NOAA Technical Memorandum ERL GLERL-38

A TWO-DIMENSIONAL LAKE CIRCULATION MODELING SYSTEM

David J. Schwab
John R. Bennett
Andrew T. Jessup

Great Lakes Environmental Research Laboratory
Ann Arbor, Michigan
October 1981

UNITED STATES DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
John V. Byrne, Administrator

Environmental Research Laboratories
George H. Ludwig, Director

NOTICE

The NOAA Environmental Research Laboratories do not approve, recommend, or endorse any proprietary product or proprietary material mentioned in this publication. No reference shall be made to the NOAA Environmental Research Laboratories, or to this publication furnished by the NOAA Environmental Research Laboratories, in any advertising or sales promotion which would indicate or imply that the NOAA Environmental Research Laboratories approve, recommend, or endorse any proprietary product or proprietary material mentioned herein, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of this NOAA Environmental Research Laboratories publication.

CONTENTS

FIGURE

A TWO-DIMENSIONAL LAKE CIRCULATION MODELING SYSTEM*

David J. Schwab, John R. Bennett, and Andrew T. Jessup

This report documents a series of computer programs for modeling circulation and water level fluctations in well-mixed lakes The FORTRAN code for three computer programs is included in an appendix. The programs are 1) a grid generation and modification program, 2) a free surface circulation model, and 3) a rigid lid circulation model. The models are based on the vertically-integrated shallow water equations and include Coriolis effects, but neglect nonlinear acceleration terms and horizontal diffusion of momentum. The user of the programs is required to supply bathymetric data, initial conditions, external forcing function, and a subroutine to generate the desired output parameters. Other than these requirements, the programs are very general and can serve as a basis for developing more complex models. A sample run of each program is included in a second appendix.

1. INTRODUCTION

Numerical lake circulation models have been very successful in forecasting and hindcasting transient, large-scale water level fluctuations. More limited success has been achieved in modeling observed mean current patterns and transient current fluctuations for *well-mixed* lakes. Modeling of circulation in stratified lakes might still be considered an arcane art because of the complexity of the models, which include the interactions between thermal and inertial processes. Some fully stratified models have been developed and tested for selected cases, but they have not had the wide applications of well-mixed models. For references to specific models, Simons (1980) provides an extensive review of the state of the art in numerical modeling of lakes and inland seas.

In view of the vast difference in complexity between well-mixed and stratified models, we will concentrate in this initial report on the simpler, *well-mixed* models. These models are not applicable to baroclinic circulation, but in many situations the barotropic circulation is all that is required. As a first approximation, we also neglect nonlinear acceleration and horizontal diffusion of momentum as compared to first-order acceleration and Coriolis terms. This precludes application of the model to near-coastal zones with large horizontal current shear. Although the models developed with these approximations are applicable mainly to whole-lake processes in well-mixed lakes, they can also serve as a basis for the development of fully stratified and nonlinear models.

*GLERL Contribution No. 280.

Within the framework of the above approximations, there are two distinct categories of water motion, namely the gravitational response and the non-divergent response. The gravitational response is characterized by divergent motion, with a significant ratio of potential to kinetic energy. The time scale of this response is proportional to the length of the lake divided by the speed of long gravity waves (the seiche period) and is on the order of minutes to hours for most lakes. For the non-divergent response, however, most of the energy is kinetic and the time scale is larger than the seiche period.

We have developed two numerical circulation models to simulate the gravitational and non-divergent response of a lake. Our intention was to build the simplest models that could encompass all significant dynamic processes for the large-scale circulation. The free surface (gravitational response) model is intended to be used mainly for forecasting and hindcasting water level fluctuations. The rigid lid (non-divergent) model simulates the transient, large-scale circulation in well-mixed lakes. Although the free surface model is more general, the neglect of free surface fluctuations in the rigid lid model results in a considerable savings in computer time and allows one to "se a more accurate finite difference scheme for the Coriolis terms.

Computer programs for the models were designed for maximum portability and ease of implementation. The user is required to supply initial conditions, external forcing in terms of overwater meteorological conditions, and a subroutine to handle the required output parameters. We feel that the FORTRAN programs themselves (appendix A) should require little or no modification by the "ser. The structure of the models is also intended to serve as a building block for more complex models.

2. THE GRID SYSTEM AND GRID GENERATION PROGRAM

The shape and bathymetry of the lake are represented by the average depths of grid boxes for an array of square grid boxes covering the lake. Land squares are assigned zero depth. For the rigid lid and free surface circulation models, we require that the grid data be in the same format as that used by Schwab and Sellers (1980). This format is well-documented in that report and is implemented for 2-km squares on the five Great Lakes and 1.2-km squares on Lake St. Clair. In addition to average depths, the format requires such auxiliary information as number of rows and columns in the grid, the latitude and longtitude of the grid origin, the grid size, orientation of the grid relative to east-west, and parameters for converting latitude and longtitude to grid units and vice versa.

For many applications, it is desirable to generate a grid with coarser resolution than the 2-km Great Lakes grids, or to rotate or otherwise modify the grids. The FORTRAN program GRID designed to carry out these functions is given in appendix A. The program reads a grid data file that conforms to the format given in Schwab and Sellers (1980), generates a new grid with specified grid size and rotation, modifies individual boxes according to the user's specifications, and produces a new grid data file in the standard format. The rigid lid and free surface circulation models require that the bathymetric grid have at least one row of zero-depth grid squares on all four sides. The program also performs this function.

Once a grid size and orientation have been selected, it is intended that the program be run twice, once to examine the new bathymetric grid, and a second time to modify individual grid squares in the new grid. Grid modifications are usually required to remove land-locked water squares, to increase the depth in artificially shallow water squares, and to otherwise modify the grid to conform to known shoreline configurations. For the rigid lid model, it may also be desirable to change all island grid squares to shallow water squares, as will be discussed later.

3. THE FREE SURFACE MODEL

3.1 Model Equations

The vertically-integrated shallow water equations, neglecting non-linear acceleration terms and horizontal diffusion of momentum, are

$$\frac{\partial M}{\partial t} - fN = -gD\,\frac{\partial H}{\partial x} + \rho^{-1}\tau_x^s - \rho^{-1}\tau_x^B \qquad (1)$$

$$\frac{\partial N}{\partial t} + fM = -gD\,\frac{\partial H}{\partial y} + \rho^{-1}\tau_y^s - \rho^{-1}\tau_y^B \qquad (2)$$

$$\frac{\partial H}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0, \qquad (3)$$

where x, y, t are the independent variables in space and time; M, N are the components of the vertically-integrated transport vector in the x, y directions; H is the displacement of the free surface from its mean level; D is the depth; f is the Coriolis parameter ($f = 2\Omega \sin\phi$, where Ω is the angular speed of rotation of the earth and ϕ is latitude); g is the gravitational constant; ρ is water density (assumed uniform); τ_x^s, τ_y^s are the x, y components of the wind stress vector; and τ_x^B, τ_y^B are the x, y components of the bottom stress vector. We will assume also that the free surface fluctuation is small compared to water depth so that D can be taken as depth relative to the mean water level. The boundary condition for (1)-(3) is that there be no transport normal to the shoreline.

If surface stress is specified as a function of x, y, and t and bottom stress is parameterized as a function of transport, (1)-(3) form a closed system of equations that can be solved for M, N, and H as functions of x, y, and t. Bottom stress is parameterized in terms of transport by a quadratic drag law, i.e.,

$$\tau_B^x = \frac{c_D}{D^2}\sqrt{M^2 + N^2}\ M \tag{4}$$

$$\tau_B^y = \frac{c_D}{D^2}\sqrt{M^2 + N^2}\ N, \tag{5}$$

where c_D is a constant drag coefficient taken here as 2×10^{-3}. This parameterization reduces to the boundary layer law that surface stress is equal to drag coefficient times the square of the velocity at a reference level if the vertically averaged velocity (transport divided by depth) is taken as the reference velocity.

3.2 The Finite Difference Scheme

The variables in (1)-(3) are discretized and defined at the points shown in figure 1 for each grid box in the bathymetric grid. Water depth and the free surface fluctuation are defined at the center of the grid box, the x-component of transport at the center of the right side, and the y-component of transport at the center of the top side. This distribution of variables is known as a single Richardson lattice and it facilitates discretization of the spatial derivations in (1)-(3). With this lattice arrangement, we can represent the spatial derivatives as central differences and incorporate the boundary condition of vanishing transport normal to the shorelines very naturally. The variables, however, are not defined at the same grid points and the free surface fluctuation is not defined at the shoreline, where it is usually required. These inconveniences are outweighed by the energy conservation properties and exact satisfaction of the boundary condition with a Richardson lattice.

The time derivatives are also represented by centered differences. The discretized free surface fluctuation is defined at $t = (2n - 1)\Delta t/2$ and the transports at $t = n\Delta t$, where Δt is the discretization interval and n = 1, 2, 3,.... This staggering of the variables in time results in an explicit, leap-frog timestepping scheme. The stability criterion for this scheme (neglecting friction) is

$$\Delta t \leq \left(\frac{2g\, D_o}{\Delta s^2} + \frac{f^2}{4}\right)^{-1/2}, \tag{6}$$

where Δs is the discretization interval in space and D_o is the maximum depth. This criterion guarantees that the fastest possible gravity wave can travel no further than half the diagonal of a grid square in a single time step. As will be shown in the next section, the viscous terms impose the additional stability criterion

$$\Delta t \leq \frac{2D}{c_D \cdot SPD},$$

but (6) is generally the more stringent condition.

For a single Richardson lattice, the x-component of transport is not defined at the same points as the y-component. Thus, to compute the Coriolis term, one must average the four nearest values to obtain the required transport component. Platzman (1972) found that, if the transport values are weighted by the inverse depth in the averaging process, the self-adjointness (and therefore energy conservation properties) of the finite difference equations is maintained. Platzman also showed that computing one of the components of the Coriolis force at the previous time step and the other with the updated values led to a leapfrog time differencing scheme and thus second-order accuracy. However, his analysis of the complete finite difference system revealed that there is a first-order error in the time derivative proportional to gf.

The finite difference forms for (1)-(3) on the grid shown in figure 1 are then

$$H_{i,j}^{n+1/2} = H_{i,j}^{n-1/2} - \Delta t \;(M_{i,j}^{n} - M_{i-1,j}^{n} + N_{i,j}^{n} - N_{i,j-1}^{n})/\Delta s \tag{7}$$

$$M_{i,j}^{n+1} = (1 - c_D\, \Delta t\, S_{i,j}^{n} / \overline{D}_{i,j}^{M})\, M_{i,j}^{n} + f\, \Delta t\, \overline{N}_{i,j}^{n}$$

$$- g \, \Delta t \, \overline{D}^{M}_{i,j} \, (H^{n+1/2}_{i+1,j} - H^{n+1/2}_{i,j})/\Delta s \; + \; \Delta t U^{n+1}_{i,j} \tag{8}$$

$$N^{n+1}_{i,j} = (1 - c_D \, \Delta t \, T^{n}_{i,j}/\overline{D}^{N}_{i,j})N^{n}_{i,j} - f \, \Delta t \, \overline{M}^{n+1}_{i,j}$$

$$- g \, \Delta t \, \overline{D}^{N}_{i,j} \, (H^{n+1/2}_{i,j+1} - H^{n+1/2}_{i,j}) /\Delta s + \Delta t \, V^{n+1}_{i,j}, \tag{9}$$

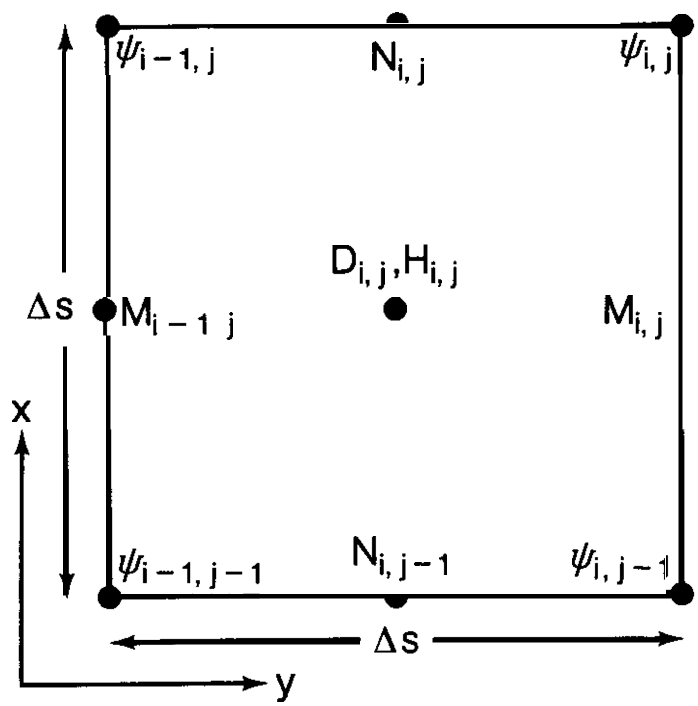

FIGURE 1.--*Definition of variables in grid box i,j for lake circulation models.*

where the following symbols have been defined:

$$H^{n+1/2}_{i,j} = H[(i+1/2)\Delta s,\ (j+1/2)\Delta s,\ (2n+1)\Delta t/2]$$

$$M^{n}_{i,j} = M(i\Delta s,\ (j+1/2)\Delta s,\ n\ \Delta t)$$

$$N^{n}_{i\ j} = N[(i+1/2)\Delta s,\ j\Delta s,\ n\Delta t]$$

$$D_{i,j} = D[(i+1/2)\Delta s,\ (j+1/2)\Delta s]$$

$$\overline{D}^{M}_{i,j} = (D_{i,j} + D_{i+1,j})/2$$

$$\overline{D}^{N}_{i,j} = (D_{i,j} + D_{i,j+1})/2$$

$$S^{n}_{i,j} = (M^{n}_{i,j})^{2} + (\overline{N}^{n}_{i,j})^{2}/\overline{D}^{M}_{i,j}$$

$$T^{n}_{i,j} = (\overline{M}^{n+1}_{i,j})^{2} + (N^{n}_{i,j})^{2}/\overline{D}^{N}_{i,j}$$

$$\overline{N}^{n}_{i,j} = \overline{D}^{M}_{i,j}\,[(1/\overline{D}^{M}_{i,j} + 1/\overline{D}^{N}_{i,j})N^{n}_{i,j} + (1/\overline{D}^{M}_{i\ j} + 1/\overline{D}^{N}_{i\ j-1})N^{n}_{i,j-1}$$

$$+ (1/\ \overline{D}^{M}_{i\ j} + 1/\overline{D}^{N}_{i+1\ j})N^{n}_{i+1,j} + (1/\overline{D}^{M}_{i\ j} + 1/\overline{D}^{N}_{i+1,j-1})N^{n}_{i+1,j-1}]/8$$

$$\overline{M}^{n}_{i,j} = \overline{D}^{N}_{i,j}\,[(1/\overline{D}^{N}_{i,j} + 1/\overline{D}^{M}_{i-1,j+1})M^{n}_{i-1,j+1} + (1/\overline{D}^{N}_{i,j} + 1/\overline{D}^{M}_{i-1,j})M^{n}_{i-1,j}$$

$$+ (1/\overline{D}^{N}_{i\ j} + 1/\overline{D}^{M}_{i\ j+1})M^{n}_{i\ j+1} + (1/\overline{D}^{N}_{i,j} + 1/\overline{D}^{M}_{i,j})M^{n}_{i,j}]/8$$

$$U^n_{i,j} = \rho^{-1} \tau^s_x [i \Delta s, (j+1/2)\Delta s, n \Delta t]$$

$$V^n_{i,j} = \rho^{-1} \tau^s_y [(i+1/2)\Delta s, j \Delta s, n \Delta t].$$

The discretized equations are implemented in the FORTRAN program FRSFC in appendix A. The symbols S and T in (8) and (9) are represented by the variable SPD in the FORTRAN program. U and V are calculated by the function TAU(T,I,J,K) with K = 1 and K = 2, respectively.

The program requires four forms of input. First, a bathymetric data file in the format used by Schwab and Sellers (1980) must be supplied. Second, a control record with the desired timestep, duration of run, and added water level is required. The added water level is intended to adjust for seasonal changes in the mean water level relative to the depths given in the bathymetric data file. Third, if initial conditions other than H = M = N = 0 are desired, a subroutine with the name INIT is required to set them. Transport values set as initial conditions on the boundary of the grid are not changed during the run so that inflow and outflow conditions can be accommodated. Note that if the sum of the inflows does not equal the sum of the outflows, the mean water level will change. Fourth, the forcing function (τ^s_x, τ^s_y) must be specified. This can be done either by supplying a function TAU(T,I,J,K) that returns stress divided by density in MKS units or by supplying wind observations to the function TAU given in appendix A. The details of the wind stress calculation from wind observations are given in section 5. An output subroutine named OUTP is called each timestep and must be supplied by the user of the program to handle the specific output requirements for a particular application.

4. THE RIGID LID MODEL

4.1 Model Equations

The momentum equations for the rigid lid model are the same as equations (1) and (2) for the free surface model. However, in the continuity equation the term $\frac{\partial H}{\partial t}$ is dropped, leaving:

$$\frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0, \qquad (10)$$

This constraint of no horizontal divergence allows the two transport components to be expressed in terms of a single variable, ψ, the stream function.

$$M = -\frac{\partial\psi}{\partial y};\; N = \frac{\partial\psi}{\partial x} \tag{11}$$

A further consequence of the rigid lid approximation is that the two momentum equations (eqs. 1 and 2) can be combined into a single equation for ψ:

$$\frac{\partial}{\partial t}\,(\nabla \cdot D^{-1}\;\nabla\psi) + fJ\,(\psi, D^{-1}) = \frac{\partial}{\partial x}\,\left(\frac{\tau_y^S - \tau_y^B}{\rho D}\right) - \frac{\partial}{\partial y}\,\left(\frac{\tau_x^S - \tau_x^B}{\rho D}\right), \tag{12}$$

where

$$J(\psi, D^{-1}) = \frac{\partial\psi}{\partial x}\frac{\partial D^{-1}}{\partial y} - \frac{\partial\psi}{\partial y}\frac{\partial D^{-1}}{\partial x} \tag{13}$$

is the Jacobian.

The boundary condition of no flow into the shoreline will be satisfied as long as ψ is a constant on any contiguous segment of shore. In the simplest case of a lake with no rivers and no islands, the boundary values can simply be set to zero. For rivers, ψ is constant for each shoreline segment between the rivers, and the difference of ψ across the river mouth is the river transport in units of square meters per second. These boundary values must be set by the user in subroutine INIT.

In general, the value of the stream function along an island coast is different from the value at the shore and must be computed from the model equations. The proper way to do this is to integrate the pressure gradient term around the island; the condition that pressure is a single valued function yields an equation between the boundary and interior values of ψ. This model, however, does not do this calculation. Therefore, we recommend treating the islands as shallow water. This can be done when program GRID is run for the second time by simply setting all island depths to the minimum depth.

4.2 The Finite Difference Scheme and Computer Program RLID

Program RLID reads in the bathymetric and grid data generated by program GRID, but unlike the free surface model, it modifies this information before using it. The finite difference scheme requires the inverse depth at the stream function points rather than the depth at water level points. For all interior points (stream function points with no surrounding water level grid points on land), a four-point average is used to compute the depth. For all points for which any of the surrounding water level points are land, the depth is set equal to 0.9 HMIN. After this, the inverse depth is computed.

Although the model uses a nonlinear bottom friction term, the current speed is calculated from the previous time step. Thus, the relation between ψ^n and ψ^{n+1} is linear. Once the speed array is calculated, this relation becomes

$$\frac{\nabla \cdot D^{-1}\ \nabla\psi^{n+1} - \nabla \cdot D^{-1}\ \nabla\psi^{n}}{\Delta t} + fJ(\overline{\psi}^{n}, D^{-1}) =$$

$$\frac{\partial}{\partial x}\left(\frac{\tau_y^S}{\rho D}\right) - \frac{\partial}{\partial y}\left(\frac{\tau_x^S}{\rho D}\right) - \frac{c_D}{\rho}\left(\frac{\partial}{\partial x}\ \frac{SPD^n\ \overline{N}^n}{D^2} - \frac{\partial}{\partial y}\ \frac{SPD^n\ \overline{M}^n}{D^2}\right), \qquad (14)$$

where the operator $\overline{(\quad)}^{n}$ is defined as

$$\overline{(\quad)}^{n} = \frac{(\quad)^{n+1} + (\quad)^{n}}{2}.$$

In equation (14), the vorticity and the transport components are evaluated with elementary second-order difference formulas, with appropriate two-point averages to compute D^{-1} at points other than at stream function points. However, the Jacobian term is evaluated according to Arakawa's (1970) method as follows:

$$
\begin{aligned}
12\,\Delta s^2\, J(\psi, D^{-1}) = {} & \psi_{i+1\ j}\,(D^{-1}_{i,j+1} + D^{-1}_{i+1,j+1} - D^{-1}_{i,j-1} - D^{-1}_{i+1,j-1}) \\
& + \psi_{i-1,j}\,(-D^{-1}_{i,j+1} - D^{-1}_{i-1,j+1} + D^{-1}_{i,j-1} + D^{-1}_{i-1,j-1}) \\
& + \psi_{i,j+1}\,(-D^{-1}_{i+1,j} - D^{-1}_{i+1,j+1} + D^{-1}_{i-1,j} + D^{-1}_{i-1,j+1}) \\
& + \psi_{i\ j-1}\,(D^{-1}_{i+1,j} + D^{-1}_{i+1,j-1} - D^{-1}_{i-1,j} - D^{-1}_{i-1,j-1}) \\
& + \psi_{i+1,j+1}\,(-D^{-1}_{i+1,j} + D^{-1}_{i,j+1}) + \psi_{i+1,\ j-1}\,(D^{-1}_{i+1,\ j} - D^{-1}_{i,j-1}) \\
& + \psi_{i-1\ j+1}\,(D^{-1}_{i-1,j} - D^{-1}_{i,j+1}) + \psi_{i-1,\ j-1}\,(-D^{-1}_{i-1,\ j} + D^{-1}_{i,j-1}).
\end{aligned}
$$

Thus equation (14) is a complicated linear equation between ψ^{n+1} and ψ^n, which is solved by a relaxation method. Program RLID in appendix A uses an over-relaxation factor of 1.65, which minimizes the total number of iterations for runs on all five Great Lakes and Lake St. Clair. The optimal factor for individual runs varies from 1.4 to 1.9. If total economy is required, this factor may be more finely tuned for a given application.

Because of the averaging of the Jacobian and friction terms over two time levels, the method is formally computationally stable for all values of Δt. However, there are two practical limits to Δt. First, if $f\Delta t$ is greater than about 1.5 (in the Great Lakes this corresponds to Δt = 4h), the model begins to consume more computer time per unit of real time than if a lower time step is used. The reason for this is that the Coriolis term makes the matrix relaxation problem ill-conditioned and more iterations are required. For higher values of $f\Delta t$, the relaxation process will not converge at all. The second practical limit is because of the friction term and can be illustrated by the following simple equation:

$$
\frac{\partial M}{\partial t} = -\,\frac{c_D \cdot SPD}{D}\, M. \tag{15}
$$

This equation may describe the decay of currents in a shallow region near the coast. The finite difference version is

$$M^{n+1} = M^n - \frac{\Delta t\ c_D \cdot SPD}{2D} (M^{n+1} + M^n).$$

Solving for M^{n+1} yields

$$M^{n+1} = M^n \frac{\left(1 - \frac{\Delta t\ c_D \cdot SPD}{2D}\right)}{\left(1 + \frac{\Delta t\ c_D \cdot SPD}{2D}\right)} \tag{16}$$

If the parameter $\left(\frac{\Delta t\ c_D \cdot SPD}{2D}\right)$ is greater than one, the current decays in an oscillatory manner instead of monotonically as the solutions of (15) do. For much larger values of this parameter, the magnitude of the current decreases slower than for smaller time steps--a clearly undesirable feature. Thus for computational economy and for numerical accuracy, there are two practical limits for the time step,

$$\Delta t \leq \frac{1.5}{f}$$

and

$$\Delta t \leq \frac{2D}{c_D \cdot SPD}.$$

The FORTRAN program RLID, given in appendix A, implements the rigid lid model. The user must supply the same data and subroutines as described for program FRSFC in the previous section, except that initial conditions are for the stream function, ψ (FORTRAN array S in program RLID), and the output routine has access only to the stream function and speed arrays.

5. DETERMINATION OF WIND STRESS

The user of programs FRSFC or RLID has the option of specifying wind stress directly by supplying a function TAU(T,I,J,K) or using the function TAU given in appendix A with a set of wind observations. If wind observations are used, they are assumed to represent overwater conditions. The required parameters for each observation are:

(1) Time from the beginning of the run,

(2) Latitude (in degrees north),

(3) Longitude (in degrees west),

(4) Height above water at which observations were made (in meters),

(5) Air temperature (in degrees centigrade),

(6) Water temperature (in degrees centigrade),

(7) Wind speed (in meters per second), and

(8) Compass direction from which wind is blowing (in degrees).

Function TAU first converts each wind observation into surface stress components with subroutine UZL (appendix A). This subroutine calculates the bulk aerodynamc coefficients for momentum and heat over a lake surface as functions of wind speed and air - lake temperature difference. It is based on the Businger *et al.* (1971) formulation of stability length and Charnock's (1955) formula. The constant in Charnock's formula is chosen so that under neutral conditions the 10-m drag coefficient is 1.6×10^{-3}. After all wind observations for the same time are converted to stress, the two-dimensional stress field is calculated as a linear function of x and y by least-squares fitting of each component to the stress components at the wind observation points. The stress field at time t is then

$$\tau_x^s\,(x,y,t) = a_1(t)x + b_1(t)y + c_1(t) \text{ and} \tag{17}$$

$$\tau_y^s\,(x,y,t) = a_2(t)x + b_2(t)y + c_2(t), \tag{18}$$

where the coefficients a_1-c_2 are determined by least-squares fitting. If all wind observations are at the same x value, $a_1 = a_2 = 0$. If all wind observations are at the same y value, $b_1 = b_2 = 0$. If only one wind observation is given, $a_1 = b_1 = a_2 = b_2 = 0$. Between times at which wind observations are made,(17)-(18) are still used, but the values a_1-c_2 are linearly interpolated between the wind observation times. If wind stress is required at a time greater than the last wind observation time, wind stress from the last wind observation time is used.

6. REFERENCES

Arakawa, A. (1970): Numerical simulation of large-scale atmospheric motions. In: *Proceedings of a Symposium in Applied Mathmatics*, G. Birkhoff and S. Varga, eds., American Mathematical Society, Durham, N.C. pp. 24-40.

Businger, J. A., J. C. Wyngaard, Y. Izumi, and E. F. Bradley (1971): Flux-profile measurements in the atmospheric surface layer. *J. Atmos. Sci.* 28:181-189.

Charnock, H. (1955): Wind stress on a water surface. *Quart. J. Roy. Meteor. Soc.* 81:639.

Platzman, G. W. (1972): Two-dimensional free oscillations in natural basins. *J. Phys. Oceanogr.* 2:117-138.

Schwab, D. J., and D. L. Sellers (1980): Computerized bathymetry and shorelines of the Great Lakes, NOAA Data Report ERL-GLERL-16, National Technical Information Service, Springfield, Va. 22151. 13 pp.

Simons, T. J. (1980): Circulation models of lakes and inland seas. Bulletin 203, The Canadian Bulletins of Fisheries and Aquatic Sciences, Department of Fisheries and Oceans, Ottowa, Ont. 146 pp.

Appendix A. FORTRAN computer programs GRID, FRSFC, RLID, and associated subroutines.

```
C******************************PROGRAM GRID******************************
C
C                    BATHYNETRIC GRID GENERATION PROGRAM
C
C PURPOSE :
C                 THE PURPOSE OF THIS PROGRAN IS TO READ DATA FROM A
C                 STANDARDIZED BATHYMETRIC DATA FILE AND GENERATE
C                 A NEW BATHYMETRIC GRID WITH USER SPECIFIED
C                 GRID SIZE AND ORIENTATION. THE STANDARDIZED BATHY-
C                 METRIC DATA FILES FOR THE GREAT LAKES ARE DESCRIBED IN
C                 SCHWAB AND SELLERS (1980):'COMPUTERIZED BATHYMETRY AND
C                 SHORELINES OF THE GREAT LAKES', NOAA DATA REPORT
C                 ERL-GLERL-16.
C
C                 THE PROGRAM GRID READS THE BATHYMETRIC DATA FILE, READS
C                 THE DESIRED GRID SIZE AND ORIENTATION (RELATIVE TO THE
C                 ORIGINAL GRID), READS GRID MODIFICATION PARAMETERS,
C                 AND GENERATES A NEW BATHYMETRIC DATA FILE BASED ON
C                 THESE PARAMETERS.
C
C ALGORITHM :
C                 AFTER READING THE INPUT BATHYMETRIC FILE (USING SUB-
C                 ROUTINE RGRID), THE PROGRAN FIRST DETERMINES THE SIZE
C                 OF THE NEW GRID (I.E.,MAXIMUM VALUES OF I AND J). THE
C                 NEW GRID IS SCANNED, SQUARE BY SQUARE, TO CALCULATE AV-
C                 ERAGE DEPTHS. FOR A GRID SQUARE TO BE CONSIDERED WATER,
C                 THREE OF THE FIVE POINTS (O,O), (D,D), (D,-D), (-D,-D),
C                 (-D,D) RELATIVE TO THE CENTER OF THE SQUARE (WHERE
C                 D=GRID SIZE/4) MUST FALL IN A WATER SQUARE OF THE ORIG-
C                 INAL GRID. THE AVERAGE DEPTH IS THE AVERAGE OF THE
C                 THREE, FOUR, OR FIVE WATER POINTS. WHEN THE NEW GRID
C                 IS FILLED. ROWS AND COLUMNS ARE ADDED OR ELIMINATED
C                 UNTIL THE GRID IS BORDERED BY A SINGLE ROW OR COLUMN
C                 OF LAND POINTS ON ALL SIDES. THIS TRIMMING AND THE
C                 TRANSLATION OF AXES TO ENCOMPASS THE OLD GRID DETERMINE
C                 THE I DISPLACEMENT AND J DISPLACEMENT. I DISPLACEMENT
C                 IS THE NUMBER OF NEW GRID SQUARES IN THE X-DIRECTION
C                 FROM THE NEW GRID ORIGIN TO THE OLD GRID ORIGIN. J
C                 DISPLACEMENT IS THE NUMBER OF NEW GRID SQUARES IN THE
C                 Y-DIRECTION FROM THE NEW GRID ORIGIN TO THE OLD GRID
C                 ORIGIN. ANY CHANGES INDICATED ON THE GRID MODIFICATION
C                 CARDS ARE THEN MADE. THE NEW GRID IS OUTPUT AS A .
C                 STANDARD BATHYNETRIC DATA FILE (UTILIZING THE SUBROUTINE
C                 WGRID).
C
C INPUT :
C    LOGICAL UNIT 5:
C       RECORD 1:                                      FORMAT   COLUMN
C          DEL  - DESIRED GRID SIZE IN METERS          G20.5     I-20
C                   (ANY FRACTION OF A METER IS IGNORED)
```

```
C        DEG  - ORIENTATION  OF  NEW  GRID  RELATIVE  TO
C               BASE  GRID.  IN  DEGREES.  POSITIVE  FOR
C               COUNTER-CLOCKWISE,  NEGATIVE  FOR
C               CLOCKWISE.                               G20.5     21-40
C               (MUST BE  BETWEEN  -go  AND  +90)
C
C      RECORD  2  AND  FOLLOWING:
C        ICHG -  I-COORDINATE  OF  GRID  BOX  TO  MODIFY     I5       1-5
C        JCHG -  J-COORDINATE  OF  GRID  BOX  TO  MODIFY     I5       6-10
C          (ICHG  AND  JCHG  REFER  TO  THE  NEW  GRID
C             COORDINATE   SYSTEM)
C        IDEP - NEW  DEPTH  OF  GRID  BOX  IN  METERS        I5       11-15
C
C        *NOTE:  END  OF  INPUT  IS  INDICATED  BY  A  RECORD  WITH  ZERO  ICHG
C        SO  THAT  AT  LEAST  ONE  BLANK  RECORD  MUST  FOLLOW  RECORD  1.
C
C    LOGICAL  UNIT  7:
C      BATHYMETRIC  DATA  FILE:
C        THE  FORMAT  OF  THE  STANDARDIZED  BATHYMETRIC  DATA  FILE  IS
C        DESCRIBED  IN  SCHWAB  AND  SELLERS  (1980):  'COMPUTERIZED
C        BATHYHETRY  AND  SHORELINES  OF  THE  GREAT  LAKES',  NOAA
C        DATA   REPORT  ERL-GLERL-16.   ONLY  THE  BATHYMETRIC  PART
C        OF  THE  FILE  IS  USED.   TWO  ADDITIONAL  FIELDS  ARE
C        REQUIRED  ON  DATA  HEADER  RECORD  2.  THESE  ARE:
C
C                                                    FORMAT      CARD   COLUMN
C               MINIMUM  DEPTH                         I5           45-49
C               BASE  GRID  ROTATION  FROM  E-W      F6.2           50-55
C
C OUTPUT :
C    LOGICAL  UNIT  6:
C      PRINTED   OUTPUT
C
C    LOGICAL  UNIT  8:
C      NEW  BATHYMETRIC  DATA  FILE:
C               THE  NEW  BATHYMETRIC  DATA  FILE  HAS  THE  SAME  FORMAT  AS
C               THE  STANDARDIZED  BATHYMETRIC  DATA  FILE  WITH  THE  ADDITION
C               OF  FIVE  FIELDS  ON  DATA  HEADER  RECORD  2.  THESE  ARE:
C
C                                                    FORMAT      CARD   COLUMN
C               MINIMUM  DEPTH                         I5           45-49
C               BASE  GRID  ROTATION  FROM  E-W      F6.2           50-55
C               I  DISPLACEMENT                        I5           56-60
C               J   DISPLACEMENT                       I5           61-65
C               ROTATION  ANGLE  FROM  BASE          ~6.2           66-71
C                    (ANGLES  ARE  MEASURED  IN  DEGREES  COUNTERCLOCKWISE)
C
C               *NOTE  THAT  THE  I  DISPLACEMENT,  J  DISPLACEMENT,  AND
C               ROTATION  FROM  BASE  FIELDS  FOR  THE  INPUT  BATHYMETRIC
C               GRID  MUST  BE  ZERO.
```

```
C
C COMMON BLOCKS :
C                 /GPARM/ RPARM(23),IPARM(54) - REAL AND INTEGER
C                   PARANETERS DESCRIBING THE BATHYMETRIC GRID - SEE
C                   SUBROUTINE RGRID.
C
C  SUBROUTINES:
C       RGRID - READS THE BATHYMETRIC DATA FILE
C       WGRID - WRITES THE BATHYMETRIC DATA FILE
C       PGRID - PRINTS GRID PARAMETERS AND GRID USING SUBROUTINE PRNT
C       PRNT -  PRINTS THE GRID
C
C HISTORY :
C                 WRITTEN BY A. T. JESSUP, 1981, GLERL, ANN ARBOR, MI
C
C
      DIMENSION NDEPTH(314,260), DEPTH(314,260)
      DIMENSION x(5), Y(5). XP(5), YP(5), IP(5), Jp(5)
      COMMON /GPARM/ RPARM(23), IPARM(54)
C
C IDIM AND JDIM ARE THE FIRST AND SECOND DIMENSIONS
C  OF THE ARRAYS NDEPTH AND DEPTH
C
      DATA IDIM, JDIM /314, 260/
      DTR = ATAN(1.) / 45.
      READ (5.470) DEL, DEG
      WRITE (6,480) DEL, DEG
      IF (DEG .GT. 90..OR. DEG .LT. -90.) GO TO 450
C
C  READ GRID FROM LOGICAL UNIT 7 AND SET PARAHETERS IN
C  COMMON BLOCK /GPARM/
C
      CALL RGRID(7, DEPTH, IDIM, JDIM)
      WRITE (6.490) (IPARM(I),I=5,54)
      MN = IPARM(1)
      NN = IPARM(2)
      DO 10 I = 1, MM
        DO 10 J = 1. NN
          NDEPTH(I,J) = INT(DEPTH(I,J))
   10 CONTINUE
      CALL PGRID(6, DEPTH, IDIM)
C
C  THE INPUT GRID MUST NOT HAVE IDISP, JDISP, OR A ROTATION
C  FROM THE BASE - IT MUST BE AN ORIGINAL BASE GRID
C
      IF (IPARM(3) .NE. 0 .OR. IPARM(4) .NE. 0 .OR. RpARM(7) .NE. 0.)
     1   GO TO 440
      IDEL = DEL
      DEL = IDEL
      ALPHA = DTR * DEG
```

```
C
C   DETERMINE THE MAXIMUM DIMENSIONS (M,N) OF NEW GRID
C
      M1  = INT(IPARM(1)*COS(ALPHA)*RPARM(3)/DEL + 1.)
      M2 = INT(ABS(IPARM(2)*SIN(ALPHA)*RPARM(3)/DEL) + 1.)
      M = M1 + N2
      N1 = INT(ABS(IPARM(1)*SIN(ALPHA)*RPARM(3)/DEL) + 1.)
      N2 = INT(IPARM(2)*COS(ALPHA)*RPARM(3)/DEL + 1.)
      N = N1 + N2
       IF ((M .GT. IDIM).OR.(N .GT. JDIM)) GO TO 460
C
C   DETERHINE NEW DEPTHS FOR EACH NEW GRID SQUARE
C
      DO 60 I = 1, M
        DO 60 J = 1, N
          SUMD = 0.
          IWATER = 0
C
C   FOR A GRID SQUARE TO BE CONSIDERED WATER, 3 OF THE 5 POINTS
C   (0,0), (D,D), (D,-D), (-D,-D), (-D,D) RELATIVE TO THE CENTER
C   OF THE SQUARE (WHERE D=GRID SIZE/4) MUST FALL IN A WATER
C   SQUARE OF THE ORIGINAL GRID
C
          X(1) = DEL * (FLOAT(I) - .5)
          X(2) = X(1) + DEL / 4.
          X(3) = X(2)
          X(4) = X(1) - DEL / 4.
          X(5) = X(4)
          Y(1) = DEL * (FLOAT(J) - .5)
          Y(2) = Y(1) + DEL / 4.
          Y(3) = Y(1) - DEL / 4.
          Y(4) = Y(3)
          Y(5) = Y(2)
C
C    TRANSFORM EACH POINT OF THE NEW GRID (X,Y) TO THE OLD
C    GRID (XP,YP)
C
          DO 40 K = 1, 5
C
C   FIRST ROTATE
C
            XP(K) = X(K) * COS (ALPHA) - Y(K) * SIN (ALPHA)
            YP(K) = Y(K) * COS (ALPHA) + X(K) * SIN (ALPHA)
C
C    NOW TAKE INTO ACCOUNT THE PROPER TRANSLATION
C
            IF (DEG .GT. 0) GO TO 20
            IF (DEG .EQ. 0) GO TO 30
            XP(K) = XP(K) - M2 * DEL * COS(ALPHA)
            YP(K) = YP(K) - M2 * DEL * SIN(ALPHA)
```

```
              GO TO 30
   20         XP(K) = XP(K) + N1 * DEL * SIN(ALPHA)
              YP(K) = YP(K) - N1 * DEL * COS(ALPHA)
   30         CONTINUE
C
C   TRUNCATE TO GET PROPER GRID SQUARE IN PRIMED SYSTEM
C
              IP(K) = INT(XP(K)/RPARM(3) + 1.)
              JP(K) = INT (YP(K)/RPARM(3) + 1.)
C
C   CHECK TO SEE IF (XP,YP) IS DEFINED IN GIVEN GRID
C
              IF ((IP(K) .LE. 0) .OR. (IP(K) .GT. IPARM(1)))
     1           GO TO 40
              IF ((JP(K) .LE. 0) .OR. (JP(K) .GT. IPARM(2)))
     1           GO TO 40
              IF (DEPTH(IP(K),JP(K)) .EQ. 0) GO TO 40
              IWATER = IWATER + 1
              SUMD = SUMD + DEPTH(IP(K),JP(K))
   40       CONTINUE
            IF (IWATER .LT. 3) GO TO 50
            NDEPTH(I,J) = INT(SUMD/IWATER + .5)
            GO TO 60
   50       NDEPTH(I,J) = 0
   60   CONTINUE
C
C SET MINIMUM DEPTH IF NOT GIVEN
C
        IF (RPARM(5) .NE. 0.) GO TO 80
        MIND = 10000
        DO 70 I = 1, MM
          DO 70 J = 1, NN
            IF (NDEPTH(I,J) .GE. MIND .OR. NDEPTH(I,J) .EQ. 0.)
     1         GO TO 70
            MIND = NDEPTH(I,J)
   70   CONTINUE
        RPARM(5) = MIND
   80   CONTINUE
C
C   TRIM EDGES TO A BORDER OF LAND ONE SQUARE WIDE
C
C   BOTTON ADJUSTMENT
C
        DO 100 JROW = 1, N
          NSUMB = 0
          DO 90 I = 1, M
   90     NSUMB = NSUMB + NDEPTH(I,JROW)
C
C CHECK IF ROW IS ALL WATER
C
```

```
      IF (NSUMB .NE. 0) GO To 110
  100 CONTINUE
C
C   CHECK IF 1ST ROW IS ALL WATER - IF SO, ADD A ROW
C
  110 IF (JROW .EQ. 1) GO TO 130
C
C   SUBRACT JROW-2 ROWS FROM BOTTOM
C
      DO 120 I = 1, M
        I1 = N - (JROW - 2)
        DO 120 J = 1, I1
  120 NDEPTH(I,J) = NDEPTH(I,J + (JROW - 2))
      JDISP = -(JROW - 2)
      GO TO 160
C
C   ADD A ROW
C
  130 DO 150 I = 1, M
        DO 140 J = 1, N
  140   NDEPTH(I,N + 2 - J) = NDEPTH(I,N + 1 - J)
  150 NDEPTH(I,1) = 0
      JDISP = 1
  160 N = N + JDISP
C
C   FURTHER ADJUST JDISP FOR TRANSLATION IF NECESSARY
C
      IF (DEG .LE. 0.) GO TO 170
      JDISP = JDISP + N1
C
C   TOP ADJUSTMENT
C
  170 JROW = N + 1
      DO 190 J = 1, N
        NSUMT = 0
        DO 180 I = 1, M
  180   NSUMT = NSUMT + NDEPTH(I,JROW - J)
C
C   CHECK IF ROW IS ALL WATER
C
        IF (NSUMT .NE. 0) Go To 200
  190 CONTINUE
C
C   CHECK IF TOP ROW IS ALL WATER
C
  200 IF (J .EQ. 1) GO TO 210
C
C   SUBTRACT J-2 ROWS BY CHANGING N
C
      N = N - J + 2
```

```
      GO TO 230
C
C     ADD A ROW
C
  210 DO 220 I = 1, M
  220 NDEPTH(I,N + 1) = 0
      N = N + 1
  230 CONTINUE
C
C     LEFT SIDE ADJUSTMENT
C
      DO 250 I = 1, M
        NSUMLT = 0
        DO 240 J = 1, N
  240   NSUMLT = NSUMLT + NDEPTH(I,J)
        ICOL = I
C
C     CHECK IF COLUMN IS ALL WATER
C
        IF (NSUHLT .NE. 0) GO TO 260
  250 CONTINUE
C
C     CHECK IF 1ST COLUMN IS ALL WATER
C
  260 IF (ICOL .EQ. 1) GO TO 280
C
C     SUBTRACT ICOL-2 COLUMNS
C
      DO 270 J = 1, N
        I3 = M - (ICOL - 2)
        DO 270 I = 1, I3
  270 NDEPTH(I,J) = NDEPTH(I + ICOL - 2,J)
      IDISP = -(ICOL - 2)
      GO TO 310
C
C     ADD A COLUMN
C
  280 DO 300 J = 1, N
        DO 290 I = 1, M
  290   NDEPTH(M + 2 - I,J) = NDEPTH(M + 1 - I,J)
  300 NDEPTH(1,J) = 0
      IDISP = 1
  310 M = M + IDISP
C
C     FURTHER ADJUST IDISP FOR TRANSLATION IF NECESSARY
C
      IF (DEG .GE. 0) GO TO 320
      IDISP = IDISP + M2
C
C     RIGHT SIDE ADJUSTMENT
```

```
C
  320 ICOL = M + 1
      DO 340 I = 1. M
        NSUMRT = 0
        DO330 J = 1, N
  330   NSUNRT = NSUNRT + NDEPTH(ICOL - I,J)
C
C    CHECK IF COLUMN IS ALL WATER
C
        IF (NSUMRT .NE. 0) GOTO350
  340 CONTINUE
C
C    CHECK IF 1ST COLUMN IS ALL WATER
C
  350 IF (I .EQ. 1) GO TO 360
C
C    SUBTRACT I-2 COLUMNS BY CHANGING M
C
      M = M - I + 2
      GO TO 380
C
C    ADD A COLUMN
C
  360 Do 370 J = 1, N
  370 NDEPTH(M + 1,J) = 0
      M = M + 1
  380 CONTINUE
C
C  MODIFY GRID AS DESIRED
C
      WRITE (6,550)
  390 READ (5.510) ICHG. JCHG. IDEP
      IF (ICHG .EQ. 0) GO TO 400
      WRITE (6,560) ICHG, JCHG, IDEP
      NDEPTH(ICHG,JCHG) = IDEP
      GO TO 390
  400 CONTINUE
C
C    FIND NAX DEPTH FOR NEW GRID
C
      MAXDEP = 0
      DO 410 I = 1, M
        DO 410 J = 1, N
          IF (NDEPTH(I,J) .LE. MAXDEP) GO TO 410
          MAXDEP = NDEPTH(I,J)
  410 CONTINUE
      RPARM(4) = FLOAT(MAXDEP)
C
C    FIND MIN DEPTH FOR NEW GRID
C
```

```
      NMIND = 10000
      DO 420 I = 1, M
        DO 420 J = 1, N
          IF (NDEPTH(I,J) .GE. NMIND .OR. NDEPTH(I,J) .EQ. 0 )
     1       GO TO 420
          NMIND = NDEPTH(I,J)
  420 CONTINUE
      RPARH(5) = FLOAT(NMIND)
C
C   NOW FIND THE NEW ARRAYS IPARM, RPARM
C
      RPARM(3) = DEL
      RPARM(7) = DEG
      IPARM(1) = M
     IPARM(2) = N
      IPARM(3) = IDISP
      IPARM(4) = JDISP
      WRITE (6,500)
      DO 430 I = 1, M
        DO 430 J = 1, N
  430 DEPTH(I,J) = FLOAT(NDEPTH(I,J))
      CALL PGRID(6, DEPTH, IDIM)
      CALL WGRID(NDEPTH, IDIM)
      GO TO 570
  440 WRITE (6,530)
      GO TO 570
  450 WRITE (6,520) DEG
      GO TO 570
  460 WRITE (6,540)
  470 FORMAT (2G20.5)
  480 FORMAT ('1', 32x, 'PROGRAM GRID'/33X,'------------'/
     1        'ODESIRED GRID SIZE (M) =', F8.0, 6X, 'DESIRED ORIENTATION'
     2        , ' (DEGREES) = ', F5.0)
  490 FORMAT ('0', 25X, 50A1/'0', 34X, 'OLD GRID'/'0', 34X, '--------')
  500 FORMAT ('1', 34X, '--------'/'0', 34X, 'NEW GRID'/'0', 34x,
     1        '--------'/'0')
  510 FORMAT (3I5)
  520 FORMAT (' ANGLE OF ROTATION (', G20.5,
     1        ') IS NOT BETWEEN -90 AND 90 - PROGRAM TERMINATED')
  530 FORMAT (' THE INPUT GRID MUST NOT HAVE AN I OR J DISPLACEMENT',
     1        ' OR A ROTATION FROM THE BASE'/' IT MUST BE AN',
     2        ' ORIGINAL BASE GRID - PROGRAM TERMINATED')
  540 FORMAT (' DESIRED GRID TOO LARGE - INCREASE IDIM AND/OR JDIM',
     1        ' IN MAIN PROGRAM - PROGRAN TERMINATED')
  550 FORMAT ('1',13X, 'GRID MODIFICATIONS'/14X, '------------------')
  560 FORMAT ('0I = ', I5, 6x. 'J = ', I5, 6X, 'NEW DEPTH = ', I5)
  570 STOP
      END
```

```
C*****************************PROGRAM FRSFC**********************************
C
C                   FREE  SURFACE  CIRCULATION  MODEL
C
C PURPOSE:
C                 THE PURPOSE OF THIS PROGRAM IS TO COMPUTE TIME-DEPEN-
C                 DENT SURFACE FLUCTUATIONS AND TRANSPORTS FOR A GRID
C                 MODEL OF A LAKE. THE LAKE IS REPRESENTED AS AN ARRAY OF
C                 SQUARE GRID BOXES AND A FINITE-DIFFERENCE FORM OF THE
C                 SHALLOW WATER EQUATIONS IS APPLIED TO THE GRID.  FREE
C                 SURFACE FLUCTUATION H(I,J) IS DEFINED AT THE CENTER
C                 OF GRID SQUARE I, J.  TRANSPORT IN THE X-DIRECTION
C                 AM(I,J) IS DEFINED AT THE CENTER OF THE RIGHT SIDE OF
C                 GRID SQUARE I, J.  TRANSPORT IN THE Y-DIRECTION AN(I,J)
C                 IS DEFINED AT THE CENTER OF THE TOP SIDE OF GRID
C                 SQUARE I, J.  THE USER IS REQUIRED TO SUPPLY A DES-
C                 SCRIPTION OF THE LAKE BATHYMETRY, THE METEOROLOGICAL
C                 FORCING CONDITIONS. A SUBROUTINE THAT SETS THE INITIAL
C                 CONDITIONS FOR FREE SURFACE FLUCTUATION AND TRANSPORT
C                 (WHICH HAY INCLUDE INFLOW AND OUTFLOW), CONTROL PARA-
C                 METERS (TIMESTEP, DURATION OF RUN. AND WATER LEVEL
C                 INCREMENT IF REQUIRED), AND A SUBROUTINE TO HANDLE
C                 OUTPUT FUNCTIONS.
C
C INPUT :
C   LOGICAL UNIT 5 :
C     CONTROL PARAMETER RECORD :
C                                                     FORMAT  CARD COLUMNS
C                 DT - TINE STEP IN SECONDS           G10.2       1 - 10
C                 TT - DURATION OF RUN IN HOURS       G10.2      11 - 20
C                 DADD - NEAN WATER LEVEL RELATIVE    G10.2      21 -30
C                        TO LOW WATER DATUM IN METERS
C
C   LOGICAL UNIT 7 :
C     BATHYMETRIC DATA FILE  :
C                 THE FORMAT OF THE BATHYMETRIC DATA FILE IS DESCRIBED
C                 IN SCHWAB AND SELLERS (1980): 'COMPUTERIZED BATHY-
C                 METRY AND SHORELINES OF THE GREAT LAKES', NOAA DATA
C                 REPORT ERL-GLERL-16.  FIVE ADDITIONAL FIELDS ARE
C                 PRESENT ON BATHYMETRIC DATA HEADER RECORD 2.  THESE
C                 ARE:
C                                                     FORMAT  CARD  COLUMNS
C                 MINIMUM DEPTH                         I5       45-49
C                 BASE GRID ROTATION FROM E-W           F6.2     50-55
C                 I DISPLACEMENT                        I5       56-60
C                 J DISPLACEMENT                        I5       61-65
C                 ROTATION  ANGLE  FROM BASE            F6.2     66-71
C                   (ANGLES MEASURED IN DEGREES COUNTERCLOCKWISE)
C
C                 THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
```

```
C                 BETWEEN LATITUDE. LONGITUDE PAIRS AN0 GRID DISTANCES.
C                 ONLY THE BATHYMETRIC PART OF THE FILE IS USED, SHORE-
C                 LINE INFORMATION NEED NOT BE INCLUDED.
C
C   LOGICAL UNIT 8 :
C     METEOROLOGICAL DATA FILE :
C                                                          FORMAT   CARD COLUMNS
C                 TLAST - TIME FROM BEGINNING              G10.4        1 - 10
C                         OF RUN (H)
C                 RLAT - LATITUDE IN DEGREES NORTH         G10.4       11 - 20
C                 RLON - LONGITUDE IN DEGREES WEST         G10.4       21 - 30
C                 Z - HEIGHT OF INSTRUMENT (M)             G10.4       31 - 40
C                 TA - TEMPERATURE OF AIR (C)              G10.4       41 - 50
C                 TW - TEMPERATURE OF WATER (C)            G10.4       51 - 60
C                 WS - WIND SPEED (M/S)                    G10.4       61 - 70
C                 WD - WIND DIRECTION (DEG)                G10.4       71 - 80
C
C          ALL DATA FOR THE SAHE TINE ARE GROUPED TOGETHER, WITH A
C          MAXIMUM OF 25 STATIONS IN A GROUP.
C   *NOTE END-OF-FILE IS INDICATED BY A RECORD WITH A NEGATIVE TIME
C
C OUTPUT :
C    LOGICAL UNIT 6 :
C      CONTROL PARAMETERS, BATHYMETRY, AND A LIST OF THE
C      METEOROLOGICAL DATA RECORDS.
C
C COMMON BLOCKS :
C                 /CPARM/ DT, TT, DADD - CONTROL PARAMETERS
C                 /GPARM/ RPARM(23), IPARM(54) - REAL AND INTEGER
C                    PARAMETERS DESCRIBING THE BATHYMETRIC GRID.
C                    SEE SUBROUTINE RGRID FOR DETAILS.
C
C  SUBROUTINES:
C        RGRID - READS THE BATHYMETRIC DATA FILE
C PGPARM ~~PGRID~~ - PRINTS GRID PARAHETERS ~~AND GRID~~
C        PRNT - FORMATS AND PRINTS DATA FROM GRID
C        UZL - CALCULATES DRAG COEFFICIENT FOR METEOROLOGICAL DATA
C        FUNCTION TAU - READS METEOROLOGICAL DATA AND CALCULATES
C                       WIND STRESS
C        FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN
C                         GIVEN LATITUDE AND LONGITUDE
C        FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN
C                         GIVEN LATITUDE AND LONGITUDE
C        FUNCTION RLAT - RETURNS LATITUDE GIVEN X AND Y DISTANCE
C                        FROM GRID ORIGIN
C
C    THE USER MUST SUPPLY TWO SUBROUTINES TO HANDLE INITIAL
C    CONDITIONS AND OUTPUT. THEY ARE:
C
C        INIT(D, H, AH. AN, IDIM) - SET INITIAL CONDITIONS FOR FREE
```

```
C          SURFACE FLUCTUATION FIELD (H) AND TRANSPORT FIELD (AM, AN).
C          D IS THE DEPTH ARRAY AND IDIM IS THE FIRST DIMENSION OF D,
C          H, AM, AND AN.
C
C        OUTP(TIME, D, H, AM, AN, IDIM) - GENERATES USER-REQUIRED
C          OUTPUT. OUTP IS CALLED BY FRSFC EACH TIMESTEP WITH THE
C          CURRENT TIME (IN SECONDS). H, AM, AND AN FIELDS. D IS
C          THE DEPTH ARRAY AND IDIM IS THE FIRST DIMENSION OF D,
C          H, AM, AND AN.
C
C HISTORY : WRITTEN BY D. J. SCHWAB, 1981, GLERL, ANN ARBOR, MI
C
      REAL D(54,17), H(54,17),AM(54,17), AN(54,17)
      COMMON /CPARM/ DT, TT, DADD
      COMMON /GPARM/ RPARM(23), IPARM(54)
C
C IDIM AND JDIM ARE THE FIRST AND SECOND DIMENSIONS OF THE
C ARRAYS D, H, AM, AND AN
c
      DATA IDIM, JDIM /54, 17/
C
C LUNB AND LUNM ARE THE LOGICAL UNIT NUMBERS FOR THE BATHYMETRIC
C DATA FILE AND THE METEOROLOGICAL DATA FILE RESPECTIVELY
C
      DATA LUNB, LUNM /7, 8/
C
C PHYSICAL CONSTANTS :
C     G - GRAVITATIONAL ACCELERATION (M/S**2)
C     ON - ANGULAR SPEED OF ROTATION OF THE EARTH (RAD/S)
C     FR - FRICTIONAL DRAG COEFFICIENT FOR BOTTON STRESS
C
      DATA G. ON, FR /9.8, 7.29E-5, 2.E-3/
C
C READ BATHYMETRIC GRID INFORMATION
C
      CALL RGRID(LUNB, D, IDIM, JDIM)
C
C READ CONTROL PARAMETERS
C
      READ (5,70) DT, TT, DADD
      NSTEPS = TT * 3600. / DT
      IN = IPARM(1)
      JM = IPARM(2)
      DS = RPARM(3)
      DMAX = RPARM(4)
      DMIN = RPARM(5) + DADD
C
C CALCULATE MAXIMUM TIME STEP FOR LONG GRAVITY WAVE
C
      DTSTAB = DS / SQRT(2.*G*(DMAX + DADD))
```

```
C
C   CALCULATE  CORIOLIS  PARAMETER  AT  CENTER  OF  GRID
C
      F = 2. * OH * SIN(RLAT(IM*DS/2.,JM*DS/2.)*ATAN(1.)/45.)
      WRITE (6,80) (IPARM(I), I=5,54), DT, TT, DADD, NSTEPS, F, DTSTAB
      IMM1 = IM - 1
      JMM1 = JM - 1
      DTODS = DT / DS
      FRDT = FR * DT
      FDT = F * DT
      GDTODS = G * DT / DS
C
C   CLEAR  ALL  ARRAYS  AND  ADD  WATER  LEVEL  INCREMENT
C
      DO 10 I = 1, IN
        DO 10 J = 1, JM
          H(I,J) = 0.
          AM(I,J) = 0.                  RPARM(5)
          AN(I,J) = 0.
          IF (D(I,J) .LT. DMIN) GO TO 10
          D(I,J) = D(I,J) + DADD
   10 CONTINUE
      ~~DMIN = DMIN + DADD~~
      IF (DMIN .LE. 0) GO TO 60
C                                      C
C   PRINT  BATHYMETRIC  NFORMATION     C PRINT  DEPTH  FIELD
C            PGPARM(
      CALL  ~~PGRID(6 )~~              C
C
C   GET  INITIAL  CONDITIONS                     WRITE(6,85)
C
      CALL INIT(D, H, AM, AN, IDIM)             CALL PRNT(6,D,IDIM,IM,
C                                                           JM,C.
C   MAIN  ITERATION  LOOP
C
      DO 50 N = 1, NSTEPS
        TIME = N * DT
C
C   ADVANCE  FREE  SURFACE  FLUCTUATION  FIELD  ONE  TIMESTEP
C
        DO 20 I = 2, IMM1
          DO 20 J = 2, JMM1
            IF (D(I,J) .LT. DMIN) GO TO 20
            H(I,J) = H(I,J) - DTODS * (AM(I,J) - AM(I - 1,J) + AN(I,J) -
     1        AN(I,J - 1))
   20   CONTINUE
C
C   ADVANCE  X-COMPONENT  OF  TRANSPORT  ONE  TIMESTEP
C
        DO 30 I = 2, IMM1
```

```
          DO 30 J = 2, JMM1
             IF (D(I,J) .LT. DMIN .OR. D(I + 1,J) .LT. DMIN)
     1          GO TO 30
C
C   PLATZMAN'S CORIOLIS FORCE AVERAGING SCHEME
C
             DM = 2. / (D(I,J) + D(I + 1,J))
             DNW = 2. / (D(I,J) + D(I,J + 1))
             DSW = 2. / (D(I,J) + D(I,J - 1))
             DNE = 2. / (D(I + 1,J) + D(I + 1,J + 1))
             DSE = 2. / (D(I + 1,J) + D(I + 1,J - 1))
             ANAVG = ((DM + DNW)*AN(I,J) + (DM + DSW)*AN(I,J - 1) + (DM +
     1        DNE)*AN(I + 1,J) + (DM + DSE)*AN(I + 1,J - 1)) /(8.*DM)
             SPD = DM * SQRT(AM(I,J)**2 + ANAVG**2)
C
C   X-COMPONENT OF TRANSPORT
C
             AM(I,J) = (1. - FRDT*SPD*DM) * AM(I,J) + FDT * ANAVG -
     1       GDTODS * (H(I + 1,J) - H(I,J)) / DM + DT * TAU(TIME,I,J,1)
   30    CONTINUE
C
C   ADVANCE Y-COMPONENT OF TRANSPORT ONE TIMESTEP
C
         DO 40 I = 2, IMM1
          DO 40 J = 2. JMM1
             IF (D(I,J) .LT. DMIN .OR. D(I,J + 1) .LT. DMIN)
     1          GO TO 40
C
C   PLATZHAN'S CORIOLIS FORCE AVERAGING SCHEME
C
             DM = 2. / (D(I,J) + D(I,J + 1))
             DNW = 2. / (D(I - 1,J + 1) + D(I,J + 1))
             DSW = 2. / (D(I - 1,J) + D(I,J))
             DNE = 2. / (D(I,J + 1) + D(I + 1,J + 1))
             DSE = 2. / (D(I,J) + D(I + 1,J))
             AMAVG = ((DM + DNW)*AM(I - 1,J + 1) + (DM + DSW)*AM(I - 1,J)
     1       +(DM + DNE)*AM(I,J + 1) + (DM + DSE)*AM(I,J)) / (8.*DM)
             SPD = DM * SQRT(AN(I,J)**2 + AMAVG**2)
C
C   Y-COMPONENT OF TRANSPORT
C
             AN (I,J) = (1. - FRDT*SPD*DM) * AN(I,J) - FDT * AMAVG -
     1       GDTODS * (H(I,J + 1) - H(I,J)) / DM + DT * TAU(TIME,I,J,2)
   40    CONTINUE
C
C   CALL OUTPUT ROUTINE
C
         CALL OUTP(TIME, D, H, AN, AN, IDIM)
C
C   END MAIN ITERATION LOOP
```

```
C
   50  CONTINUE
       GO TO 100
   60  WRITE (6,90) DADD
   70  FORMAT (3G10.2)
   80  FORMAT ('1FREE SURFACE CIRCULATDN MODEL FOR ', 50A1/
     1        ' TIME STEP(S): DT= ', F6.2/ ' DURATION OF RUN(H): TT= ',
     2        F7.2/' MEAN WATER LEVEL(H), (RELATIVE TO L.W.D) : DADD= ',
     3        F5.2/' NUMBER OF TIME STEPS: NSTEPS= ', I6/
     4        ' CORIOLIS PARAMETER (S**-1): F=', E10.3/
     5        ' MAXIMUM TIMESTEP FOR LONG GRAVITY WAVE (S): DTSTAB=',
     6        F10.2)
   90  FORMAT (' THE WATER LEVEL INCREMENT', F8.2,
     1        ' RESULTS IN A NEGATIVE'.
     2        ' MINIMUM DEPTH - PROGRAM TERMINATED')
  100  STOP
       END

   85  FORMAT(1X,'DEPTH RELATIVE TO MEAN WATER
                                          LEVEL')
```

```
C****************************PROGRAM RLID**********************************
C
C                    RIGID LID CIRCULATION MODEL
C
C PURPOSE:
C              THE PURPOSE OF THIS PRDGRAH IS TO COMPUTE THE TIME-DEP-
C              ENDENT CIRCULATION FOR A GRID MODEL OF A LAKE. THE
C              CIRCULATION IS ASSUMED TO BE NON-DIVERGENT SO THAT IT
C              CAN BE REPRESENTED IN TERNS OF A STREAM FUNCTION. THE
C              LAKE IS REPRESENTED AS AN ARRAY OF SQUARE GRID BOXES
C              AND A FINITE DIFFERENCE FORM OF THE VORTICITY
C              EQUATION IS APPLIED TO THE GRID. THE STREAH FUNCTION
C              S(I,J) IS DEFINED AT THE TOP RIGHT CORNER OF GRID SQUARE
C              I, J. THE USER IS REQUIRED TO SUPPLY A DESCRIPTION OF
C              THE LAKE BATHYHETRY, THE METEOROLOGICAL FORCING CONDIT-
C              IONS, A SUBROUTINE THAT SETS THE INITIAL CONDITION FOR
C              THE STREAM FUNCTION FIELD (WHICH NAY INCLUDE INFLOW AND
C              AND OUTFLOW CONDITIONS), CONTROL PARAMETERS (TIMESTEP,
C              DURATION OF RUN, AND WATER LEVEL INCREMENT IF RE-
C              QUIRED), AND A SUBROUTINE TO HANDLE OUTPUT FUNCTIONS.
C INPUT :
C   LOGICAL UNIT 5 :
C     CONTROL PARAMETER RECORD :
C                                                   FORMAT  CARD  COLUMNS
C               DT - TIME STEP IN HOURS             G10.2       1 - 10
C               TT - DURATION OF RUN IN HOURS       G10.2      11 - 20
C               DADD - MEAN WATER LEVEL RELATIVE    G10.2      21 - 30
C                      TO LOW WATER DATUM IN METERS
C
C   LOGICAL UNIT 7 :
C     BATHYMETRIC DATA FILE :
C               THE FORMAT OF THE BATHYMETRIC DATA FILE IS DESCRIBED
C               IN SCHWAB AND SELLERS (1980):'COMPUTERIZED BATHY-
C               METRY AND SHORELINES OF THE GREAT LAKES', NOAA DATA
C               REPORT ERL-GLERL-16. FIVE ADDITIONAL FIELDS ARE
C               PRESENT ON BATHYMETRIC DATA HEADER RECORD 2. THESE
C               ARE:
C                                                   FORMAT  CARD  COLUMNS
C               MINIMUM DEPTH                         I5      45-49
C               BASE GRID ROTATION FROM E-W           F6.2    50-55
C               I DISPLACEMENT                        I5      56-60
C               J DISPLACEMENT                        I5      61-65
C               ROTATION ANGLE FROM BASE              ~6.2    66-71
C                 (ANGLES MEASURED IN DEGREES COUNTER CLOCKWISE)
C
C               THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
C               BETWEEN LATITUDE, LONGITUDE PAIRS AND GRID DISTANCES.
C               ONLY THE BATHYMETRIC PART OF THE FILE IS USED. SHORE-
C               LINE INFORMATION NEED NOT BE INCLUDED.
C
```

```
C    LOGICAL  UNIT  8 :
C      HETEOROLOGICAL  DATA  FILE  :
C                                                                FORMAT   CARD  COLUMNS
C                   TLAST  -  TINE  FROM  BEGINNING              G10.4          1  -  10
C                            OF  RUN  (H)
C                   RLAT  -  LATITUDE  IN  DEGREES  NORTH        G10.4         11  -  20
C                   RLON -  LONGITUDE  IN  DEGREES  WEST         G10.4         21  -  30
C                   Z -  HEIGHT  OF  INSTRUMENT  (M)             G10.4         31  -  40
C                   TA  -  TEMPERATURE  OF  AIR  (C)             610.4         41  -  50
C                   TW  -  TEMPERATURE  OF  WATER  (C)           G10.4         51  -  60
C                   WS -  WIND  SPEED  (M/S)                     G10.4         61  -  70
C                   WD  -  WIND  DIRECTION  (DEG)                G10.4         71  -  80
C
C          ALL  DATA  FOR  THE  SAME  TINE  ARE  GROUPED  TOGETHER,  WITH  A
C          MAXIMUM OF 25  STATIONS  IN  A  GROUP.
C
C    *NOTE  END-OF-FILE  IS  INDICATED  BY  A  RECORD  WITH  A  NEGATIVE  TIME
C
C  OUTPUT  :
C     LOGICAL  UNIT  6  :
C        CONTROL  PARAMETERS,  BATHYNETRY,  AND  A  LIST  OF  THE
C        HETEOROLOGICAL  DATA  RECORDS.
C
C  COMMON  BLOCKS  :
C                   /CPARM/ DT, TT. DADD - CONTROL PARAMETERS
C                   /GPARM/ RPARM(23), IPARM(54) - REAL  AND  INTEGER
C                      PARANETERS  DESCRIBING  THE  BATHYNETRIC  GRID.
C                      SEE  SUBROUTINE  RGRID  FOR  DETAILS.
C                   /ITPARM/  DSTNAX,  STMIN,  STMAX,  FRAC,  ITS
C                      RELAXATION  PARAMETERS:
C                           DSTNAX  - MAXIMUM  CHANGE  IN  ABSOLUTE  VALUE  OF
C                                     STREAM  FUNCTION  AT  LAST  ITERATION
C                           STMIN -  MINIMUM  VALUE  OF  STREAN  FUNCTION
C                           STNAX  - MAXIMUM  VALUE  OF  STREAM  FUNCTION
C                           FRAC  - DSTMAX / (STMAX - STMIN)
C                           ITS  -  NUMBER  OF  ITERATIONS  TAKEN  TO  CONVERGE
C
C  SUBROUTINES:
C        RGRID -  READS  THE  BATHYNETRIC  DATA  FILE
C  DGPARM ~~PGRID~~ -  PRINTS  GRID  PARAHETERS  ~~AND GRID~~
C        PRNT  -  FORMATS  AND  PRINTS  DATA  FROM  GRID
C        UZL  -  CALCULATES  DRAG  COEFFICIENT  FOR  HETEOROLOGICAL  DATA
C        FUNCTION  TAU  -  READS  METEOROLOGICAL  DATA  AND  CALCULATES
C                          WIND  STRESS
C        FUNCTION  XDIST -  RETURNS  X  DISTANCE  FROM  GRID  ORIGIN
C                           GIVEN  LATITUDE  AND  LONGITUDE
C        FUNCTION  YDIST -  RETURNS  Y  DISTANCE  FROM  GRID  ORIGIN
C                           GIVEN  LATITUDE  AND LONGITUDE
C        FUNCTION  RLAT  -  RETURNS  LATITUDE  GIVEN  X  AND  Y  DISTANCE
C                           FRON  GRID  ORIGIN
```

```
C
C     THE USER MUST SUPPLY TWO SUBROUTINES TO HANDLE INITIAL
C     CONDITIONS AND OUTPUT. THEY ARE:
C
C        INIT(D, S, IDIM) - SETS INITIAL CONDITION FOR STREAH FUNCTION
C          FIELD (S). D IS THE DEPTH ARRAY (INTERPOLATED TO STREAM
C          FUNCTION POINTS) AND IDIM IS THE FIRST DIMENSION OF D AND S.
C
C        OUTP(TIME, D, S, SPD, IDIM) - GENERATES USER-REQUIRED OUTPUT.
C          OUTP IS CALLED BY RLID EACH TIMESTEP WITH THE CURRENT TIME
C          (TINE IN SECONDS), THE STREAM FUNCTION FIELD (S), AND THE
C          CURRENT SPEED FIELD (SPD). D IS THE DEPTH ARRAY (INTER-
C          POLATED TO STREAM FUNCTION POINTS) AND IDIM IS THE FIRST
C          DIMENSION OF D, S, AND SPD.
C
C HISTORY: WRITTEN BY J. R. BENNETT AND D. J. SCHWAB, 1981,
C          GLERL, ANN ARBOR, MI
C
      DIMENSION D(42,42), S(42,42), RHS(42,42), SPD(42,42)
      COMMON /CPARM/ DT, TT, DADD
      COMMON /GPARM/ RPARM(23), IPARM(54)
      COMMON /ITPARM/ DSTHAX, STMIN, STMAX, FRAC, ITS
C
C IDIM AND JDIM ARE THE FIRST AND SECOND DIMENSIONS OF THE ARRAYS
C D, S, RHS, AND SPD
C
      DATA IDIM, JDIM /42, 42/
C
C  LUNB AND LUNH ARE THE LOGICAL UNIT NUMBERS FOR THE BATHYMETRIC DATA
C  FILE AND THE METEOROLOGICAL DATA FILE RESPECTIVELY
C
      DATA LUNB, LUNM /7, 8/
C
C  PHYSICAL CONSTANTS:
C    OM - ANGULAR SPEED OF ROTATION OF THE EARTH (RAD / S)
C    FR - FRICTIONAL DRAG COEFFICIENT FOR BOTTOM STRESS
C
      DATA OM, FR /7.29E-5, 2.E-3/
C
C  CONTROL PARMETERS FOR STREAM FUNCTION RELAXATION SCHEME
C    RELAX - OVERRELAXATION FACTOR FOR ITERATIVE SOLUTION OF STREAM
C            FUNCTION EQUATION AT EACH TIMESTEP
C    ITMAX - MAXIMUM NUMBER OF ITERATIONS FOR RELAXATION SCHEME AT
C            EACH TIMESTEP
C    CONV - RELATIVE CONVERGENCE CRITERION FOR RELAXATION SCHEME
C
      DATA RELAX, ITMAX, CONV /1.65, 50, 1.E-3/
      PI = ATAN(1.) * 4.
C
C  READ BATHYMETRIC GRID INFORMATION
```

```
C
      CALL  RGRID(LUNB, D, IDIM, JDIM)
C
C  READ  CONTROL  PARAMETERS
C
      READ  (5,100)  DT,  TT.  DADD
      NSTEPS  =  TT  /  DT
      IM = IPARM(1)
      JN  = IPARM(2)
      DS  = RPARM(3)
      DNAX  =  RPARM(4)
      DMIN  =  RPARM(5)+DADD
C
C  CALCULATE  CORIOLIS  PARANETER  AT  CENTER  OF  GRID
C
      F  =  2.  *  ON  *  SIN(RLAT(IM*DS/2.,JM*DS/2.)*PI/180.)
      WRITE  (6,110)  (IPARM(I),I=5,54),  DT,  TT.  DADD,  NSTEPS,  F
      IMM1 = IM  -  1
      JMM1 = JM  -  1
      IMM2  =  IN  -  2
      JMM2  =  JN  -  2
      DT  =  DT  *  3600.
      FoT24  =  F  *  DT  /  24.
C
C        ADJUST  RELAXATION  FACTOR  FOR  GRID  SIZE
C
      RELAX  =  RELAX  /  (1.  +  SIN(ACOS(0.5*(COS(PI/IMM2)  +  COS(PI/JMM2)))
     1))
C
C  INTERPOLATE  DEPTH  TO  STREAM  FUNCTION  POINTS
C  USING  SPEED  ARRAY  FOR  TENPORARY  STORAGE
C
      DO  10  I  =  1,  IM
        DO  10  J  =  1,  JM
          SPD(I,J)  =  0.0          RPARM(5)
          S(I,J)  =  0.
          IF  (D(I,J)  .LT.  DMIN)  GO  TO  10
          D(I,J) = D (I,J)  + DADD
   10 RHS(I,J)  =  0.
      DMIN  =  DMIN  +  DADD
C
C  IF  WATER  LEVEL  INCREMENT  RESULTS  IN  A  NEGATIVE  DMIN,  STOP.
C
      IF  (DMIN  .LE.  0.0)  GO  TO  90
      DO  20  I  =  1.  IM
        DO 20 J = 1, JM                 0.99999
          SPD(I,J)  = 0.9 * DMIN
          IF  (I  .EQ.  IM  .OR.  J  .EQ.  JM)  GO TO 20
          IF  (D(I,J)  .LT.  DMIN)  GO  TO  2 0
          IF  (D(I + 1,J)  .LT.  DMIN)  GO  TO  20
```

```
          IF (D(I,J + 1) .LT. DMIN) GO TO 20
          IF (D(I + 1.J + 1) .LT. DMIN) GO TO 20
          SPD(I,J) = 0.25 * (D(I,J) + D(I + 1,J) + D(I,J + 1) + D(I + 1,
     1    J + 1))
   20 CONTINUE
C
C PRINT BATHYMETRIC GRID INFORMATION
C                PGPARM (6)                     C PRINT THE DEPTH FIELD
      CALL PGRID(6, SPD, IDIM)                  C
C                                                  WRITE(6,125)
C STORE INVERSE DEPTH BACK IN D                    CALL PRNT(6,D,IDIM,IDM,JM,0.)
C
      DMINI = 1. / DMIN
      DO 30 I = 1, IM
        DO 30 J = 1. JH
          D(I,J) = 1. / SPD(I,J)
   30 SPD(I,J) = 0.0
C
C GET INITIAL CONDITIONS
C
      CALL INIT(D, S, SPD, IDIM)
C
C MAIN ITERATION LOOP                 TIME = 0.
C
      DO 80 N = 1, NSTEPS
        TIME = N * DT TIME + DT/2.
C
C CALCULATE CURRENT SPEED AT CENTER OF GRID BOX I, J
C
        DO 40 I = 2, IMM1
          DO 40 J = 2, JMM1
            DU = 0.5 * (D(I,J) + D(I,J - 1))
            DV = 0.5 * (D(I,J) + D(I - 1,J))
            DUN = 0.5 * (D(I - 1,J) + D(I - 1,J - 1))
            DVM = 0.5 * (D(I,J - 1) + D(I - 1,J - 1))
            SPD(I,J) = (0.5/DS) * SQRT(((((S(I,J) - S(I,J - 1))*DU) + ((
     1      S(I - 1,J) - S(I - 1,J - 1))*DUM))**2 + (((S(I,J) - S(I - 1.
     2      J))*DV) + ((S(I,J - 1) - S(I - 1,J - 1))*DVM))**2))
   40 CONTINUE
C
C ITERATE TO CALCULATE STREAM FUNCTION AT NEXT TIME STEP WITH ALTER-
C NATING SWEEP DIRECTIONS
C
        DO 60 K = 1, ITMAX
          DSTMAX = 0.             KK = K+N
          STMIN = 0.
          STMAX = 0.
Leave in  ITS = K
          DO 50 II = 1, IN
            I = II
```

KK

```
          IF  (MOD(K,2)  .EQ.  0)  I = IM - II + 1
          DO 50 JJ = 1, JM
            J = JJ
            IF  (MOD(K,2)  .EQ.  0)  J = JM - JJ + 1
            IF  (D(I,J)  .GT.  DMINI)  GO  TO  50
            DUP  =  0.5  * (D(I,J + 1)  +  D(I,J))
            DVP  =  0.5  * (D(I + 1.J)  +  D(I,J))
            SPDUP  =  0.5  * (SPD(I + 1,J + 1)  +  SPD(I,J + 1))
            SPDVP  =  0.5  * (SPD(I + 1,J + 1)  +  SPD(I + 1,J))
            DU  =  0.5  * (D(I,J)  +  D(I,J - 1))
            DV  =  0.5  * (D(I,J)  +  D(I - 1,J))
            SPDU  =  0.5  * (SPD(I + 1,J)  +  SPD(I,J))
            SPDV  =  0.5  * (SPI)(I,J + 1)  +  SPD(I,J))
            DCENT  =  DVP  +  DV  +  DUP  +  DU
C
C LAPLACIAN  TERN
C
            TERM1  =  DVP  * S(I + 1,J)  +  DV  * S(I - 1,J)  +  DUP  * S(I,J +
     1       1)  +  DU * S(I,J - 1)  -  DCENT  * S(I,J)
C
C  ARAKAWA'S  JACOBIAN
C
            TERM2 = S(I + 1,J) * (D(I,J + 1) + D(I + 1,J + 1) - D(I,J
     1      -1)-D(I+1,J-  1))  +  S(I - 1,J) * (-D(I,J + 1) - D(I -
     2       1,J+1)+D(I,J -  1)  +  D(I - 1,J - 1)) + S(I,J + 1) * (-
     3      D(I+1,J)- D(I + 1,J + 1) + D(I - 1,J) + D(I - 1,J + 1))
     4      +S(I,J-1)* (D(I + 1,J) + D(I + 1,J - 1) - D(I -1,J)-
     5      D(I-1,J-1))+S(I + 1,J + 1) * (-D(I + 1,J) + D(I,J+
     6      1)) + S(I + 1,J - 1) * (D(I + 1,J) - D(I,J - 1)) + S(I -
     7       1,J + 1) * (D(I - 1,J) - D(I,J + 1)) + S(I - 1,J - 1) * (-
     8      D(I - 1,J) + D(I,J -  1))
            TYP = -DVP * (S(I + 1,J) - S(I,J))  *  FR  *  SPDVP
            TYM =  -DV  * (S(I,J) - S(I - 1,J)) *  FR  * SPDV
            TXP = DUP * (S(I,J + 1)  - S(I,J)) *  FR  *  SPDUP
            TXM = DU * (S(I,J) - S(I,J - 1))  *  FR * SPDU
C  FRICTION
C  FORCING TERM
C
            TERM3 = (DVP*TYP - DV*TYM - DUP*TXP + DU*TXM)
C
C  SET  RIGHT  HAND  SIDE  THE  FIRST  TIME  THROUGH
C
            IF  (K  .EQ.  1)  RHS(I,J) = TERM1 - FDT24 * TERN2  +  DT  *  0.5
     1      * TERM3 + DT * DS  * (DVP*TAU(TIME,I + 1,J,2) - DV*TAU(
     2      TIME,I,J,2) - DUP*TAU(TIME,I,J + 1.1)  +  DU*TAU(TIME,I,J,1)
     3      )
            IF  (K  .EQ.  1)  GO  TO  50
C
C  CALCULATE  NEW  STREAM  FUNCTION
```

```
            D4 = DCENT + FR *0.5 * DT * (DVP**2*SPDVP + DV**2*SPDV +
     1      DUP**2*SPDUP + DU**2*SPDU)
            DST = (TERN1 + FDT24*TERM2 - 0.5*DT*TERM3 - RHS(I,J)) / D4
            S(I,J) = S(I,J) + RELAX * DST
            DSTMAX = AMAX1(DSTMAX,ABS(DST))
            STMIN = AMIN1(STMIN,S(I,J))
            STMAX = AMAX1(STMAX,S(I,J))
   50     CONTINUE
C
C  CALCULATE RELATIVE CHANGE IN STREAM FUNCTION FOR ALL ITERATIONS BUT
C  THE FIRST
C                      OK
          ~~IF (K .EQ. 1) GO TO 60~~ <---- IF ( STMAX.EQ.STMIN ) GO TO 70
          FRAC = DSTMAX / (STMAX - STMIN)
          IF (FRAC .LE. CONV) GO TO 70
   60   CONTINUE
   70   CONTINUE <----------   & UPDATE TIME
                                 TIME = N*DT
C
C  CALL OUTPUT ROUTINE
C
        CALL OUTP(TIME, D, S, SPD. IDIM)
C
C END MAIN ITERATION LOOP
C
   80  CONTINUE
      GO TO 130
   90 WRITE (6,120) DADD
  100 FORMAT (3G10.2)
  110 FORMAT ('1RIGID LID CIRCULATION MODEL FOR ',
     1       50A1/' TINE STEP(H): DT= ', F10.2/
     1       ' DURATION OF RUN(H): TT= ', F7.2/
     2       ' MEAN WATER LEVEL(M) (RELATIVE TO L.W.D.): DADD= ', F5.2/
     3       ' NUMBER OF TINE STEPS: NSTEPS=', I6/
     4       ' CORIOLIS PARAMETER (S**-1): F= ', E10.3)
  120 FORMAT (' THE WATER LEVEL INCREMENT', F8.2, ' RESULTS IN A',
     1       ' NEGATIVE MINUMUM DEPTH - PROGRAN TERMINATED')
  130 STOP
      END

 125 FORMAT (1X,' DEPTH RELATIVE TO MEAN WATER LEVEL')
```

```
      SUBROUTINE WGRID(ND,IDIM)
C  PURPOSE:
C                   TO WRITE A BATHYMETRIC GRID DATA FILE
C
C  ARGUMENTS:
C                   ND - INTEGER ARRAY CONTAINING GRID DEPTHS
C                   IDIM - FIRST DIMENSION OF NO IN DIMENSION STATEMENT
C                          OF CALLING PROGRAM
C
C  COMMON BLOCKS:
C                   GPARM/RPARM(23), IPARN(54)
C
      DIMENSION ND(IDIM,1), NRPARM(6)
      COMMON /GPARM/ RPARM(23), IPARM(54)
      DO 10 K = 3, 6
   10 NRPARM(K) = INT(RPARM(K))
      WRITE (8,20) (IPARM(I),I=5,54), IPARM(1), IPARM(2), RPARM(1),
     1RPARM(2), (NRPARM(I),I=3,5), RPARM(6), IPARM(3), IPARM(4),
     2RPARM(7), (RPARM(I),I=8,23)
      IN = IPARM(1)
      JM = IPARM(2)
      WRITE (8,30) ((ND(I,J),I=1,IM),J=1,JM)
   20 FORMAT (50(A1)/2I5, 2F12.7, 3I5, F6.2, 2I5, F6.2/
     1         3(4E15.6/), 4E15.6)
   30 FORMAT (19I4, 4X)
   40 RETURN
      END
```

```
      SUBROUTINE  RGRID(LUN, D, IDIM, JDIM)
C  PURPOSE:
C                        TO READ A STANDARD BATHYMETRIC GRID DATA FILE
C                        AND RETURN GRID PARAMETERS AND DEPTHS.
C  ARGUHENTS:
C  ON  INPUT:
C                        LUN - LOGICAL UNIT NUMBER OF BATHYMETRIC DATA FILE
C                        IDIM - FIRST DIMENSION OF ARRAY D IN
C                               DIMENSION STATEMENT OF CALLING PROGRAM
C                        JOIN - SECOND DIMENSION OF ARRAY D IN
C                               DIMENSION STATEHENT OF CALLING PROGRAM
C  ON  OUTPUT:
C                         D - DEPTH ARRAY. ZERO FOR LAND, AVERAGE DEPTH
C                             OF GRID BOX IN NETERS FOR WATER.
C                         RPARN - ARRAY CONTAINING REAL-VALUED BATHYMETRIC
C                                 GRID PARAMETERS AS FOLLOWS:
C                              1.  BASE LATITUDE
C                              2.  BASE LONGITUDE
C                              3.  GRID SIZE (M)
C                              4.  MAXIMUM DEPTH (M)
C                              5.  MINIMUM DEPTH (M)
C                              6.  BASE ROTATION (COUNTERCLOCKWISE FROM E-W)
C                              7.  ROTATION FRON BASE (COUNTERCLOCKWISE)
C                              8-11.  GEOGRAPHIC-TO-NAP COORDINATE CONVERSION
C                                     COEFFICIENTS FOR X
C                              12-15.  GEOGRAPHIC-TO-NAP COORDINATE CONVERSION
C                                      COEFFICIENTS FOR Y
C                         16-15-19.  NAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                                      COEFFICIENTS FOR LONGITUDE
C                              20-23.  HAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                                      COEFFICIENTS FOR LATITUDE
C                        IPARM - ARRAY CONTAINING INTEGER-VALUED BATHYMETRIC
C                                GRID PARAMETERS AS FOLLOWS:
C                              1.  NUNBER OF GRID BOXES IN X DIRECTION
C                              2.  NUMBER OF GRID BOXES IN Y DIRECION
C                              3.  I DISPLACEMENT - THE NUMBER OF NEW GRID
C                                  SQUARES IN THE X-DIRECTION FROM THE NEW
C                                  GRID ORIGIN TO THE OLD GRID ORIGIN
C                              4.  J DISPLACEMENT - THE NUMBER OF NEW GRID
C                                  SQUARES IN THE Y-DIRECTION FROM THE NEW
C                                  GRID ORIGIN TO THE OLD GRID ORIGIN
C                              5-54.  LAKE NAME (50A1)
C                         NOTE : IF GRID IS TOO LARGE FOR DIMENSIONS OF D,
C                                THE IPARM ARRAY IS SET TO ZERO
C
C  COMMON  BLOCK:
C                        /GPARM/RPARM(23), IPARM(54)
C
      DIMENSION D(IDIM,JDIM)
      COMMON /GPARM/ RPARM(23), IPARM(54)
```

```
      READ (LUN,30) (IPARM(I),I=5,54), IPARM(1), IPARM(2),
     1(RPARM(I),I=1,6), IPARM(3), IPARM(4), RPARM(7)
      READ (LUN,60) (RPARM(I) ,I=8,23)
      IM = IPARM(1)
      JM = IPARM(2)
      IF (IPARM(1) .GT. IDIM .OR. IPARM(2) .GT. JDIM) GO TO 10
      READ (LUN,40) ((D(I,J),I=1,IM) ,J=1,JM)
      RETURN
10    DO 20 I = 1, 54
20    IPARM(I) = 0
      WRITE (6,50)
30    FORMAT (50(A1)/2I5, 2~12.7, 3F5.0, F6.2, 215, F6.2)
40    FORMAT (19F4.0, 4X)
50    FORMAT (' BATHYMETRIC GRID TOO LARGE - INCREASE DIMENSIONS OF',
     1       ' NDEPTH AND DEPTH IN MAIN PROGRAM')
60    FORMAT (4E15.6, 20X)
70    RETURN
      END
```

```
      SUBROUTINE PGRID(LUN, D, IDIM)
C  PURPOSE:
C                   TO PRINT THE GRID DESCRIPTION PARAMETERS (RPARM
C                   AND IPARM IN COMMON BLOCK /GPARM/) AND THE
C                   BATHYMETRIC GRID
C  ARGUHENTS:
C                   LUN - LOGICAL UNIT NUMBER ON WHICH TO PRINT
C                   D - BATHYMETRIC GRID
C                   IDIM - FIRST DIMENSION OF D IN DIMENSION STATERENT
C                          OF CALLING PROGRAM
C COMMON BLOCK:
C                   /GPARM/ RPARM(23),IPARM(54)
C
      COMMON /GPARM/ RPARM(23), IPARM(54)
      DIMENSION D(IDIM,1)
      WRITE (LUN,10) (IPARM(I),I=5,54)
      WRITE (LUN,20) (IPARM(I), =1,4)
      WRITE (LUN,30) (RPARM(I),I=1,7)
      WRITE (LUN,40) (RPARM(I),I=8,23)
      CALL PRNT(LUN, D, IDIM, IPARM(1), IPARM(2), O.)
   10 FORMAT ('0'. 10X, 'GRID DESCRIPTION FOR ', 50A1)
   20 FORMAT ('OIDIMENSION : IPARM(1) = ', I5, 6X, 'JDIMENSION : ',
     1       'IPARM(2) = ', 8X, I5/'0', 14X,
     2       'DISPLACEMENT OF ORIGIN IN ', 'NUMBER OF GRID SQUARES'/
     3       'OIDISPLACEMENT : IPARM(3) =', I5, 6X.
     4       'JDISPLACEHENT : IPARM(4) =', 6X. I5)
   30 FORMAT ('OBASE LATITUDE : RPARM(1) = ', F12.7/'OBASE LONGITUDE ',
     1       ': RPARM(2) =', F12.7/'OGRID SIZE (M) : RPARM(3) = ', F6.0,
     2       5X, 'MAX DEPTH (M) : RPARM(4) = ', 6X, F5.0/
     3       'OMIN DEPTH (M) : ', 'RPARM(5) = ', F5.0, 5X,
     4       'BASE ROTATION : RPARM(6) = ', 8X, F5.2/
     5       'OANGLE ROTATED : RPARM(7) = ', F5.0)
   40 FORMAT ('0', 11X, 'GEOGRAPHIC-TO-NAP COORDINATE CONVERSION ',
     1       'COEFFICENTS FOR X'/'ORPARM(8) = ', 6X, E15.6, 5X.
     2       'RPARM(9) = ', 12x, E15.6/'ORPARM(10) = ', 5X, E15.6, 5X,
     3       'RPARM(11) = ', 11X, E15.6/'0', 11X,
     4       'GEOGRAPHIC-TO-HAP COORDINATE CONVERSION COEFFICIENTS',
     5       ' FOR Y'/'ORPARM(12) = ', 5X, E15.6, 5X, 'RPARM(13) = ',
     6       11X, E15.6/'ORPARM(14) = ', 5X, E15.6, 5X, 'RPARM(15) = ',
     7       11X, E15.6/'0', 7X,
     8       'MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS',
     9       ' FOR LONGITUDE'/'ORPARM(16) = ', 5X, E15.6, 5X,
     *       'RPARM(17) = ', 11X, E15.6/'ORPARM(18) = ', 5X, E15.6, 5X,
     1       'RPARM(19) = ', 11X, E15.6/'0', 7X.
     2       'MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS'.
     3       ' FOR LATITUDE'/'ORPARM(20) = ', 5X. E15.6, 5X,
     4       'RPARM(21) = ', 11x, E15.6/'ORPARM(22) = ', 5X, E15.6, 5X,
     5       'RPARM(23) = ', 11X, E15.6)
      RETURN
      END
```

```
      SUBROUTINE  PRNT(LUN, A. IDIM, IMAX,  JNAX,  SPVAL)
C
C PURPOSE:
C                     TO NORMALIZE AND PRINT THE TWO DIMENSIONAL ARRAY A
C
C ALGORITHM:
C                     PRNT FIRST DETERMINES THE MAXIMUM ABSOLUTE VALUE OF
C                     DATA IN ARRAY A.  ONLY POINTS FOR WHICH A(I,J) IS
C                     NOT EQUAL TO SPVAL ARE CONSIDERED.  IT THEN FINDS
C                     THE POWER OF TEN BY WHICH AMAX MUST BE MULTILPIED
C                     FOR IT TO LIE BETWEEN 100 AND 1000. THE POWER OF
C                     TEN IS PRINTED, FOLLOWED BY THE NORNALIZED DATA,
C                     FORMATTED AS 3 DIGIT INTEGERS. POINTS AT WHICH
C                     A(I,J) IS EQUAL TO SPVAL ARE MASKED BY ASTERISKS.
C                     DATA ARE OUTPUT IN BLOCKS WITH J DECREASING DOWN
C                     AND I INCREASING ACROSS THE PAGE.  THE NUHBER OF
C                     VALUES PRINTED ACROSS A PAGE IS AN INTERNAL PARA-
C                     NETER IN THE SUBROUTINE.
C ARGUMENTS:
C                     LUN - LOGICAL UNIT NUNBER ON WHICH TO PRINT
C                     A - TWO-DIMENSIONAL ARRAY TO BE NORNALIZED AND
C                        PRINTED. UNCHANGED BY PRNT.
C                     IDIM - FIRST DIMENSION OF A AND D IN DIMENSION
C                        STATENENT OF CALLING PROGRAM
C                     IMAX - MAXIMUM I VALUE ACTUALLY USED IN A AND D
C                     JNAX - MAXIMUM J VALUE ACTUALLY USED IN A AND D
C                     SPVAL - SPECIAL MASKING VALUE: ONLY THE A(I,J)
C                              WHICH ARE NOT EQUAL TO SPVAL ARE PRINTED
C
C
      DIMENSION  INTEG(30), A(IDIM,1)
C
C NCOL IS THE NUNBER OF VALUES TO PRINT ACROSS A PAGE
C
      NCOL  =30
C
C AMAX=MAXIMUM ABSOLUTE VALUE OF ARRAY A
C
      AMAX = 0.0
      DO 10 I = 1, IMAX
        DO 10 J = 1, JMAX
          IF (A(I,J) .EQ. SPVAL) GO TO 10
          AMAX = AMAX1(AMAX,ABS(A(I,J)))
   10 CONTINUE
C
C NOW FIND THE POWER OF TEN BY WHICH WE MUST MULTIPLY AMAX
C SO THAT IT FALLS BETWEEN 100 AND 1000.
C
C INITIALLY THE POWER IS ZERO
C
```

```
      MP = 0
      IF (AMAX .EQ. 0) GO TO 20
C
C  TAKE BASE 10 LOGARITHM OF AMAX
C
      AP = ALOG10(AMAX)
C
C  IF AMAX IS GREATER THAN 1000, MP IS NEGATIVE
C
      IF (AP .GT. 3.) HP = -IFIX(AP - 2.)
C
C  IF AMAX IS LESS THAN 100, MP IS POSITIVE
C
      IF (AP .LT. 2.) HP = IFIX(3. - AP)
   20 CONTINUE
C
C  PRINT THE GRID
C
      II = 1
      II = (IMAX - 1) / NCOL + 1
      IRMDR = IMAX - NCOL * (II - 1)
      DO 50 L = 1. II
C
C  WHEN L-II ONLY PRINT IRMDR VALUES
C
         IF (L .EQ. II) NCOL = IRMDR
C
C  PRINT THE POWER
C
         WRITE (LUN,60) MP
         I2 = I1 + NCOL - 1
         DO 40 JJ = 1, JNAX
            J = JMAX - JJ + 1
            DO 30 I = I1, I2
               I3 = 1 + I - I1
               INTEG(I3) = -9999
               IF (A(I,J) .NE. SPVAL) INTEG(I3) = INT(A(I,J)*10.**MP +
     1         SIGN(0.5,A(I,J)*10.**MP))
   30       CONTINUE
            WRITE (LUN,70) (INTEG(I),I=1,NCOL)
   40    CONTINUE
         I1 = I2 + 1
   50 CONTINUE
   60 FORMAT ('1 VALUES MULTIPLIED BY 10**', I3)
   70 FORMAT (' ', 30I4)
      RETURN
      END
```

```
      FUNCTION TAU(T, I, J. K)
C
C PURPOSE :
C                 TO RETURN WIND STRESS DIVIDED BY WATER DENSITY IN
C                 METERS**2/SECOND**2 FOR GRID BOX I,J AT TIME T.
C                 THE PARANETER K INDICATES WHICH COMPONENT OF WIND
C                 STRESS IS RETURNED. THE X-CONPONENT OF WIND STRESS
c                 IS CALCULATED AT THE MIDDLE OF THE RIGHT SIDE OF THE
C                 GRID BOX. THE Y-CONPONENT IS CALCULATED AT THE MIDDLE
C                 OF THE TOP SIDE. BOTH COMPONENTS ARE LINEAR FUNCTIONS
C                 OF X AND Y.
C
C ALGOR ITHN:     THIS FUNCTION READS NETEOROLOGICAL DATA FRON A FILE,
C                 CONVERTS IT TO WIND STRESS, AND INTERPOLATES THE WIND
C                 STRESS IN TINE AND SPACE. ALL DATA FOR THE SANE TINE
C                 ARE GROUPED TOGETHER, WITH A MAXIMUM OF25 STATIONS IN
C                 A GROUP. AS DATA RECORDS ARE READ, SUBROUTINE UZL IS
C                 USED TO CONVERT NETEOROLOGICAL NEASURENENTS TO SUR-
C                 FACE STRESS FOR EACH STATION. THE TWO COMPONENTS OF
C                 THE WIND STRESS ARE ASSUMED TO BE LINEAR FUNCTIONS OF
C                 X AND Y AND THE BEST-FIT LINEAR SURFACE FOR THE GIVEN
C                 DATA POINTS IS CALCULATED. EACH TINE TAU IS CALLED,
C                 THE TINE PARANETER IS CHECKED AGAINST THE TINE OF THE
C                 LAST SET OF NETOROLOGICAL DATA READ. IF IT IS GREATER,
C                 ANOTHER GROUP IS READ. IF IT IS LESS, THE APPROPRIATE
C                 WIND STRESS CONPONENT IS CALCULATED USING LINEAR INTER-
C                 POLATION BETWEEN THE LAST TWO LINEAR SURFACES COM-
C                 PUTED FOR THAT COMPONENT. IF THE END-OF-FILE IS EN-
C                 COUNTERED, WIND STRESS IS CALCULATED WITH THE LAST
C                 LINEAR SURFACE COEFFICJENTS.
C
C                 THE X-COMPONENT OF STRESS FOR GRID BOX I,J HAS CO-
C                 ORDINATES (I*DELTA,(J-1/2)*DELTA) RELATIVE TO THE GRID
C                 ORIGIN, AND THE Y-CONPONENT OF STRESS HAS COORDINATES
C                 ((I-1/2)*DELTA,J*DELTA), WHERE DELTA IS THE GRID SIZE.
C
C                 MANY OF THE VARIABLES USED IN THIS FUNCTION ARE ASSUM-
C                 ED TO HAVE RETAINED THEIR VALUES FROM THE PREVIOUS
C                 CALL. IF YOUR FORTRAN SYSTEM DOES NOT DO THIS AUTO-
C                 NATICALLY, PROVISION MUST BE MADE TO RETAIN VALUES
C                 FOR THE VARIABLES:ISTA, AOLD1, BOLD', COLD', AOLD2.
C                 BOLD2, COLD2, ANEW1, BNEW1, CNEW1, ANEW2, BNEW2, CNEW2,
C                 TOLD, TNEW, TLAST. RLAT. RLON, Z, TA, TW, WS, WD.
C
C ARGUNENTS:
C                 T - TINE IN SECONDS FROM BEGINNING OF RUN AT WHICH
C                     STRESS IS TO BE CALCULATED
C                 I - FIRST INDEX OF GRID BOX FOR WHICH STRESS IS TO
C                     BE CALCULATED
C                 J - SECOND INDEX OF GRID BOX FOR WHICH STRESS IS TO
```

```
C                    BE  CALCULATED
C                 K - SPECIFIES WHETHER X-COMPONENT OR Y-COMPONENT
C                    OF STRESS IS RETURNED.  IF K=1, THE X-CONPONENT
C                    OF STRESS AT THE MIDDLE RIGHT SIDE OF GRID BOX
C                    I, J IS RETURNED.  IF K=2, THE Y-CONPONENT OF
C                    STRESS AT THE MIDDLE TOP SIDE OF GRID BOX I, J
C                    IS RETURNED.
C
C INPUT:
c  LOGICAL UNIT 8:
C      METEOROLOGICAL DATA FILE
C      CARD COLUNN        FORMAT       PARANETER
C          1-10            G10.4       TLAST - TINE (HRS) FRON BEGINNING
C                                                OF RUN
C         1 1-20           610.4       RLAT - LATITUDE INDEGREES NORTH
C         21-30            G10.4       RLON - LONGITUDE IN DEGREES WEST
C         31-40            G10.4       Z - HEIGHT OF INSTRUMENTS
C         41-50            G10.4       TA - TEMPERATURE OF AIR
C         51-60            G10.4       TW - TENPERATURE OF WATER
C         61-70            G10.4       WS - WIND SPEED (M/S)
C         71-80            G10.4       WD - WIND DIRECTION (DEG)
C
C         ALL DATA FOR THE SANE TINE ARE GROUPED TOGETHER, WITH A
C         MAXIMUM OF 25 STATIONS IN A GROUP.
C
C   *NOTE THAT END OF FILE IS INDICATED BY A RECORD WITH A NEGATIVE
C         TINE.
C
C OUTPUT :
C  LOGICAL UNIT 6:
C      PRESENTATION OF METEOROLOGICAL DATA STATION BY STATION
C
C COMMON BLOCKS:
C                 /GPARM/RPARM(23), I PARM(54)
C
C  SUBPROGRAM :
C      FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN
C                         GIVEN LATITUDE AND LONGITUDE
C      FUNCTION YDIST - RETURNS Y DISTANCE FRON GRID ORIGIN
C                         GIVEN LATITUDE AND LONGITUDE
C      SUBROUTINE UZL - RETURNS DRAG COEFFICIENT CD
C
C HISTORY :
C             WRITTEN BY A. T. JESSUP, 1981, GLERL, ANN ARBOR. MI
C
      DIMENSION X (25). U(2), Y(25), ATAU(25.2)
      LOGICAL XEQ. YEQ
      COMMON /GPARM/ RPARN(23), I PARM(54)
      DATA RHOAW, ISTA /1.25E-3, 0/
      DATA LUN /8/
```

```
      DATA AOLD1, BOLD', COLD', AOLD2, BOLD2, COLD2, TOLD /7*0./
C
C  IF THIS IS THE FIRST TINE THROUGH, READ A RECORD
C
      IF (ISTA .EQ. 0) GO TO 30
C
C  CHECK IF NECESSARY TO READ ANOTHER RECORD
C
      IF (T .LT. TNEW .OR. TNEW .LT. 0.) GO TO 90
   10 AOLD1 = ANEW'
      BOLD' = BNEW1
      COLD' = CNEW1
      AOLD2 = ANEW2
      BOLD2 = BNEW2
      COLD2 = CNEW2
      TOLD = TNEW
   20 TNEW = TLAST
      IF ( TNEW .LT. 0) GO TO 90
C
C  FIND ATAU, THE WIND STRESS FOR THE CURRENT STATION
C
      TD = TA - TW
      X(ISTA) = XDIST(RLAT,RLON)
      Y(ISTA) = YDIST(RLAT,RLON)
      U(1) = WS * COS((270. - WD - RPARM (6) - RPARM (7) )*ATAN(1.) /45.)
      u(2) = ws * SIN((270. - WD - RPARN (6) - RPARM(7))*ATAN(1.) /45.)
      CALL UZL(WS, Z, TD, Z, CO, CH, ZO, FL)
      CDD = CD * 1.E3
      TTLAST = TLAST / 3600.
      IF (ISTA .LE. 1) WRITE (6,160)
      XKN = X(ISTA)/1000.
      YKN = Y(ISTA)/1000.
      WRITE (6,150) TTLAST, RLAT, RLON, Z, TA, TW. WS, WD, CDD,
     1 XKM, YKN
      ATAU(ISTA,1) = CD * RHOAW * U(1) * ws
      ATAU(ISTA,2) = CD * RHOAW * U(2) * ws
   30 READ (LUN,130) TLAST, RLAT, RLON, Z, TA, Tw. WS, WD
      TLAST = TLAST * 3600.
      IF (T .LT. TLAST .AND. ISTA .EQ. 0) GO TO 120
      ISTA = ISTA + 1
C
C  IF FIRST TINE THROUGH, FIND ATAU
C
      IF (ISTA .EQ. 1) GO TO 20
C
C  CHECK IF LAST RECORD AT TINE TLAST HAS BEEN READ
C
      IF (TLAST .EQ. TNEW) GO TO 20
C
C  NOW FIND THE BEST-FIT LINEAR SURFACE
```

C
C STATEMENT FUNCTION
C DET

```
C
      SX = 0.
      SY = 0.
      SXY = 0.
      SX2 = 0.
      SY2 = 0.
      SXTAU1 = 0.
      SYTAU1 = 0.
      SATAU1 = 0.
      SXTAU2 = 0.
      SYTAU2 = 0.
      SATAU2 = 0.
      N = ISTA - 1
      AN = FLOAT(N)
      XEQ = .TRUE.
      YEQ = .TRUE.
      DO 40 IN = 1, N
        SX = SX + X(IN)
        SY = SY + Y(IN)
        SXY = SXY + X(IN) * Y(IN)
        SX2 = SX2 + X(IN) ** 2
        SY2 = SY2 + Y(IN) ** 2
        SXTAU1 = SXTAU1 + X(IN) * ATAU(IN,1)
        SYTAU1 = SYTAU1 + Y(IN) * ATAU(IN,1)
        SXTAU2 = SXTAU2 + X(IN) * ATAU(IN,2)
        SYTAU2 = SYTAU2 + Y(IN) * ATAU(IN,2)
        SATAU1 = SATAU1 + ATAU(IN,1)
        SATAU2 = SATAU2 + ATAU(IN,2)
        IF (X(IN) .NE. X(1)) XEQ = .FALSE.
        IF (Y(IN) .NE. Y(1)) YEQ = .FALSE.
   40 CONTINUE
C
C CALCULATE COEFFICIENTS ANEW, BNEW, CNEW, WHERE
C                        TAU = ANEW * X + BNEW * Y + CNEW
C                        FOR EACH COMPONENT.
C
      ANEW1 = 0.
      ANEW2 = 0.
      BNEW1 = 0.
      BNEW2 = 0.
      CNEW1 = ~~ATAU(ISTA - 1,1)~~ SATAU1 / AN
      CNEW2 = ~~ATAU(ISTA - 2)~~ SATAU2 / AN
C
C SOLVE THE SYSTEM OF EQUATIONS :
C
C  1)   : SX2  SXY  SX :   : ANEW :   : SXTAU :
C  2)   : SXY  SY2  SY : * : BNEW : = : SYTAU :
C  3)   : SX   SY   AN :   : CNEW :   : SATAU :
C
C BY COMBINING EQUATIONS 1 AND 2 AND EQUATIONS 2 AND3, WE
```

```
C     ELIMINATE B AND OBTAIN 2 EQUATIONS IN A AND C. THEN
C     EXPRESSIONS FOR A AND C ARE OF THE FORN:
C                     A = EXPRESSION' / ACHK
C                     c = EXPRESSION2 / (-ACHK)
C     WHERE ACHK IS THE DETERMINANT OF THE 2 EQUATION SYSTEM.  IF ALL Y
C     VALUES ARE EQUAL. WE SET A = 0 AND USE A DIFFERENT EXPRESSION TO
C     FIND C.
C
C     BY COMBINING EQUATIONS 1 AND 2 AND EQUATIONS 2 AND 3, WE
C     ELIMINATE A AND OBTAIN 2 EQUATIONS IN B AND C. THEN
C     EXPRESSIONS FOR B AND C ARE OF THE FORN:
C                     B = EXPRESSION3 / BCHK
C                     c = EXPRESSION4 / (-BCHK)
C     WHERE BCHK IS THE DETERMINANT OF THE 2 EQUATION SYSTEM. IF ALL X
C     VALUES ARE EQUAL, WE SET B = 0 AND USE A DIFFERENT EXPRESSION TO
C     FIND C.
C
C     IF BOTH A AND B EQUAL 0, THEN C HAS THE DEFAULT VALUE ASSIGNED
C     ABOVE. (THIS WILL OCCUR IF ONLY ONE/STATION IS USED.)
C
         IF (YEQ) GO TO 50
         ACHK = (SY2*SX2 - SXY**2) * (SY**2 - AN*SY2) - (SY*SXY - SX*SY2)
        1 (SX*SY2 - SY*SXY)
         ANEW' = ((SXTAU1*SY2 - SYTAU1*SXY)*(SY**2 - AN*SY2) - (SYTAU1*SY
        1 SATAU1*SY2)*(SX*SY2 - SY*SXY)) / ACHK
         ANEW2 = ((SXTAU2*SY2 - SYTAU2*SXY)*(SY**2 - AN*SY2) - (SYTAU2*SY
        1 SATAU2*SY2)*(SX*SY2 - SY*SXY))' / ' ACHK
   50    IF (XEQ) GO TO 60
         BCHK = (SXY**2 - SX2*SY2) * (SY*SX - AN*SXY) - (SY2*SX - SY*SXY)
        1 (SX*SXY - SY*SX2)
         BNEW1 = ((SXTAU1*SXY - SYTAU1*SX2)*(SY*SX - AN*SXY) - (SYTAU1*SX
        1 SATAU1*SXY)*(SX*SXY - SY*SX2)) / BCHK
         BNEW2 = ((SXTAU2*SXY - SYTAU2*SX2)*(SY*SX - AN*SXY) - (SYTAU2*SX
        1 SATAU2*SXY)*(SX*SXY - SY*SX2)) / BCHK
   60 IF (XEQ .AND. YEQ) GO TO 80
         IF (YEQ) GO TO 70
         CNEW1 = ((SXTAU1*SY2 - SYTAU1*SXY)*(SY*SXY - SX*SY2) - (SYTAU1*SY
        1- SATAU1*SY2)*(SY2*SX2 - SXY**2)) / (-ACHK)
         CNEW2 = ((SXTAU2*SY2 - SYTAU2*SXY)*(SY*SXY - SX*SY2) - (SYTAU2*SY
        1- SATAU2*SY2)*(SY2*SX2 - SXY**2)) / (-ACHK)
         GO TO 80
   70 CNEW1 = ((SXTAU1*SXY - SYTAU1*SX2)*(SY*SX - AN*SXY) - (SYTAU1*SX -
        1 SATAU1*SXY)*(SX*SXY - SY*SX2)) / (-BCHK)
         CNEW2 = ((SXTAU2*SXY - SYTAU2*SX2)*(SY*SX - AN*SXY) - (SYTAU2*SX -
        1 SATAU2*SXY)*(SX*SXY - SY*SX2)) / (-BCHK)
   80    CONTINUE
         ISTA = 1
         IF (T .GT. TNEW) GO TO 10
         WRITE (6,170)
C
```

```
C   DETERMINE PROPER LOCATION XS,YS AND CALCULATE STRESS
C
   90 IF (K .EQ. 2) GO TO 100
      XS = I * RPARM(3)
      YS = (FLOAT(J) -.5) * RPARM(3)
      TAUNEW = ANEW1 * XS + BNEW1 * YS + CNEW1
      TAUOLD = AOLD1 * XS + BOLD' * YS + COLD1
      GO TO 110
  100 xs = (FLOAT(I) - .5) * RPARM(3)
      YS = J * RPARM(3)
      TAUNEW = ANEW2 * XS + BNEW2 * YS + CNEW2
      TAUOLO = AOLD2 * XS + BOLD2 * YS + COLD2
C
C   INTERPOLATE IN TINE
C
  110 TAU = (TAUNEW - TAUOLD) / (TNEW - TOLD) * (T - TOLD) + TAUOLD
      GO TO 180
  120 WRITE (6.140) TLAST. T
      STOP
  130 FORMAT (8G10.4)
  140 FORMAT (' TINE OF FIRST METEOROLOGICAL DATA', G'0.4,
     1        ' SECONDS IS', ' GREATER THAN T = ', G10.4,
     2        ' - PROGRAM TERMINATED')
  150 FORMAT (' ', F5.1, ' * ', F10.7, ' * ', F10.7, ' * ', F4.1, ' * ',
     1        F5.2, ' * ',F5.2, ' * ', F6.2, ' * ', F4.0, ' * ', F5.2,
     2        ' * ', F5.0, ' * ', F5.0)
  160 FORMAT ('', 95('*')/
     1        ' TINE * LATITUDE * LONGITUDE * Z * T-AIR * ',
     2        'T-H20 * W-SPD * W-DIR * CDE3 * X * Y'/' ', 95('*'))
  170 FORMAT (' ', 95('*'))
  180 RETURN
      END

  175 FORMAT (' THESE DATA POINTS ARE CO-LINEAR OR
            NEARLY CO-LINEAR')
```

```
C
C  INITIAL  GUESS  FOR  ZO
C
      ZO = .00459 * UST1 * UST1
      S = UN * UN * TBAR / (9.8*DTHETA)
      IF (ABS(S) .GT. 1.E6) S = SIGN(1.E6,S)
      X = ALOG(H/ZO)
C
C  INITIAL  GUESS  FOR  L
C
      FL = S / X
      DO 60 ITER = 1, 20
        X = ALOG(H/ZO)
        IF (ABS(FL) .GT. 3.E6) FL = SIGN(3.E6,FL)
        IF (FL .GT. 0.) GO TO 20
C
C  UNSTABLE  SECTION  (L LT 0 OR DT LT 0)
C
        FLI = 1. / FL
C
C ASSUME  5  ITERATIONS  SUFFICIENT
C
        DO 10 I = 1, 5
          X1 = GANT * FLI
          ARG1 = SQRT(1. - X1*H)
          ARG2 = SQRT(1. - X1*ZO)
          A = BETA * ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*(ARG2 -
     1    1.)))
          X1 = GANN * FLI
          ARG1 = (1. - X1*H) ** (.25)
          ARG2 = (1. - X1*ZO) ** (.25)
          B = ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*(ARG2 - 1.))) +
     1    2. * (ATAN(ARG1) - ATAN(ARG2))
          FL = S * A / (B*B)
          IF (ABS(FL) .GT. 3.~6) FL = SIGN(3.E6,FL)
          FLI = 1. / FL
   10   CONTINUE
        GO TO 50
C
C  STABLE  SECTION
C
C  TRY  MILDLY  STABLE-
C
   20   CONTINUE
        AA = X * X
        X1 = H - ZO
        BB = 9.4 * X1 * X - .74 * S * X
        cc = 4.7 * X1
        cc = cc * cc - cc * S
        ROOT = BB * BB - 4. * AA * CC
```

```
      IF (ROOT .LT. 0.) GO TO 30
      FL = (-BB + SQRT(ROOT)) / (2.*AA)
      IF (FL .LE. H) GO TO 30
      B = X + 4.7 * X1 / FL
      A = BETA * X + 4.7 * X1 / FL
      GO TO 50
C
C STRONGLY STABLE-
C
   30 CONTINUE
      IF (FL .LE. ZO) FL = ZO t 1.E-5
      DO 40 I = 1, 5
        ARG1 = FL / ZO
        X1 = ALOG(ARG1)
        X2 = ALOG(H/FL)
        ARG1 = 1. - 1. / ARG1
        A = .74 * X1 + 4.7 * ARG1 + 5.44 * X2
        B = X1 + 4.7 * ARG1 + 5.7 * X2
        FL = A * S / (B*B)
        IF (FL .LE. ZO) FL = ZO + 1.E-5
        IF (FL .GT. H) FL = H
   40 CONTINUE
C
C CALCULATE USTAR AND ZONEW
C
   50 CONTINUE
      TSTAR = FK * DTHETA / A
      USTAR = FK * UN / B
      ZONEW = .00459 * USTAR * USTAR
      IF (ITER .GT. 5 .AND. ABS((USTAR - UST1)/UST1) .LT. EPS)
     1   GO TO 80
      UST1 = USTAR
      ZO = ZZONEW
   60 CONTINUE
C
C IF CONE HERE, TOO NANY ITERATIONS (UGH - UGH)
C
      WRITE (6,701
   70 FORMAT ('OTOO MANY ITERATIONS ON ZO IN SUBROUTINE UZL - CHECK ',
     1        'METEOROLOGICAL DATA - PROGRAM TERMINATED')
      STOP
   80 CONTINUE
      ZO = ZONEW
      CO = (USTAR/UM) ** 2
      CH = FK * FK / (A*B)
   go RETURN
      END
```

```
      FUNCTION XDIST (RLAT, RLON)
C
C  PURPOSE:
C                  TO RETURN X DISTANCE IN METERS FRON THE GRID ORIGIN
C                  DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN
C                  LATITUDE AND LONGITUDE
C  ARGUMENTS:
C                  RLAT - LATITUDE OF POINT
C                  RLON - LONGITUDE (WEST) OF POINT
C                  RPARM,IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                        PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C COMMON BLOCK:  /GPARM/ RPARN (23) , I PARN (54)
C
      COMMON /GPARM/ RPARN (23) , IPARM (54)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8) * DLON + RPARM(9) * DLAT + RPARM(10) * DLON * DLAT +
     1     RPARM(11) * DLON ** 2
      YP = RPARM(12) * DLON + RPARM(13) * DLAT + RPARM(14) * DLAT *
     1     DLON + RPARM(15) * DLON ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      XDIST = (XP*COS(ALPHA) + YP*SIN(ALPHA)) * 1000.
C
C  NOW TRANSLATE
C
      XDIST = XDIST + IPARM(3) * RPARM(3)
      RETURN
      END
```

```
      FUNCTION YDIST(RLAT, RLON)
C
C  PURPOSE:
C                 TO RETURN Y DISTANCE IN METERS FROM THE GRID ORIGIN
C                 DESCRIBED BY THE COMMON BLOCK / GPARM /, GIVEN
C                 LATITUDE AND LONGITUDE
C ARGUMENTS:
C                 RLAT - LATITUDE OF POINT
C                 RLON - LONGITUDE (WEST) OF POINT
C                 RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                        PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C COMMON BLOCK:   /GPARM/ RPARM(23),IPARM(54)
C
      COMMON /GPARM/ RPARM(23), IPARM(54)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARN(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8) * DLON + RPARM(9) * DLAT + RPARM(10) * DLON * DLAT +
     1     RPARM(11) * DLON ** 2
      YP = RPARN(12) * DLON + RPARM(13) * DLAT + RPARM(14) * DLAT *
     1     DLON + RPARM(15) * DLON ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C   FIRST ROTATE
C
      YDIST = (YP*COS(ALPHA) - XP*SIN (ALPHA)) * 1000.
C
C   NOW TRANSLATE
C
      YDIST = YDIST + IPARM(4) * RPARM(3)
      RETURN
      END
```

```
      FUNCTION  RLAT(X,Y)
C
C  PURPOSE:
C                 TO RETURN LATITUDE OF A POINT ON THE GRID
C                 DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN THE
C                 X AND Y DISPLACEMENTS FRON THE GRID ORIGIN
C  ARGUNENTS:
C                 X - X DISTANCE FRON THE GRID ORIGIN (M)
C                 Y - Y DISTANCE FRON THE GRID ORIGIN (M)
C                 RPARM, IPARN - ARRAYS CONTAINING BATHYMETRIC GRID
C                     PARANETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C COMMON BLOCK:  /GPARM/RPARM(23), IPARM(54)
C
      COMMON/GPARM/RPARM(23), IPARM(54)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
C
C TRANSFORM THE POINTS TO THE 'PRIMED' COORDINATE SYSTEM,
C  IE., THAT OF THE STANDARD BATHYMETRIC GRID
C
C  FIRST TRANSLATE
C
      XX = X - IPARM(3) * RPARM(3)
      YY = Y - IPARN(4) * RPARM(3)
C
C  NOW ROTATE
C
      XP=(XX*COS(ALPHA)-YY*SIN(ALPHA))/1000.
      YP=(YY*COS(ALPHA)+XX*SIN(ALPHA))/1000.
      DLAT = RPARM(20) * XP + RPARM(21) * YP + RPARM(22) * XP * yp +
     1       RPARM(23) * XP ** 2
      RLAT = RPARM(1) + DLAT
      RETURN
      END
```

```
      FUNCTION RLON (X,Y)
C
C  PURPOSE:
C                   TO RETURN LONGITUDE OF A POINT ON THE GRID
C                   DESCRIBED BY THE CONMON BLOCK /GPARM/, GIVEN THE
C                   X AND Y DISPLACEMENTS FROM THE GRID ORIGIN
C ARGUMENTS:
C                   X - X DISTANCE FRON THE GRID ORIGIN (M)
C                   Y - Y DISTANCE FRON THE GRID ORIGIN (N)
C                   RPARN, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                        PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C  COMMON BLOCK:  /GPARM/RPARM (23) , I PARH (54)
C
      COMMON/GPARM/RPARM (23) . I PARM (54)
      PI = ATAN2(O.,-1.)
      ALPHA = RPARN (7) * PI / 180.
C
C TRANSFORM THE POINTS TO THE 'PRIMED' COORDINATE SYSTEM,
C  IE., THAT OF THE STANDARD BATHYNETRIC GRID
C
C  FIRST TRANSLATE
C
      XX = X + IPARM(3) * RPARM(3)
      YY = Y + IPARM(4) * RPARM(3)
C
C  NOW ROTATE
C
      XP=(XX*COS (ALPHA) -YY*S IN (ALPHA)) /1000.
      YP=(YY*COS(ALPHA)+XX*SIN(ALPHA))/1000.
      DLON = RPARM(16) * XP + RPARM(17) * YP + RPARN(18) * Xp * Yp +
     1       RPARM(19) * XP ** 2
      RLON = RPARN(2) - DLON
      RETURN
      END
```

Appendix B. Sample runs of programs GRID, FRSFC, AND RLID

```
             2000.                    45.
  0     0     0
```

Input for program GRID on LUN 5

```
LAKE ST. CLAIR BATHYMETRY
 35    36   42.30415     82.93158     1200     6
  .822690E+02    -.418687E+01    -.892958E+00    .549244E+00
  .284351E+01     .110232E+03     .182918E+00    .235336E+00
  .121387E-01     .462637E-03     .110856E-05   -.102938E-05
 -.313049E-03     .905963E-02    -.238882E-06   -.279918E-06
 0  0  0  0  0  0  0  0  0  1  1  1  2  2  2  2  2  1  1   1
 1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   2
 0  0  0  ,  1  1  2  3  3  3  3  3  3  3  3  3  3  3  2   3
 1  1  0  0  0  0  1  1  1  0  0  0  0  0  0  0  0  1  1   4
 2  2  3  3  3  4  4  4  4  4  4  4  3  3  4  3  3  2  1   5
 1  1  1  2  2  2  1  0  0  0  4  3  2  2  2  2  3  3  3   6
 3  4  4  4  5  5  4  4  4  4  4  4  4  4  3  3  3  3  3   7
 3  3  3  3  2  0  0  4  4  3  3  3  3  3  4  3  4  4  4   8
 5  5  5  5  5  4  4  4  4  4  4  4  4  3  3  4  4  4  4   9
 3  3  ,  0  2  4  5  3  3  4  4  4  4  4  4  5  5  5  5  10
 5  5  4  4  5  5  5  5  4  4  4  4  4  4  4  4  4  3  3  11
 1  0  1  5  4  3  3  4  4  4  4  5  5  5  5  5  5  5  4  12
 5  5  5  5  5  5  5  5  5  5  5  5  5  5  4  4  1  0  0  13
 2  5  5  2  3  4  4  5  5  5  5  5  5  5  5  5  5  5  5  14
 5  5  5  5  5  5  5  5  5  5  4  3  3  1  0  0  0  3  5  15
 4  2  4  4  4  4  5  5  5  5  5  5  5  5  5  5  5  5  5  16
 5  5  5  5  5  5  4  4  3  2  1  0  0  0  3  4  5  5  4  17
 4  4  4  5  5  5  6  5  5  5  5  5  5  5  5  5  5  5  5  18
 5  4  4  4  3  4  3  1  0  0  0  3  4  4  5  5  4  5  5  19
 5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  4  4  3  20
 3  3  3  2  0  0  0  0  2  4  4  4  5  5  5  5  6  5  6  2,
 6  5  5  5  5  5  5  5  5  5  4  4  4  3  3  3  3  3  3  22
 1  0  0  0  0  0  3  4  4  4  5  6  5  5  5  5  6  6  6  23
 5  5  5  5  5  5  5  4  4  4  3  3  1  1  3  2  1  0  0  24
 0  0  0  3  4  4  4  4  5  6  5  5  5  5  5  5  5  6  5  25
 5  5  5  5  4  4  3  2  2  1  1  2  2  1  0  0  0  0  0  26
 3  4  4  4  4  5  6  6  5  5  5  5  5  5  5  5  5  5  4  27
 3  3  2  1  1  1  1  1  1  1  0  0  0  0  0  1  3  4  4  28
 4  5  5  5  6  6  5  5  5  5  5  5  5  4  3  2  1  1  1  29
 1  1  1  1  1  1  1  1  1  0  0  0  2  3  4  4  5  5  5  30
 5  5  6  6  5  5  5  5  4  4  3  2  1  1  ,  1  1  1  1  31
 1  1  ,  1  1  1  0  0  0  2  3  4  4  4  5  5  5  5  5  32
 6  6  5  5  5  4  4  2  1  1  1  1  1  1  0  1  1  1  0  33
 0  1  1  0  0  0  2  3  4  4  4  5  4  5  5  5  5  5  5  34
 4  4  4  4  2  1  0  0  1  1  1  0  0  0  0  0  0  0  0  35
 0  0  0  1  3  3  4  4  4  4  4  4  5  4  4  6  3  2  2  36
 3  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  37
 1  3  3  4  4  4  4  4  4  4  4  2  0  0  0  ,  ,  1  1  38
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  39
 3  4  4  4  4  4  4  3  1  1  0  0  0  0  0  0  0  0  0  40
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  3  3  3  41
 3  3  3  3  2  1  1  1  0  0  0  0  0  0  0  0  0  0  0  42
 0  0  0  0  0  0  0  0  0  0  0  0  2  3  3  2  1  2  3  43
 3  1  2  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  44
 0  0  0  0  0  0  0  0  0  0  1  1  1  0  ,  2  2  1  1  45
 1  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  46
 0  0  0  0  0  0  0  0  0  0  1  2  2  2  1  1  1  0  0  47
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  48
 0  0  0  0  0  0  0  ,  2  3  2  1  1  1  1  0  0  0  0  49
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  50
 0  0  0  0  0  2  3  2  1  0  0  1  0  0  0  0  0  0  0  51
```

Input for program GRID on LUN 7

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	52
1	1	2	3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	53
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	54
3	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	3	3	2	56
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57
0	0	0	0	0	0	0	0	0	1	1	2	2	3	3	2	1	1	1	58
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59
0	0	0	0	0	0	0	0	1	1	3	3	3	2	1	1	1	1	1	60
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	61
0	0	0	0	0	0	0	1	3	3	3	2	2	2	1	1	0	0	0	62
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63
0	0	0	0	0	1	2	3	3	3	2	1	1	0	0	0	0	0	0	64
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65
0	0	0	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	66
0	0	0	0	0	0														67

Input for program GRID on LUN 7
(continued)

PROGRAM GRID

DESIRED GRID SIZE (M) = 2000. DESIRED ORIENTATION (DEGREES) = 45

LAKE ST. CLAIR BATHYMETRY

OLD GRID

GRID DESCRIPTION FOR LAKE ST. CLAIR BATHYMETRY

IDIMENSION : IPARM(1) = 35 JDIMENSION : IPARM(2) = 36

DISPLACEMENT OF ORIGIN IN NUMBER OF GRID SQUARES

IDISPLACEMENT : IPARM(3) = 0 JDISPLACEMENT : IPARM(4) = 0

BASE LATITUDE : RPARM(1)= 42.3041534

BASE LONGITUDE : RPARM(2)= 92.9315796

GRID SIZE (M) : RPARM(3) = 1200. MAX DEPTH (M) : RPARM(4)= 6.

MIN DEPTH (M) : RPARM(5)= 0. BASE ROTATION : RPARM(6)= 0.0

ANGLE ROTATED : RPARM(7)= 0.

GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICENTS FOR X

RPARM(8) = 0.822690E+02 RPARM(9) = -0.418687E+01

RPARM(10) = -0.892958E+00 RPARM(11) = 0.549244E+00

GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICIENTS FOR Y

RPARM(12) = 0.284351E+01 RPARM(13) = 0.110232E+03

RPARM(14) = 0.182918E+00 RPARM(15) = 0.235336E+00

MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LONGITUDE

RPARM(16) = 0.121387E-01 RPARM(17) = 0.462637E-03

RPARM(18) = 0.110856E-05 RPARM(19) = -0.102938E-05

MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LATITUDE

RPARM(20) = -0.313049E-03 RPARM(21) = 0.905963E-02

RPARM(22) = -0.238882E-06 RPARM(23) = -0.279918E-06

Page 1 of printed output from program GRID

```
VALUES MULTIPLIED BY 10**  2
******************** ************************************   100 200 100*****************************************
*******  ****************************************   100 200 300 300 300 200 100 100*******************************
****************************************** 100 300 300 300 2   200 200 100 100***********************************
************************************** 100 100 300 300 300 200 100 100 100 **0 100*******************************
***************************** 100 100 200 200 300 300 200 100 100 100**************************************
***************************** 100 200 200 300 300 300 200 100 100 100**************************************
***************************** 100 100 200 200 300 300 100 100 100*******************************************
***************************** 100 100 100 200 300 300 100**************************************************
*************************************** 200 300 200 100********* 100***************************************
********************************** 100 200 300 200 100 100 100 100***************************************
********************************** 100 200 200 200 100 100 100*******************************************
************************** 100 100 100**** 100 200 200 100 100 100**** 1   1*0***********************************
********************** 200 300 300 200 100 200 300 300 100 200 100 100 100**o*********************************
***************** 100 200 300 300 300 300 300 300 300 200 100 100 100*****************************************
***************** 200 300 300 400 400 400 400 400 400 300 100 100*********************************************
************ 100 300 300 400 400 400 400 400 400 400 400 200************ 100 100 100 100**************************
************ 100 300 300 400 400 400 400 400 400 500 400 400 600 300 200 200 300 100 100**************************
************ 200 300 400 400 400 500 400 500 500 500 500 500 500 400 400 400 400 200 100******** 100 100 100*********
************ 200 300 400 400 400 500 500 500 500 500 600 600 500 500 500 400 400 200 100 100 100 100 100 100*** 100 100
************ 200 300 400 400 500 500 500 500 500 600 600 500 500 500 500 400 400 300 200 100 100 100 100 100 100 100 100
************ 100 300 400 400 400 500 500 500 600 600 500 500 500 500 500 500 500 400 300 200 100 100 100 100 100 100 100
**************** 300 400 400 400 400 500 600 600 500 500 500 500 500 500 500 500 500 500 400 300 300 100 100 100 100 100
**************** 300 400 400 400 400 500 600 500 500 500 500 500 500 500 600 500 500 500 500 500 400 100 300 200 200 100
**************** 300 400 400 400 500 600 500 500 500 500 600 600 600 500 500 500 500 500 500 500 400 200 400 300 300 100
************ 200 400 400 400 500 500 500 500 600 500 600 600 500 500 500 500 500 500 500 500 500 400 400 400 300 300 300
************ 300 400 400 500 500 400 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 400 500 400 400 300
************ 300 400 500 500 400 400 400 400 500 500 500 600 500 500 500 500 500 500 500 500 500 500 500 500 500 400 400
************ 300 500 400 200 400 400 400 400 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500
******** 200 500 500 200 300 400 400 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500 500
**** 100 500 400 300 300 400 400 400 400 500 500 500 500 500 500 500 400 500 500 500 500 500 500 500 500 500 500 500 500
 200 400 500 300 300 400 400 400 400 400 400 500 500 500 500 500 500 400 400 500 500 500 500 400 400 500 400 400 400 400
 400 400 300 300 300 300 300 400 300 400 400 400 500 500 500 500 500 400 400 400 400 400 400 400 400 400 300 400 400 400
 400 300 200 200 200 200 300 300 300 300 400 400 400 500 500 400 400 400 400 400 400 400 400 300 300 300 300 300 300 300
 **************** 100 100 200 200 300 300 300 400 400 400 400 400 400 400 300 300 400 300 300 200 100 100 100 100 200 200
 ************************ 100 100 100 200 300 300 300 300 300 300 300 300 300 300 300 200 100 100**************** 100 100
 ******************************** 100 100 100 200 200 200 200 200 100 100 100 100*****************************************
```

Page 2 of printed output from program GRID

VALUES MULTIPLIED BY 10** 2

```
*******************
*******************
*******************
•  ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠♦⊠
•  ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠♦
*******************
*******************
*******************
*******************
•  ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠
*******************
• ⊠♦⊠⊠♦⊠⊠⊠⊠⊠⊠⊠⊠♦⊠⊠⊠⊠⊠⊠
•  ⊠⊠⊠⊠⊠⊠⊠⊠⊠♦⊠⊠⊠⊠⊠⊠⊠⊠
*******************
*******************
•  ⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠⊠♦⊠⊠♦
*******************
•  ⊠⊠⊠⊠⊠⊠⊠⊠⊠♦⊠⊠⊠⊠⊠⊠⊠⊠♦⊠
 100*******   □□□  100
 100 100 100 100  □□□
 100 100   □□□   □□□
 □□□  □□□  100********
 100 200□□    100****
 100  □□□  □□□  100****
 □□□  □□□  □□□  100****
 □□□  300  □□□  200****
 □□□  □□□  □□□  □□□  □□□
 □□□  400□□   □□□   •□□
 500  □□□  □□□  300  □□□
 □□□  500  ••□□  □□□  100
 □□□  □□□□   □□□   □□□
 ••□□  300  □□□  100***
 □□□  □□□  200********
 200  100************
 100***************
•  ⊠⊠♦⊠♦⊠⊠♦♦⊠⊠⊠⊠⊠⊠⊠⊠⊠
```


Page 3 of printed output from program GRID

GRID MODIFICATIONS
----

4 of output GRID

NEW GRID

GRID DESCRIPTION FOR LAKE ST. CLAIR BATHYMETRY

IDIMENSION : IPARM(1) = 24 JDIMENSION : IPARM(2) = 26

DISPLACEMENT OF ORIGIN IN NUMBER OF GRID SQUARES

IDISPLACEMENT : IPARM(3) = 0 JDISPLACEMENT : IPARM(4) = 14

BASE LATITUDE : RPARM(1) = 42.3041534

BASE LONGITUDE : RPARM(2) = 82.9315796

GRID SIZE (M) : RPARM(3) = 2000. MAX DEPTH (M) : RPARM(4) = 6.

MIN DEPTH (M) : RPARM(5) = 1. BASE ROTATION : RPARM(6) = 0.0

ANGLE ROTATED : RPARM(7) = 45.

GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICENTS FOR X

RPARM(8) = 0.822690E+02 RPARM(9) = -0.418687E+01

RPARM(10) = -0.892958E+00 RPARM(11) = 0.549244E+00

GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICIENTS FOR Y

RPARM(12) = 0.284351E+01 RPARM(13) = 0.110232E+03

RPARM(14) = 0.182918E+00 RPARM(15) = 0.235336E+00

MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LONGITUDE

RPARM(16) = 0.121387E-01 RPARM(17) = 0.462637E-03

RPARM(18) = 0.110856E-05 RPARM(19) = -0.102938E-05

MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LATITUDE

RPARM(20) = -0.313049E-03 RPARM(21) = 0.905963E-02

RPARM(22) = -0.238882E-06 RPARM(23) = -0.279918E-06

Page 5 of printed output from program GRID

```
VALUES MULTIPLIED BY 10** 2
*******************************************************************************************************************
************************************************************************** 100*************************************
********************************************************************** 100 200 200 200******************************
********************************************************************** 100 200 300 300 300 300 200*******************
************************************************** 100 2 0 0 100******** 2 0 0 300 3 0 0 100 1 0 0 300*******************
*********************************************** 200 300 300 300 200 100 200 300 200 100 100 too 100 100****
******************************************** 200 300 400 400 400 300 200 200 200**********************************
******************************************** 300 400 400 500 400 400 300 200 100 100 100*********************
****************************300 300 4 0 0 4 0 0 500 5 0 0 5 0 0 'loo 400 100 100*****************************
********200********300 400 400 4 0 0 400 500 5 0 0 5 0 0 5 0 0 3 0 0 100 1 0 0 1 0 0 100*********************
**** 4 0 0 400 5 0 0 4 0 0 500 500 500 5 0 0 5 0 0 6 0 0 6 0 0 6 0 0 6 0 0 5 0 0*******************************
********300 300 3 0 0 2 0 0 4 0 0 'loo 5 0 0 5 0 0 500 5 0 0 5 0 0 5 0 0 500 4 0 0 100****************************
************200 300 400 400 'loo 4 0 0 5 0 0 500 6 0 0 5 0 0 5 0 0 5 0 0 4 0 0 2 0 0 100***********************
************ 100 300 300 400 500 500 500 500 500 600 500 500 300 100***************************
************ 1 0 0 2 0 0 300 4 0 0 5 0 0 5 0 0 500 500 5 0 0 5 0 0 5 0 0 5 0 0 4 0 0 100 1 0 0 100*********************
**************** 200 300 400 5 0 0 500 5 0 0 5 0 0 5 0 0 500 500 5 0 0 4 0 0 2 0 0 1 0 0 100*********************
**********************300 400 500 500 500 5 0 0 5 0 0 5 0 0 500 5 0 0 5 0 0 3 0 0 1 0 0 1 0 0 100*****************
****************************300 400 4 0 0 400 'loo 5 0 0 5 0 0 5 0 0 5 0 0 4 0 0 4 0 0 1 0 0 1 0 0 1 0 0 too************
******************************** 3 0 0 4 0 0 'loo 4 0 0 500 5 0 0 500 5 0 0 5 0 0 3 0 0 3 0 0 100 1 0 0 100************
************************************ 3 0 0 4 0 0 400 4 0 0 4 0 0 5 0 0 5 0 0 5 0 0 4 0 0 3 0 0 1 0 0 1 0 0 1 0 0 100********
**************************************** 2 0 0 3 0 0 3 0 0 4 0 0 4 0 0 500 5 0 0 'loo 3 0 0 3 0 0 200*****************
******************************************** 100 1 0 0 3 0 0 4 0 0 4 0 0 5 0 0 4 0 0 3 0 0 300**********************
************************************************** 100 3 0 0 4 0 0 4 0 0 4 0 0 3 0 0 200**********************
************************************************** 100 200 300 300 300 100*************************
****************************************************** 100 100 200********************************
*******************************************************************************************************************
```


Page 6 of printed output from program GRID

LAKE ST. CLAIR BATHYMETRY

```
 24    26   42.3041534    82.9319796  2000     6      1  0.0    .0     14 45.00
 0.822690E+02  -0.418687E+01  =0.892958E+00   0.549244E+00
 0.284351E+01   0.110232E+03   0.182918E+00   0.235336E+00
 0.121387E-01   0.462637E-03   0.110856E-05  -0.102938E-05
-0.313049E-03   0.905963E-02  -0.238882E-06  -0.279918E-06
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	3	3	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	3	4	4	4	3	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	3	4	4	5	4	3
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	3	4
4	5	5	4	3	3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	4	4	4	4	5	5	5	4	3	1	1	1	1	0	0	0	0	0
0	0	0	0	3	4	4	4	5	5	5	5	5	3	3	1	1	1	0
0	0	0	0	0	0	0	0	3	4	4	4	4	5	5	5	5	4	4
1	1	1	1	0	0	0	0	0	0	0	0	3	4	5	5	5	5	5
5	5	5	5	3	1	1	1	0	0	0	0	0	0	0	0	2	3	4
5	5	5	5	5	5	5	5	4	2	1	1	0	0	0	0	0	0	0
0	1	2	3	4	5	5	5	5	5	5	5	5	4	,	1	1	0	0
0	0	0	0	0	0	1	3	3	4	5	5	5	5	5	6	5	5	3
1	0	0	0	0	0	0	0	0	0	0	2	3	4	4	4	4	5	5
6	5	5	5	4	2	1	0	0	0	0	0	0	0	0	3	3	3	2
4	4	5	5	5	5	5	5	5	4	1	0	0	0	0	0	0	0	0
4	4	5	4	5	5	5	5	5	6	6	6	6	5	0	0	0	0	0
0	0	0	0	0	0	2	0	0	3	4	4	4	4	5	5	5	5	3
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	3	3	4	4
5	5	5	4	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	3	4	4	5	4	4	3	2	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	2	3	4	4	4	3	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	3	3	2
1	2	3	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	2	1	0	0	2	3	3	1	1	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	2	3	3	3	3	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Output from program GRID on LUN 8

```
     150.          5.          1
```

Input for program FRSFC on LUN 5

```
LAKE ST. CLAIR BATHYMETRY
  24    26   42.3041534  92.9315796  2000     6     1  0.0     0     14 45.00
  0.822690E+02   -0.418687E+01   -0.892958E+00    0.549244E+00
  0.284351E+01    0.110232E+03 0  .182918E+00    0.235336E+00
  0.121387E-01    0.462637E-03    0.110856E-05   -0.102938E-05
 -0.313049E-03    0.905963E-02   -0.238882E-06   -0.279918E-06
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
 .  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  1  2  3  3  3  1  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  1  3  4  4  4  3  2  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  1  1  3  4  4  5  4  3
 3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  3  4
 4  5  5  4  3  3  2  0  0  0  0  0  0  0  0  0  0  0  0
 3  4  4  4  4  5  5  5  4  3  1  1  1  1  0  0  0  0  0
 0  0  0  0  3  4  4  4  5  5  5  5  5  3  3  1  1  1  0
 0  0  0  0  0  0  0  0  3  4  4  4  4  5  5  5  5  4  'I
 1  1  1  1  0  0  0  0  0  0  0  0  3  4  5  5  5  5  5
 5  5  5  5  3  1  1  1  0  0  0  0  0  0  0  0  2  3  4
 5  5  5  5  5  5  5  5  4  2  1  1  0  0  0  0  0  0  0
 0  1  2  3  4  5  5  5  5  5  5  5  5  4  1  1  1  0  0
 0  0  0  0  0  0  1  3  3  4  5  5  5  5  5  6  5  5  3
 1  0  0  0  0  0  0  0  0  0  0  2  3  4  4  4  4  5  5
 6  5  5  5  4  2  1  0  0  0  0  0  0  0  0  3  3  3  2
 4  4  5  5  5  5  5  5  5  4  ,  0  0  0  0  0  0  0  0
 4  4  5  4  5  5  5  5  5  6  6  6  6  5  0  0  0  0  0
 0  0  0  0  0  0  2  0  0  3  4  4  4  4  5  5  5  5  3
 1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  3  3  4  4
 5  5  5  4  4  1  1  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  3  4  4  5  4  4  3  2  1  1  1  0  0  0  0  0
 0  0  0  0  0  0  0  0  2  3  4  4  4  3  2  2  2  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  3  3  2
 1  2  3  2  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0
 0  0  1  2  1  0  0  2  3  3  1  1  3  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  1  2  3  3  3  3  2  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  2
 2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Input for program FRSFC on LUN 7

```
     0        42.5     82.75     10.        4        4.       10       270
    -1
```

Input for program FRSFC on LUN 8

```
      SUBROUTINE INIT(D,H,AM, AN,IDIM)
C
C SUBROUTINE TO SET INITIAL CONDITIONS
C
      WRITE(6,1)
    1 FORMAT(1H1)
      RETURN
      END

      SUBROUTINE OUTP(TIME,D,H,AM. AN.IDIM)
C
C SUBROUTINE TO PRINT FREESURFACE FLUCTUATION FIELD AT LAST TIMESTEP
C
      DIMENSION H(IDIM,1)
      COMMON /CPARM/ DT, TT, DADD
      COMMON /GPARM/RPARM(23),IPARM(54)
      IF (TIME .LT. TT*3600.) RETURN
      IM = IPARM(1)
      JM = IPARM(2)
      CALL PRNT(6,H, IDIM, IM, JM, O.)
      RETURN
      END
```

User-supplied subroutines for program FRSFC

```
FREE SURFACE CIRCULATON MODEL FOR LAKE ST. CLAIR BATHYMETRY
TIME STEP(S):DT= 150.00
DURATION OF RUN(H):TT=      5.00
MEAN WATER LEVEL(M), (RELATIVE TO L.W.D):DADD= 1.00
NUMBER OF TIME STEPS: NSTEPS=     120
CORIOLIS PARAMETER (S**-1):F=0.984E-04
MAXIMUM TIMESTEP FOR LONG GRAVITY WAVE (S):DTSTAB=     170.75

          GRID DESCRIPTION FOR LAKE ST. CLAIR BATHYMETRY

*DIMENSION : IPARM(1)=       24       JDIMENSION  :  IPARM(2) =          26

              DISPLACEMENT OF ORIGIN IN NUMBER OF GRID SQUARES

IDISPLACEMENT  :  IPARM(3) =    0     JDISPLACEMENT  :  IPARM(4) =       14

BASE LATITUDE  :  RPARM(1) =   42.3041534

BASE LONGITUDE  :  RPARM(2) =  82.9315796

GRID SIZE(M):RPARM(3)= 2000.      MAX DEPTH(M):RPARM(4)=              6.

MIN DEPTH(M):RPARM(5)=     1.     BASE ROTATION  :  RPARM(6) =        0.0

ANGLE ROTATED :RPARM(7)=45

          GEOGRAPHIC-TO-MAP  COORDINATE  CONVERSION  COEFFICENTS FOR X

RPARM(8) =            0.822690E+02     RPARM(9) =              -0.418687E+01

RPARM(10) =          -0.892958E+00     RPARM(11) =              0.549244E+00

          GEOGRAPHIC-TO-MAP  COORDINATE  CONVERSION  COEFFICIENTS  FOR  Y

RPARM(12)=            0.284351E+01     WARM, 13)=               0.110232E+03

RPARM(14)=            0.182918E+00     WARM, 15)=               0.235336E+00

        MAP-TO-GEOGRAPHIC  COORDINATE  CONVERSION  COEFFICIENTS  FOR LONGITUDE

RPARM, 16)=           0.121387E-01     RPARM(17) =              0.462637E-03

RPARM(18) =           0.110856E-05     RPARM(19) =             -0.102938E-05

        MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LATITUDE .

RPARM(20) =          -0.313049E-03     RPARM(21) =              0.905963E-02

RPARM(22) =          -0.238882E-06     RPARM(23) =             -0.279918E-06
```

Page 1 of printed output from program FRSFC

```
VALUES MULTIPLIED BY 10** 2
*************************************************************************************************************
******************************************************************************** 200***************************
**************************************************************************** 200 300 300 300*******************
**************************************************************************** 200 3  0  0 400 400 400 400 300********
************************************************************* 200 300 200******** 300 400 400 200 200 400********
********************|*****.********** 300 400 400 400 3 0 0 200 300 400 300 200 200 200 200 200****
********************************** 300 'loo 500 500 500 400 300 300 300***************************
• [illegible] 400 500 500 600 500 500 400 300 200 200 200*******************
************************** 400 [illegible] 500 [illegible] 500 200 200*************************
********300******** 'loo 500 500 500 500 600 600 600 600 400 200 200 200 200******************
**** 500 500 600 500 600 600 600 600 600 700 700 700 700 600*********************************
******** 400 400 400 300 500 500 600 600 600 600 600 600 600 500 200**************************
************ 300 400 500 500 500 500 600 600 700 600 600 600 500 300 200**********************
************ 200 400 400 500 600 600 600 600 600 700 600 600 400 200*************************
•   *********** 200 300 400 500 600 600 600 600 600 600 600 600 500 200 200 200*******************
**************** 300 'loo 500 600 600 600 600 600 600 600 600 500 300 200 200******************
******************** 400 500 600 600 600 600 600 600 600 600 600 'loo 200 200 200****************
• [illegible] 400 [illegible] 600 500 500 200 200 200 200************
***************************** [illegible] 600 600 400 400 200 200 200• [illegible]
******************************** 400 500 500 500 500 600[illegible] 200********
************************************** 300 'loo 400 500 500 600 600 500 400 'loo 300****************
************************************ 200 200 400 500 500 600 500 400 400*******************
****************************************** 200 400 500 500 500 400 300*******************
****************************************** 200 300 400 400 400 200**********************
********************************************* 200 200 300*******************************
*************************************************************************************************************
```

71

Page 2 of printed output from program FRSFC

```
********************************************************************************************************
TIME  •  LATITUDE *  LONGITUDE *Z   *  [illegible]  * T-H2O • W-SPD  *   [illegible]   * CDE3 * X * Y
********************************************************************************************************
[illegible] • [illegible] • [illegible] * [illegible] * [illegible] * [illegible] • [illegible] • [illegible] *  1.29   *  26.  *   34.
```


Page 3 of printed output from program FRSFC

```
VALUES MULTIPLIED BY10** 3
************************************************************************************************************************
******************************************************************************** -19**********************************
************************************************************************** -17 -10 -4   3*************************
************************************************************************** -5 -     3 2  9  15  23  33**********
******************************************************* -54 -46 -39********** -1 6 1  3  2 ,  3 ,  40**********
********************************************* -61 -54 -47 -39 -32 -25 -11  0  10  22  35  45  51  64****
**************************************** -65 -57 -50 -42 -35 -28 -20 -10  -2************************************
**************************************** -61 -54 -46 -39 -32 -26 -19 -11 -3  10  23*************************
****************************** -74 -67 -58 -51 -43 -36 -3, -25 -20 -14  -7************************************
******** -95******** -77 -70 -62 -55 -47 -40 -33 -28 -23 -18 -12 -6  -9  -7*************************
**** -96 -91 -86 -80 -73 -66 -59 -51 -43 -37 -30 -25 -20 -15*********************************************
******** -87 -82 -75 -68 -6, -54 -47 -40 -33 -26 -20 -15 -10 -4   0***********************************
************ -74 -67 -61 -55 -47 -40 -34 -27 -21 -15  -9  -4   2   7  12******************************
************ -66 -60 -54 -47 -40 -34 -28 -23 -16 -10 -4  2   8  14***********************************
************ -56 -51 -45 -39 -33 -27 -22 -17 -12 -5  2  8  15  21  29  40*************************
**************** -41 -36 -31 -26 -21 -16 -12 -6  1   7  14  22  30  38  46*************************
******************** -30 -25 -20 -15 -11 -5  ,  7  14   2,  29  37  46  54  61********************
************************ -17 -12 -8 -4  ,  7  14  21  29  37  44  52  62  72  84***************
**************************** -5  0  4   9  14  21  28  36  44  53  61  68  80  93***************
******************************** ,  12  16  21  29  34  43  51  60  67  75  87  102 113**********
************************************ 19  23  28  35  41  49  58  66  74  81  go*********,******
**************************************** 3  ,  37  42  48  56  64  72  79  85*************************
************************************************ 49  55  62  70  77  84  91*************************
************************************************ 57  64  70  78  84  92******************************
**************************************************** 75 83  go*****************************.*
************************************************************************************************************************
```

Page 4 of printed output from program FRSFC

```
          1.        24.         1

LAKE ST. CLAIR BATHYMETRY
  24    26  42.3041534  82.9315796  2000   6   1  0.0   0   14 45.00
  0.822690E+02  -0.418687E+01  -0.892958E+00  0.549244E+00
  0.284351E+01  -0.110232E+03   0.182918E+00  0.235336E+00
  0.121387E-01   0.462637E-03   0.110856E-05 -0.102938E-05
 -0.313049E-03   0.905963E-02  -0.238882E-06 -0.279918E-06
  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  1
  1  2  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  1  2  3  3  3  1  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  1  3  4  4  4  3  2  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  1  1  3  4  4  5  4  3
  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  3  4
  4  5  5  4  3  3  2  0  0  0  0  0  0  0  0  0  0  0  0
  3  4  4  4  4  5  5  5  4  3  1  1  1  1  0  0  0  0  0
  0  0  0  0  3  4  4  4  5  5  5  5  5  3  3  1  1  1  0
  0  0  0  0  0  0  0  0  3  4  4  4  4  5  5  5' 5  4  4
     1  1  1  0  0  0  0  0  0  0  0  3  4  5  5  5  5  5
  5  5  5  5  3  1  1  1  0  0  0  0  0  0  0  0  2  3  4
  5  5  5  5  5  5  5  5  4  2  1  1  0  0  0  0  0  0  0
  0  1  2  3  4  5  5  5  5  5  5  5  5  4  1  1  1  0  0
  0  0  0  0  0  0  1  3  3  4  5  5  5  5  5  6  5  5  3
  1  0  0  0  0  0  0  0  0  0  0  2  3  4  4  4  4  5  5
  6  5  5  5  4  2  1  0  0  0  0  0  0  0  0  3  3  3  2
  4  4  5  5  5  5  5  5  5  4  1  0  0  0  0  0  0  0  0
  4  4  5  4  5  5  5  5  5  6  6  6  6  5  0  0  0  0  0
  0  0  0  0  0  0  2  0  0  3  4  4  4  4  5  5  5  5  3
  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  3  3  4  4
  5  5  5  4  4  1  1  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  3  4  4  5  4  4  3  2  1  1  1  0  0  0  0  0
  0  0  0  0  0  0  0  0  2  3  4  4  4  3  2  2  2  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  3  3  2
  1  2  3  2  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0
  0  0  1  2  1  0  0  2  3  3  1  1  3  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  1  2  3  3  3  3  2  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  2
  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

     0.      42.5     82.75     10.     4      4.     10     270.
    -1.
```

}Input for program RLID on LUN 5

Input for program RLID on LUN 7

}Input for program RLID on LUN 8

User-supplied subroutines for program RLID

```
      SUBROUTINE INIT (D, S, IDIM)
C
C  SUBROUTINE TO SET INITIAL CONDITIONS
C
      WRITE (6, 1)
    1 FORMAT(1H1)
      RETURN
      END
      SUBROUTINE OUTP (TIME, D, S, SPD, IDIM)
C
C  SUBROUTINE TO PRINT FREESURFACE FLUCTUATION FIELD AT LAST TIMESTEP
C
      DIMENSION S(IDIM,1)
      COMMON /CPARM/ DT, TT, DADD
      COMMON /GPARM/ RPARM(23), IPARM(54)
      IF (TIME .LT. TT*3600 ) RETURN
      IM = IPARM(1)
      JM = IPARM(2)
      CALL PRNT(6, S, IDIM, IM  JM, 0.)
      RETURN
      END
```

```
RIGIDLID CIRCULATION MODEL FOR LAKE ST. CLAIRBATHYMETRY
TIME STEP(H):DT=        1.00
DURATION OF RUN(H):TT=    24.00
MEAN WATER LEVEL(M) (RELATIVE TO L.W.D.):DADD= 1.00
NUMBER OF TIME STEPS! NSTEPS=     24
CORIOLIS PARAMETER(S**-1):F=0.984E-04

        GRID DESCRIPTION FOR LAKE ST. CLAIRBATHYMETRY

IDIMENSION  :  IPARM(1) =        24        JDIMENSION  :  IPARM(2) =          26

            DISPLACEMENT OF ORIGIN IN NUMBER OF GRIDSQUARES

IDISPLACEMENT  :  IPARM(3) =    0        JDISPLACEMENT  :  IPARM(4) =         14

BASE LATITUDE :RPARM(1)=    42.3041534

BASE LONGITUDE :RPARM(2)=82.9315796

GRID SIZE(M):RPARM(3)=2000.       MAX DEPTH(M):RPARM(4)=          6.

MIN DEPTH(M):RPARM(5)=     1.       BASE ROTATION :RPARM(6)=         0.0

ANGLE ROTATED :RPARM(7)=45.

         GEOGRAPHIC-TO-MAP  COORDINATE  CONVERSION  COEFFICENTS FOR X

RPARM(8) =            0.822690E+02      RPARM(9) =              -0.418687E+01

RPARM(10) =          -0.892958E+00      RPARM(11) =              0.549244E+00

         GEOGRAPHIC-TO-MAP  COORDINATE  CONVERSION  COEFFICIENTS  FOR  Y

RPARM(12) =           0.284351E+01      RPARM( 13) =             0.110232E+03

RPARM( 14) =          0.182918E+00      RPARM( 15) =             0.235336E+00

       MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LONGITUDE

RPARM(16) =           0.121387E-01      RPARM(17) =              0.462637E-03

RPARM(18) =           0.110856E-05      RPARM( 19) =            -0.102938E-05

       MAP-TO-GEOGRAPHIC  COORDINATE  CONVERSION  COEFFICIENTS  FOR  LATITUDE

RPARM(20) =          -0.313049E-03      RPARM(21) =              0.905963E-02

RPARM(22) =          -0.238882E-06      RPARM(23) =             -0.279918E-06
```

Page 1 of printed output from program RLID

VALUES MULTIPLIED BY 10** 2

180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180
180	1.90	180	180	180	180	180	180	180	180	180	180	180	180	180	250	325	350	180	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	350	400	350	300	325	180	180	180
180	180	180	180	180	180	180	180	180	180	180	325	300	180	180	180	350	325	250	200	250	180	180	180
180	180	180	180	180	180	180	180	180	‘loo	450	450	400	300	275	325	180	180	180	180	180	180	180	180
180	180	180	180	180	180	180	180	400	475	525	525	475	‘loo	325	275	180	180	180	180	180	180	180	180
180	180	180	180	180	180	180	180	475	525	575	575	525	475	350	225	180	180	180	180	180	180	180	180
180	180	180	180	180	180	450	475	500	550	600	600	575	500	325	200	180	180	180	180	180	180	180	180
180	180	180	180	180	525	550	550	550	600	650	650	650	575	180	180	180	180	180	180	180	180	180	180
180	180	475	475	450	500	550	575	600	625	650	650	650	625	180	180	180	180	180	180	180	180	180	180
180	180	180	375	‘loo	450	500	525	575	600	625	625	600	600	550	375	180	180	180	180	190	180	180	180
180	180	180	325	425	475	525	550	575	600	625	650	625	600	525	350	180	180	180	180	180	180	180	180
180	180	180	275	375	450	550	600	600	600	600	625	625	600	525	325	180	180	180	180	180	180	180	180
180	180	180	180	350	450	550	600	600	600	600	600	600	600	550	375	225	200	180	180	180	180	180	180
180	180	180	180	180	450	550	600	600	600	600	600	600	600	575	450	275	200	180	180	180	180	180	180
180	180	180	180	180	180	500	550	550	550	575	600	600	600	575	500	325	200	200	180	180	180	180	180
180	180	180	180	180	180	180	475	500	500	550	600	600	600	575	500	375	250	200	200	180	180	180	180
180	180	180	180	180	180	180	180	475	500	525	550	575	600	600	525	425	300	200	200	180	180	180	180
180	180	180	180	180	180	180	180	180	425	450	475	525	575	600	550	450	350	275	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	300	375	475	525	575	550	450	400	180	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	180	180	375	475	525	525	450	375	180	180	180	180	180	180
180	180	180	180	180	180	190	180	180	180	180	180	275	400	450	450	375	180	180	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	180	180	180	275	325	180	180	180	180	180	180	180	180	180
180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180	180

Page 2 of printed output from program RLID

```
*******************************************************************************************
TIME  *  LATITUDE  *  LONGITUDE *  Z   * T-AIR * T-H2O * W-SPD  * W-DIR * CDE3 *   X   *   Y
*******************************************************************************************
  0.0 * 42.5000000 * 82.7500000 * 10.0 *  4.00 *  4.00 *  10.00 * 270.  *  1.29 *  26.  *  34.
```


Page 3 of printed output from program RLID