and h in advance for the entire field or for selected locations.

3. The grid is best thought of as a collection of finite-volume cells, in which the cell-centered quantities actually represent volume-averaged quantities for each cell. The cell faces are then local boundaries through which influence is transmitted from one cell to another. In other words, the only way that two cells can affect each other is for information to cross their common cell face.

4. In its present form, VAXPVC can calculate time-varying flow fields subject to advection, diffusion, and friction. Advection and diffusion are handled automatically by the predictor-corrector scheme, but the user can specify the diffusion coefficient (viscosity) and the friction coefficient (Manning's n). Friction may vary from cell to cell, but the viscosity is uniform for the entire field. Only on no-slip boundaries may a distinct (but

---

* References cited in this Appendix are included in the References at the end of the main text.

uniform) viscosity coefficient be used for imposing resistance due to the no-slip condition.

5. Each grid cell is labeled by a pair of indices (I,J), with I running left to right from 0 to IMAX+1 , and J running bottom to top from 0 to JMAX+1 . Furthermore, each cell has a user-given type designation into one of six categories:

FIELD	Any cell lying in the computed flow field, but not adjacent to a boundary
OUT	Any cell not lying in the computed flow field
SLIP	Any cell adjacent to a slip boundary
NOSLIP	Any cell adjacent to a no-slip boundary
FLUX	Any cell adjacent to a boundary segment with a fixed normal component of mass flux
OPEN	Any cell adjacent to a boundary segment with a variable normal component of mass flux

6. The code reads the grid data from the WESCOR output, and automatically designates all cells as either FIELD, OUT, or SLIP. Cells lying inside obstacles are OUT, and other cells not adjacent to boundaries are FIELD. The remaining boundary-adjacent cells are SLIP by default, unless altered by subsequent user input. The user must designate all cells that are to be NOSLIP, FLUX, or OPEN.

7. After all cell types have been assigned, the code reads user input for general (global) values for flow rate, bottom-friction coefficient (Manning's n), kinematic viscosity, x-velocity, y-velocity, and depth. These will be the initial values used for all cells unless altered by subsequent user input.

8. Next the code reads user input for section values on specified groups of cells for x-velocity, y-velocity, bottom friction, and depth. Finally the code reads input for lines of cells along which these same quantities will be interpolated (by arc length) from specified values at the end points.

9. At this point, if the user so designates, the code will recompute the depth as the solution to a Laplace equation, for which the section-input and line-input values will be held fixed. General input values will be used for boundary-adjacent cells, unless superseded by section or line input. This option facilitates the generation of smoothly varying depth (bottom topography) with sparse input data.

10. With all velocities and depths now assigned, flux components on all inflow and outflow boundary segments are adjusted to comply with one of three user-selected options:

a. The net inflow and outflow must match a specified flow rate.

b. The net outflow must match the net inflow computed from the user-specified velocities.

c. The net inflow must match the net outflow computed from the user-specified velocities.

With this step complete, mass is conserved insofar as total flux through the boundaries is concerned; but it is not necessarily conserved for the individual grid cells. So the code computes the interior flux corrections necessary for global mass conservation. These corrections are provided by the gradient of a scalar potential, obtained from the solution to a Poisson equation. Here the user has two options:

a. The interior flux components can be calculated from the user-specified velocities, inserted into the Poisson equation, and then corrected with the resulting scalar potential.

b. The interior flux components can be set to zero, inserted into the Poisson equation, and then corrected with the resulting scalar potential.

11. Option a preserves any circulation (vorticity) inherent in the user-specified velocity field, so the mass-conserving flux field will preserve the same vorticity. Option b eliminates all circulation inside the flow field (but not on the boundaries), and the mass-conserving flux field will be potential flow.

12. This completes the steps for a cold start. Hot-start calculations will begin by reading data from a previous output file before proceeding from this point.

13. Now the code marches the flow field through time, beginning with either (a) the initial mass-conserving fluxes (cold start) or (b) the mass-conserving fluxes and pressures from a previous code run (hot start). Since the existing algorithm is an explicit scheme, there is a numerical stability condition that limits the maximum allowable time-step size. The code calculates and prints this quantity for the initial flow, and for subsequent intervals designated by the user, so there need be little or no guesswork about what size time-step to use. For each code run, the user has to select the size of the time-step, the total number of time-steps, the first time-step to be stored, the subsequent intervals to be stored, the first time-

step for which information is to be printed, and the subsequent intervals for the same.

14. Information to be stored is written (unformatted) onto separate external files for plots and hot starts. The file used by the plot codes (VAXVEC and VAXCON) receives information at the user-designated intervals discussed in the preceding paragraphs. The file used for hot starts receives information at the same intervals.

Explanation of User Input

15. The VAXPVC code accepts user input from three namelists: BEGIN, PARAM, and INPUT. These namelists must appear in the run stream after the job-control command that executes the code. The first namelist is BEGIN, which contains only a single character variable START, whose possible values are as follows:

START = 'COLD'	Cold start.
'HOT'	Hot start (default).

16. Next comes namelist PARAM, which contains character, integer, and floating-point variables. The character variables and their possible values are as follows:

FLOW = 'FLOWRATE'	Net flow rate will be specified by user.
'OUTFLOW'	Net flow rate will be computed from user-specified outflow velocities.
'INFLOW'	Net flow rate will be computed from user-specified inflow velocities (default).
SOLVER = 'POINT'	Poisson equation will be solved by using point-successive over-relaxation (SOR) iteration scheme.
'CHECK'	Poisson equation will be solved by using checkerboard-SOR iteration scheme.
'IGCG'	Poisson equation will be solved by using idealized generalized conjugate-gradient (IGCG) iteration scheme (default).

TOPO = 'YES' Bottom topography will be calculated as the solution of a Laplace equation using the same solver as the Poisson equation.

'NO' Bottom topography will be specified only by user input (default).

RESIST = 'YES' Resistance due to bottom friction and no-slip condition will be included in the calculation (default).

'NO' Resistance due to bottom friction and no-slip condition will be omitted.

MAPOUT = 'YES' Symbolic map will be printed, showing cell types and positions in the computational plane (default).

'NO' No symbolic map will be printed.

RITINT = 'YES' Initial values of mass-conserving flow variables will be printed before first time-step begins.

'NO' Initial values will not be printed for flow variables (default).

RITOUT = 'YES' Flow variables will be printed whenever output is written on external files for storage (default).

'NO' Flow variables will not be printed when output is written on external files.

PUNITS = 'ENGLISH' Printed information will be in non-SI (English) units: feet, cubic feet, and seconds.

'METRIC' Printed information will be in SI (Metric) units: metres, cubic metres and seconds (default).

17. In namelist PARAM, the integer variables are as follows:

STEPS Number of time-steps to be computed. (Default = 1)

STORIT First time-step for which flow field is to be stored. (Default = 1)

STORINT	Number of elapsed time-steps before next storage of flow field. (Default = 1)
INFIT	First time-step for which condensed information is to be printed. (Default = 1)
INFINT	Number of elapsed time-steps before next print of condensed information. (Default = 1)
IPOTS	Number of iterations allowed by the Poisson solver for the initial mass-conserving flow field. (Default = 50)
ITERS	Number of iterations allowed by the Poisson solver for each predictor and each corrector phase of each time-step. (Default = 3)
IBOTS	Number of iterations allowed by the Laplace solver for calculating bottom topography. (Default = 50)
IREF	I-index of reference cell which will be assigned a zero value of pressure. Default will be first non-OUT cell to be found in a left-to-right, bottom-to-top search of the entire grid.
JREF	J-index of reference cell which will be assigned a zero value of pressure.
IMAP	Maximum width of printed map of cell types. Map will extend from left to right by this number of spaces, but will be vertically unbroken. Segments wider than IMAP will be continued on additional maps following. (Default = 130)

18. In namelist PARAM, the floating-point variables are as follows:

DELTAT	Size of time-step in seconds. (Default = 1.0)
VSTART	Dimensionless factor by which interior fluxes are multiplied before initial correction for conservation of mass. A value of 0.0 gives potential flow, and a value of 1.0 preserves any vorticity that may exist in the user-specified velocity field. (Default = 0.0)

AVIS — Dimensionless factor by which the general (global) viscosity is multiplied for calculating diffusion. (Default = 1.0)

BVIS — Dimensionless factor by which the general (global) viscosity is multiplied for calculating resistance due to the no-slip condition. (Default = 1.0)

SOR — Dimensionless parameter needed for accelerating convergence of point-SOR and checkerboard-SOR iteration. Value chosen should fall in range 1.0 to 2.0 for best convergence. Not used for IGCG iteration. (Default = 1.0)

19. The code reads namelists BEGIN and PARAM at the beginning of each run; but only those parameters in PARAM that are different from the values in the previous run need to be respecified for hot starts, since the code stores the existing values at the end of each run. Furthermore, only those parameters different from default values need ever be specified at all.

20. The namelists BEGIN and PARAM must be present in the run streams for all cold starts and hot starts. The namelist INPUT is read by the code only for cold starts, and its presence (or absence) has no effect on hot starts.

21. The namelist INPUT is used repeatedly in all run streams for cold starts, and the value of the character variable ITEM determines the kind and the amount of information read from INPUT by the code. Through its repeated appearance in the run stream, the namelist INPUT is used to specify cell types, general (global) values, section values, and line values, in that order.

22. The namelist INPUT is used first to specify user-designated cell types in the following manner:

a. A namelist INPUT appears, containing only the variable

ITEM = 'CELL TYPES'

b. Another namelist INPUT appears, containing the character variable ITEM and the integers I1 , I2 , J1 , and J2 , whose values indicate the following designations:

ITEM = 'OUT' OUT cells

'FLUX'	FLUX cells
'OPEN'	OPEN cells
'SLIP'	SLIP cells
'NOSLIP'	NOSLIP cells

I1	Smallest I-index for section
I2	Largest I-index for section
J1	Smallest J-index for section
J2	Largest J-index for section

c. Step b must be repeated for each distinct type of cell to be specified, and for each distinct section of cell types to be specified.

d. After all user-designated cell types have been specified, one more namelist INPUT must appear, containing only the character variable

ITEM = 'END'

23. The namelist INPUT is used next to specify user-designated general (global) values for certain parameters and flow variables in the following manner:

a. A namelist INPUT appears, containing only the variable

ITEM = 'GENERAL'

b. Another namelist INPUT appears, containing the character variables ITEM and UNITS, in addition to the floating-point variable VALUE, whose values indicate the following designations:

ITEM =	'COORD'	Units will be specified for grid.
	'FLOWRATE'	Net flow rate will be specified.
	'DEPTH'	Global depth will be specified.
	'X-VELOCITY'	Global x-component of velocity will be specified.
	'Y-VELOCITY'	Global y-component of velocity will be specified.
	'VISCOSITY'	Global kinematic viscosity will be specified.
	'MANNING'	Global value for Manning's "n" will be specified (no units).
UNITS =	'METRIC'	Specified units will be metric.
	'ENGLISH'	Specified units will be English.
VALUE		Specified (floating-point) value for designated quantity.

c. Step b must be repeated for each distinct quantity that is to be assigned a global value. UNITS default is 'METRIC'. VALUE default for 'FLOWRATE' and 'DEPTH' is 1. VALUE default for 'X-VELOCITY' and 'Y-VELOCITY' and 'MANNING' is 0. VALUE default for 'VISCOSITY' is 1.E-6. Note that UNITS and VALUE must be specified for each namelist INPUT unless the defaults are to be used. (Only UNITS need be specified for ITEM = 'COORD'.)

d. After all of the user-designated global values have been specified, one more namelist INPUT must appear, containing only the character variable

ITEM = 'END'

24. The namelist INPUT is used next to specify user-designated section values for depth, velocity, and bottom friction in the following manner:

a. A namelist INPUT appears, containing only the variable

ITEM = 'SECTION'

b. Another namelist INPUT appears, containing the character variables ITEM and UNITS, the one-dimensional (1-D) floating-point array VALUES, and the integers I1, I2, J1, and J2, whose values indicate the following designations:

ITEM =	'X-VELOCITY'	X-components of velocity
	'Y-VELOCITY'	Y-components of velocity
	'MANNING'	Manning's n (no units)
	'DEPTH'	Depth
	I1	Smallest I-index for section
	I2	Largest I-index for section
	J1	Smallest J-index for section
	J2	Largest J-index for section
	VALUES	Values specified for designated quantities in rectangular section of cells extending (inclusively) from I1 to I2 and from J1 to J2 in the computational plane. The code begins at J1 and reads the input values from I1 to I2 for each J-index through J2. In other words, the code reads row by row.
UNITS =	'METRIC'	SI (metric) units (default)
	'ENGLISH'	Non-SI (English) units

c. Step b must be repeated for each distinct quantity to be specified, as well as for each distinct grid section.

d. After all of the user-designated section values have been specified, one more namelist INPUT must appear, which contains only the character variable

ITEM = 'END'

25. The final use of the namelist INPUT is to designate lines in the computational plane (which may be curves in the physical plane), along which initial values will be specified by linear arc-length interpolation. In this case the user specifies only the values at the cell centers lying on the end points of the line. The code then computes the remaining intermediate values by interpolation so that the designated quantity varies linearly with arc length along the line or curve in the physical plane. The run stream must contain the following:

a. A namelist INPUT appears, containing only the variable

ITEM = 'LINE'

b. Another namelist INPUT appears, containing the character variables ITEM and UNITS, the 1-D floating-point array VALUES, and the integers I1 , I2 , J1 , and J2 , whose values indicate the following designations:

ITEM =	'X-VELOCITY'	X-components of velocity
	'Y-VELOCITY'	Y-components of velocity
	'MANNING'	Manning's n (no units)
	'DEPTH'	Depth
	I1	Smallest I-index for line
	I2	Largest I-index for line
	J1	Smallest J-index for line
	J2	Largest J-index for line
	VALUES	End-point values specified for designated quantity. The first input value will be assigned to the cell having indices (I1,J1). The second will be assigned to the cell having indices (I2,J2). One of the following conditions must be met: either I1 = I2 or J1 = J2
UNITS =	'METRIC'	SI (metric) units (default)
	'ENGLISH'	Non-SI (English) units

c. Step b must be repeated for each distinct quantity to be specified, as well as for each distinct grid line.

d. After all of the user-designated variables and grid lines have been specified for arc-length interpolation, one more namelist INPUT must appear, containing only the character variable

ITEM = 'END'

26. This completes the input needed to execute the VAXPVC code. Even if none of the options offered by namelist INPUT is exercised, namelist INPUT must nevertheless appear at least eight times in a run stream for a cold start, because the code expects it. The minimum acceptable input will be eight occurrences of namelist INPUT, each containing only one character variable (ITEM), in the following order:

```
ITEM = 'CELL TYPES'
ITEM = 'END'
ITEM = 'GENERAL'
ITEM = 'END'
ITEM = 'SECTION'
ITEM = 'END'
ITEM = 'LINE'
ITEM = 'END'
```

27. In other words, at least two INPUT namelists must appear, in this order, for each of the four possible user designations: cell types, general values, section values, and line values.

28. If the run stream lacks any of the essential namelists (BEGIN, PARAM, or INPUT), the code will not run properly. Furthermore, if the user intends to use non-SI (English) units, but fails to so designate for some input quantities, the code output will be worthless. The code uses SI (metric) units internally, and it converts input data to the same, based on the value specified for the character variable UNITS.

Code Dimensions

29. The two-dimensional arrays in the VAXPVC code are dimensioned from 0 to IDIM in the I-index, and from 0 to JDIM in the J-index. These dimensions must be large enough to accommodate the maximum indices of the grid cells (IMAX+1,JMAX+1), so in general IDIM has to be equal to or greater than IMAX+1 , and JDIM equal to or greater than JMAX+1 . For maximum efficiency, it is best to have IDIM = IMAX+1 and JDIM = JMAX+1 ; but this demands that the code be recompiled for each new set of grid dimensions. Such frugality is not imperative for small problems (say, 20 × 20), but it is recommended for larger problems (say, 50 × 50).

30. The dimensions IDIM and JDIM are specified internally in the code

with parameter statements; e.g.,

```
PARAMETER(IDIM = 32,JDIM = 32)
```

Whenever these array dimensions are to be changed, they must be given exactly the same integer value for each occurrence of the parameter statement; otherwise the code will not run properly. In addition, the codes used for vector and contour plots of the flow field must have the same integer values for IDIM and JDIM. It is not necessary, however, that IDIM = JDIM.

31. Grid cells with both indices lying in the range 1 to IMAX and 1 to JMAX may be of any cell type: FIELD, OUT, SLIP, NOSLIP, FLUX, or OPEN. Cells for which I = 0 or I = IMAX+1 , or for which J = 0 or J = JMAX+1 , will always be of type OUT. The outer rows (J = 0 and JMAX+1) and columns (I = 0 and IMAX+1) are strictly dummy cells for which no permanent information is stored, but they are needed for proper execution of the numerical algorithm. This is why the array dimensions are larger than the number of meaningful grid cells.

32. In setting up a calculation, it is important to see that there exist at least three non-OUT cells (horizontally, vertically, and diagonally) between each pair of OUT cells in the computational plane. This means that every pair of vertical boundary segments (facing each other in the computational plane) must be separated horizontally by at least three non-OUT cells. Similar pairs of horizontal boundary segments must be separated vertically in the same way. Convex boundary corners must be separated diagonally from each other by at least three non-OUT cells in the flow field as it appears in the computational plane.

33. On the other hand, obstacles (such as islands) and indentions (such as dikes) may be composed of single rows or columns of OUT cells, as long as they are separated vertically, horizontally, and diagonally by at least three non-OUT cells in each direction.

External Input and Output Files

34. The VAXPVC code is written in FORTRAN 77, and it reads grid and hot-start data from external files other than the standard input file (File 5). Likewise, it writes hot-start data and data for plotting to files

other than the standard output file (File 6). The job-control commands for accessing and attaching these files will vary from one computer system to another, but the specific files needed will usually be the same. The input and output files discussed here are distinct from namelists, which may be either included in the run stream or attached by the job control.

35. The computational grid, which is created in advance by a separate grid-generation code, is read unformatted from File 10. This is done in ENTRY GRIDL of SUBROUTINE ENTREE, by statements beginning with the command

READ(10)

36. Data to be used for subsequent hot starts are written unformatted to File 11. This is done in ENTRY WRITER of SUBROUTINE ENTREE, by statements beginning with the command

WRITE(11)

37. Data to be used later by the plot codes are written unformatted to File 12. This is done in ENTRY WRITER of SUBROUTINE ENTREE, by statements beginning with the command

WRITE(12)

38. Data from a previous run, to be used for a current hot start, are read unformatted from File 13. This is done in ENTRY READER of SUBROUTINE ENTREE, by statements beginning with the command

READ(13)

39. If the user wishes to overwrite the existing file containing the hot-start data, then statements beginning with READ(13) should be changed so that they all begin with the command

READ(11)

40. In this case File 13 will no longer be used, and it need not be included in the job control.