I.O.S.

USER'S MANUAL FOR STEEP AND BREAKING WAVE COMPUTER PROGRAMS

E. D. Cokelet

Report No. 87

1979

NATURAL ENVIRONMENT RESEARCH COUNCIL
INSTITUTE OF
OCEANOGRAPHIC
SCIENCES

INSTITUTE OF OCEANOGRAPHIC SCIENCES

Wormley, Godalming,
Surrey, GU8 5UB.
(0428 - 79 - 4141)

(Director: Dr. A.S. Laughton)

Bidston Observatory,
Birkenhead,
Merseyside, L43 7RA.
(051 - 653 - 8633)

(Assistant Director: Dr. D.E. Cartwright)

Crossway,
Taunton,
Somerset, TA1 2DW.
(0823 - 86211)

(Assistant Director: M.J. Tucker

On citing this report in a bibliography the reference should be followed by the words UNPUBLISHED MANUSCRIPT.

Institute of Oceanographic Sciences

USER'S MANUAL FOR STEEP AND BREAKING WAVE COMPUTER PROGRAMS

*E. D. Cokelet

Report No. 87

1979

Institute of Oceanographic Sciences,
Wormley, Godalming,
Surrey GU8 5UB U. K.

*Present address:

Pacific Marine Environmental Laboratory/NOAA
3711 15th Avenue Northeast,
Seattle, Washington 98105, U. S. A.

CONTENTS

I. General Introduction

This manual contains information about computer programs and magnetic tapes at IOS which were the results of work done on steep and breaking water waves by E. D. Cokelet between November 1975 and November 1978. The programs fall into three main groups: (1) those which calculate the velocity and acceleration field within the interior of a breaking wave, (2) those which calculate the properties of steep, steady waves, and (3) miscellaneous numerical and graphical routines which may be of use to general IOS computer users.

In all of the theoretical and numerical work about water waves referred to herein the physical variables are nondimensionalized such that $\rho = g = k = 1$ where ρ is the density of the fluid, g is the acceleration of gravity and $k = 2\pi/\lambda$ where λ is the wavelength of the free surface. λ is the horizontal scale on which the two-dimensional flow repeats itself. For the experiments on the stability of deep-water waves as discussed in section II, two waves of generally unequal wavelengths are included within the distance λ.

II. INTVEL: Velocities and accelerations within a breaking wave.

II.A. Introduction

INTVEL is a FORTRAN program which calculates the fluid velocities and accelerations interior to a breaking wave in deep water given quantities evaluated only on the free surface. These surface-evaluated quantities are actually stored as data on a magnetic tape (M907 at IOS, backup tape M908). This tape contains output from various numerical experiments performed on Cambridge University's IBM 370/165 computer and on the Rutherford High Energy Laboratory's (RHEL) IBM 360/195 computer by E. D. Cokelet between 1975 and 1978. The accompanying pre-print in section II.B. (Cokelet, E. D. 1979 Breaking waves - the plunging jet and interior flow-field, Proc. Symp. Mech. Wave-induced Forces on Cylinders, 3-6 September 1978, University of Bristol) explains briefly how the velocity and acceleration fields are calculated and gives some examples of the graphical output from INTVEL. Results of some of the numerical experiments (but not the interior flow fields) appear in two papers by M. S. Longuet-Higgins and E. D. Cokelet (The deformation of steep surface waves on water I. A numerical method of computation, 1978, Proc. R. Soc. Lond. A 350, 1-26 and II. Growth of normal-mode instabilities, 1979, Proc. R. Soc. Lond. A 354, 1-28).

A conformal transformation is used to map the flow from the physical

$z\,(= x + iy)$ plane to a closed region in the

$\zeta\,(= re^{i\theta})$ plane. The mapping is $\zeta = e^{-iz}$

which means that

$$\theta = -x$$
$$r = e^{y}.$$

In INTVEL most of the numerical calculations are done in the ζ-plane, but the graphical output consisting of the wave profile and the velocity and acceleration vectors are in the z-plane. There are usually N = 60, 90 or 120 points along the surface of one wave cycle. The storage arrays and their physical meanings are as follows: The array sizes are all N + 1 unless stated otherwise.

R	r values along the surface
PHI	velocity potential, ϕ, along surface
PRES	surface pressure
DPHIDS	$\frac{\partial\phi}{\partial s}$ = tangential derivative in ζ-plane
DPHIDN	$\frac{\partial\phi}{\partial n}$ = normal derivative
DSDP	$\frac{ds}{dp}$ = derivative of arc-length with respect to a parameter p (point number) which increases monotonically along the surface profile
S(N + 8)	S = arc - length along surface in ζ-plane. S(N)- S(0)= S(N) which is the total arc length around one wave cycle in the ζ -plane. For convenience $S(N+j) = S(N) + S(j)$, $j = 1, 2, \dots, 8$.
DTHDS	$\frac{d\theta}{ds}$ along the surface
DRDS	$\frac{dr}{ds}$ along the surface
DTHDT	$\frac{D\theta}{Dt}$ total time derivative of θ.

D2PDNS	$\frac{\partial^2\phi}{\partial n\,\partial s}$ in the ζ-plane
D2RDS2	$\frac{d^2 r}{d s^2}$ in the ζ-plane
D2TDS2	$\frac{d^2\theta}{d s^2}$ in the ζ-plane
PHIT	$\frac{\partial\phi}{\partial t}$
PSIT	$\frac{\partial\psi}{\partial t}$
PRESO (IE,JE)	2-D array of pressure inside the fluid in the Z-plane. The array sizes IE and JE depend on the size of the region and the grid spacing on which one wishes to calculate the flow quantities.

II.B. Breaking waves - the plunging jet and interior flow-field.

The following is a preprint to appear in the *Proceedings of the Symposium on Mechanics of Wave-induced Forces on Cylinders*, held at the University of Bristol, 3-6 September 1978. In it we describe briefly the mathematical theory behind INTVEL, and we present some of the graphical results.

To appear in Proceedings of the Symposium on Mechanics of Wave-induced Forces on Cylinders, held at the University of Bristol, 3-6 September 1978.

BREAKING WAVES - THE PLUNGING JET AND INTERIOR FLOW-FIELD

by

E.D. Cokelet

Institute of Oceanographic Sciences,
Brook Road, Wormley, Godalming,
Surrey, England.

Abstract

Breaking waves cause some of the largest forces exerted on marine structures, but until quite recently it has not been possible to calculate the flow-field in a breaking wave even in the absence of the structure itself. Our aim is to show that during breaking but before vorticity and turbulence are generated the flow can be calculated by using a numerical technique (Longuet-Higgins & Cokelet, 1976) based on the exact theory of irrotational flow. This has succeeded in following the free surface of a two-dimensional, deep-water wave as it steepens and overturns ejecting a jet of water from its crest. We present some examples to show that the fluid velocities and accelerations can exceed those in a highest, steady wave.

Introduction

For a structural member whose diameter is less than about 1/5 of a wavelength the usual method for predicting forces is to take the statistically derived 50-year wave's height and period, to fit a wave flow to these parameters, and to calculate the drag and inertia from Morison's equation.

All three steps have their disadvantages, but we shall concentrate on the second step. Often a steady, progressive wave is chosen, and the flow-field is calculated by using an analytical (e.g. Stokes' n^{th} order expansion) or numerical method (e.g. Dean's (1965) stream function wave theory). This presupposes that the wave is symmetric about the crest and propagates without change of shape; hence it is not a breaking wave. Sometimes a measured wave profile is available, and a flow-field is fitted to it with Dean's stream function method in which it is assumed that the free-surface is a streamline when viewed from an appropriate moving reference frame. Such an assumption is not valid if the wave is asymmetrical or if it is breaking.

Breaking is essentially time-dependent; the free surface distorts as fluid moves normal to it. Longuet-Higgins & Cokelet (1976) have developed a numerical technique capable of following the fully nonlinear development of the flow. The method is based on the fact that for an irrotational, incompressible fluid the kinematic and dynamic boundary conditions describe the evolution of the flow in terms of quantities specified only on the boundaries. For deep-water waves the only boundary is the free surface, and its location is given by the position of N specified fluid particles. In order to follow the motion of the surface we need know only the tangential and normal velocities of particles on it. Given the velocity potential, ϕ, along the surface we find the tangential velocity, $\partial\phi/\partial s$, by numerical differentiation. To find the normal velocity, $\partial\phi/\partial n$, we use Green's third identity to derive an integral equation for it in terms of other surface-evaluated quantities. Having solved this numerically we march forward in time with the evolution equations:

$$\frac{Dx}{Dt} = \frac{\partial\phi}{\partial x} = \frac{dx}{ds}\frac{\partial\phi}{\partial s} + \frac{dy}{ds}\frac{\partial\phi}{\partial n}$$

$$\frac{Dy}{Dt} = \frac{\partial\phi}{\partial y} = \frac{dy}{ds}\frac{\partial\phi}{\partial s} - \frac{dx}{ds}\frac{\partial\phi}{\partial n}$$

$$\frac{D\phi}{Dt} = \frac{1}{2}\left(\frac{\partial\phi}{\partial s}^2 + \frac{\partial\phi}{\partial n}^2\right) - gy - \frac{p}{\rho}$$

These specify how the Cartesian coordinates, (x,y), and the velocity potential of a fluid particle change with time. The tangential direction is defined such that s increases as one moves along the surface with the fluid on the left, and n is the outward normal.

The method has been used to demonstrate that irrotational flow alone can account for the steepening and plunging of a breaking wave. The plunging jet remains well-rounded and can be followed until either the minimum radius of curvature of the free surface approaches the computational particle spacing or until the tip of the jet is about to touch the wave's forward face.

In the next section we shall show how to calculate the fluid velocity and acceleration within the fluid interior given the surface-evaluated quantities used in the evolution equations. Then we shall give some examples of fluid velocities and accelerations near the crest of some breaking waves in deep water.

Calculation of the interior fluid velocity and acceleration

The free surface is perhaps the region of greatest interest since it is here that nonlinear effects are most noticeable. Here, too, the velocity takes on its extreme value since Laplace's equation is the governing field equation. But to the offshore engineer the flow in the fluid interior is also important. With his requirements in mind we have set out to calculate the interior velocities and accelerations.

We define a Cartesian coordinate system with x increasing horizontally, **y** increasing upwards and the line $y = 0$ coinciding with the mean free surface. For computational convenience we assume that the sea surface is horizontally periodic. (This restriction could be removed if we were willing to specify what was happening outside of the computational region, but this introduces further complication which will not be considered here.) We map the semi-infinite region between the free surface and the bottom into a closed region in the ζ-plane with the mapping

$$\zeta = e^{-iz}$$

where $z = x + iy$. This maps the undisturbed free surface into a circle.

Now for $W(\zeta)$ a function harmonic inside the region bounded by a closed contour C — the image of the free surface — Cauchy's theorem relates W at some interior point ζ_o to the integral of its value on the boundary weighted by the inverse of the distance from ζ_o to the boundary, i.e.

$$W(\zeta_o) = \frac{1}{2\pi i}\int_C \frac{W(\zeta)}{\zeta-\zeta_o}\,d\zeta. \qquad (1)$$

If we let W be the complex velocity, W = u-iv, we can use (1) to find the velocity inside the fluid once we know it on the surface.

There is a complication when the point ζ_o is near the boundary. The denominator of (1) becomes large, and any quadrature formula which we may use does not accurately represent the integrand in the region giving the largest contribution to the integral. This problem may be overcome by subtracting off the singularity and solving iteratively for $W(\zeta_o)$ (Swarztrauber, 1972). Thus we may write

$$W_{n+1}(\zeta_o) = \frac{1}{2\pi i}\int_C \frac{W(\zeta)-W_n(\zeta_o)}{\zeta-\zeta_o}\,d\zeta + W_n(\zeta_o) \qquad (2)$$

where the subscript "n" denotes the n^{th} iterate.

If we write (2) as

$$W_{n+1} = f(W_n)$$

then $f'(W_n) = 0$ as well as all the higher derivatives of f, and the theoretical convergence of (2) is guaranteed. In practice (2) usually converges to within 10^{-5} in 5 or 6 iterations when the quadrature is performed using Simpson's rule with 60 computational particles. Occasionally for points very near the boundary (2) diverges, but in this case a Taylor series extrapolation from the boundary usually suffices.

To find the fluid acceleration at an interior point in terms of surface-evaluated quantities is more complicated since the acceleration does not satisfy Laplace's equation. If we can calculate the pressure, p, then we can differentiate numerically with, say, a 3-point, centred-difference formula to find the pressure gradient and then the acceleration from the Euler equations. Bernoulli's equation,

$$\frac{p}{\rho} = -\frac{\partial\phi}{\partial t} - gy - \frac{1}{2}(u^2 + v^2),$$

gives p provided we can find $\partial\phi/\partial t$. This does satisfy Laplace's equation, and we can calculate $\partial\phi/\partial t$ at the surface from Bernoulli's equation since p is specified there. However to find $\partial\phi/\partial t$ at an interior point from Cauchy's theorem, (1) or (2), we need also to know its conjugate harmonic function $\partial\psi/\partial t$ where ψ is the stream function. This is found along the surface from

$$\psi = \int \frac{\partial\psi}{\partial s}\, ds + \text{const.} = \int \frac{\partial\phi}{\partial n}\, ds + \text{const.}$$

The constant is evaluated such that $\psi\,(x, -\infty, t) = 0$. $\partial\psi/\partial t$ is calculated for each computational particle from

$$\frac{\partial\psi}{\partial t} = \frac{D\psi}{Dt}$$

by using a 3-point, centred-difference formula in time. Since differentiations in both time and space are needed to calculate the accelerations they are necessarily less accurate than the velocities.

Results

We shall present some results for the interior velocities and accelerations of breaking waves produced from two different types of initial conditions. For the first type, breaking develops from a slightly perturbed train of fully nonlinear, periodic, steady waves. Longuet-Higgins (1978 a,b) has shown that such a wave train is unstable if the initial wave steepness, ak, falls within certain limits. The numerical experiments of Longuet-Higgins and Cokelet (1978) have confirmed the growth-rates predicted by the stability theory and have shown that the instabilities lead to breaking. We shall show the results for ak = 0.25. Such a wave is unstable. Perturbations grow such that the neighbouring wave loses energy, and the growing wave slowly gains energy. When the local steepness approaches 0.41 the perturbation grows rapidly and breaking ensues.

Figure 1 shows a close-up of the wave crest at the moment, $t = t_v$, the wave face becomes vertical. Dots on the free surface mark the computational fluid particles. Quantities are nondimensionalized such that $\rho = g = k = 1$ where ρ is the fluid density, g is gravity and k is the wave number. Figure 1a shows the velocity field, and the horizontal vector in the upper right-hand corner represents a velocity of magnitude $(g/k)^{\frac{1}{2}} = 1$ which equals the phase speed of a linear, deep-water wave. The maximum fluid velocity shown occurs at the crest and is 1.02. This compares with the maximum particle velocity of 1.0922 at the crest of the highest, steady wave (Cokelet, 1977). Such a theoretically highest wave ($ak = 0.4431$) has a sharp crest with enclosed angle of 120° located at $y = 0.5964$. The acceleration field is shown in figure 1b with the scaling vector of length $g = 1$ at the upper right. The maximum acceleration indicated has magnitude 0.6g. The acceleration at the crest of the highest, steady wave has magnitude 0.5g directed radially away from the crest (Longuet-Higgins, 1963), although recent evidence indicates that 0.39g may be the more relevant value for an almost-highest, steady wave with a slightly-rounded crest (Longuet-Higgins & Fox, 1977). Figure 2 shows the same wave at about 1/50 of a wave period later ($t = t_v + 1/8$). Here the maximum velocity is 1.09, and the maximum acceleration is 0.7g. We have not been able to calculate reliably the acceleration field for points nearer the free surface than those shown in the diagrams.

For the second type of initial condition we have conducted a series of numerical experiments in which the wave is sinusoidal in shape but of large amplitude, i.e.

$$\eta = a \sin (x - t)$$

$$\phi = a \cos (x - t)\, e^{-y}$$

at $t = 0$. Such a wave contains all of its energy in the first harmonic and therefore cannot be a finite-amplitude, steady wave. If the initial amplitude is large enough the wave will break. Figure 3 shows the velocity and acceleration fields for a wave of initial amplitude $ak = 0.31$. It contains about 2/3 of the energy, E, of the most energetic, steady wave, $E_{max} = 0.07403$. The wave face is just vertical. The similarity to the breaking wave in figure 1 which was produced by the more "natural" mechanism of instability is remarkable; the two figures almost coincide. The maximum velocity and acceleration in figure 3 are 1.09 and 0.7g, respectively, which exceed those of figure 1. For this wave it has been possible to follow the motion until $t = t_v + 1/4$. The velocities and accelerations are shown in figure 4. The peak values are 1.24 and 0.8g.

In figure 5 we show the results for similar initial conditions but with $ak = 0.38$ corresponding to a wave with $E \simeq E_{max}$. The radius of curvature at the crest is larger here, and more fluid will be swept into the plunging jet. The maximum velocity displayed is 0.93. This is lower than previous values because the grid of points at which the velocities were calculated for the diagram does not intersect at the maximum velocity region near the emerging jet. The maximum acceleration shown is 0.9g. This occurs as in all the previous examples not at the crest but rather in the region below the vertical waveface. Both the drag and the inertial forces in this region will have a strong horizontal component.

The velocities and accelerations are shown in figure 6 when $t = t_V + 3/4$. Here the plunging jet is well-formed. There is a substantial region of horizontal velocity feeding water into the jet, and there is an indication that the fluid is being accelerated toward the right due to large horizontal pressure gradients. The maximum velocity indicated is 1.33 in the jet which would produce a drag force of 1.5 times that of a highest, steady wave. The maximum acceleration shown in 1.6g which is about 3 times that of a highest, steady wave.

Conclusions

We have shown that the velocity and acceleration fields interior to irrotational, breaking waves can be calculated from quantities evaluated at the free surface. Sample calculations indicate that fluid speeds can exceed $1.3\ (g/k)^{\frac{1}{2}}$ in the breaking region. Maximum accelerations approach g and can have appreciable horizontal components. It is difficult to say how typical these values are, but similar results do come from two different types of initial conditions thus implying that a similarity solution may exist.

In future we plan to compare in detail the velocity and acceleration fields with those of steep, steady waves. It also remains to compare the numerical results with laboratory and field data.

This work was supported by a contract from the Departments of Energy and Industry.

References

Cokelet, E.D. 1977 Steep gravity waves in water of arbitrary uniform depth. Phil. Trans. R. Soc. Lond. A 286, 183-230.

Dean, R.G. 1965 Stream function representation of non-linear ocean waves. J.Geophys. Res. 70, 4561-4572.

Longuet-Higgins, M.S. 1963 The generation of capillary waves by steep gravity waves. J. Fluid Mech. 16, 138-159.

Longuet-Higgins, M.S. 1978a The instabilities of gravity waves of finite amplitude in deep water I. Superharmonics. Proc. R. Soc. Lond. A 360, 471-488.

Longuet-Higgins, M.S. 1978b The instabilities of gravity waves of finite amplitude in deep water II. Subharmonics. Proc. R. Soc. Lond. A 360, 489-505.

Longuet-Higgins, M.S. & Cokelet, E.D. 1976 The deformation of steep surface waves on water I. A numerical method of computation. Proc. R. Soc. Lond. A 350, 1-26.

Longuet-Higgins, M.S. & Cokelet, E.D. 1978 The deformation of steep surface waves on water II. Growth of normal-mode instabilities. Proc. R. Soc. Lond. A, in press.

Longuet-Higgins, M.S. & Fox, M.J.H. 1977 Theory of the almost-highest wave: the inner solution. J. Fluid Mech. 80, 721-741.

Swarztrauber, P.N. 1972 On the numerical solution of the Dirichlet problem for a region of general shape. SIAM J. Numer. Anal. 9, 300-306.

```
      IN = 2*(M-L-R)
      A = A + SIGMA(IN)*SIGMA(2)*ALPHA(IN,L+1)*ALPHA(2,R+1) - S1
  490 CONTINUE
      IN = 2*(M-L)
      D5 = D5 + SIGMA(IN)*SIGMA(2)*ALPHA(IN,1)*ALPHA(2,L+1) + A
     1  - 2.*SIGMA(IN+1)*ALPHA(IN+1,L+1)
  500 CONTINUE
  502 CONTINUE
      ALPHA(2,M) = (C4*(B5 - A5*B2/A2 - B3*D5/D3) - B4*(C5 - C3*D5/D3
     1  ))/(C4*(B1 - A1*B2/A2 - B3*D1/D3) - B4*(C1 - C3*D1/D3))
      ALPHA(NROW-1, NCOL+1-M) = (A5 - A1*ALPHA(2,M))/A2
      IF (2*M-1 .EQ. NORDER) GO TO 164
      ALPHA(3,M) = (D5 - D1*ALPHA(2,M))/D3
      ALPHA(NROW-2, NCOL+1-M) = (C5 - C3*D5/D3 - (C1 - C3*D1/D3)*
     1  ALPHA(2,M))/C4
C
C...CALCULATE ALPHA(NROW, NCOL-MO/2).
C
      M = MX
      A = 0.0
      DO 530 L=1,M
      IRE = M - L + 1
      S1 = 0.0
      DO 520 IR=1,IRE
      R = IR - 1
      S1 = S1 + ALPHA(L+1, R+1)*ALPHA(L+1, M-L-R+1)
  520 CONTINUE
      A = A + SIGMA(2*L+1)*S1
  530 CONTINUE
      ALPHA(NROW, NCOL-M) = A
C
  162 CONTINUE
  164 CONTINUE
C
C...COMPUTE THE GAMMA(L).
C
      LE = INT(FLOAT(NORDER)/2. + .01) + 1
      DO 324 L=1,LE
      GAMMA(L) = 0.0
      DO 300 KI = 1,L
      K = KI-1
      GAMMA(L) = GAMMA(L) + DELTA(L-K)*ALPHA(NROW, NCOL-K)
  300 CONTINUE
      IF (L .LT. 2) GO TO 322
      IRE = L-1
      DO 320 IR=1,IRE
      R = IR-1
      S1 = 0.0
      IME = R+1
      DO 310 IM=1,IME
      M = IM-1
      S1 = S1 + ALPHA(L-R, R-M+1)*ALPHA(NROW+1-L+R, NCOL-M)
  310 CONTINUE
      GAMMA(L) = GAMMA(L) - 2.*D(L-R)*S1/FLOAT(L-R-1)
  320 CONTINUE
  322 CONTINUE
  324 CONTINUE
      RETURN
      END
```

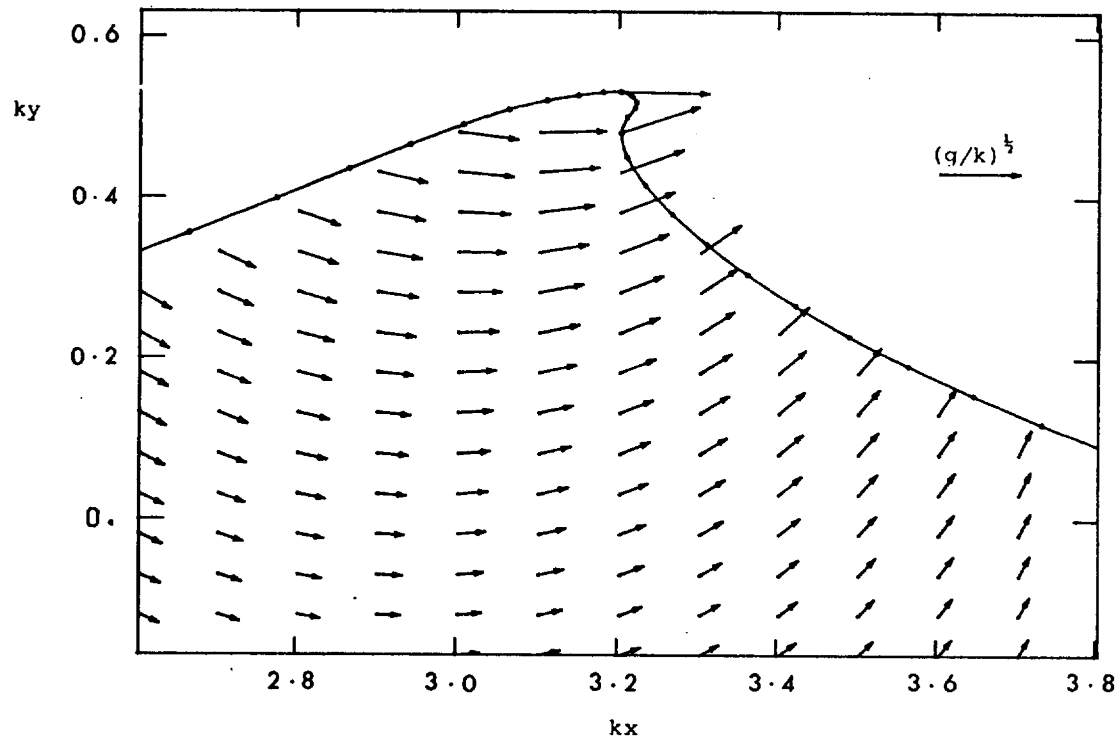

Figure 2a. As for figure 1a but with $t=t_v + 1/8$.

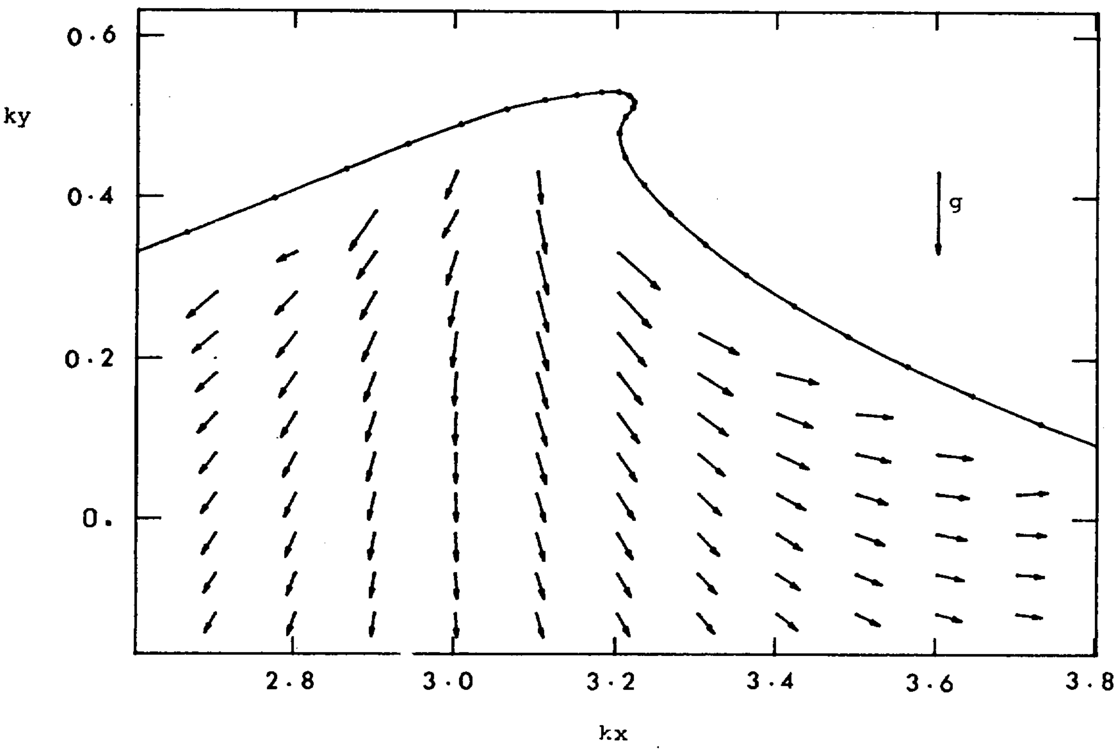

Figure 2b. As for figure 1b but with $t=t_v + 1/8$.

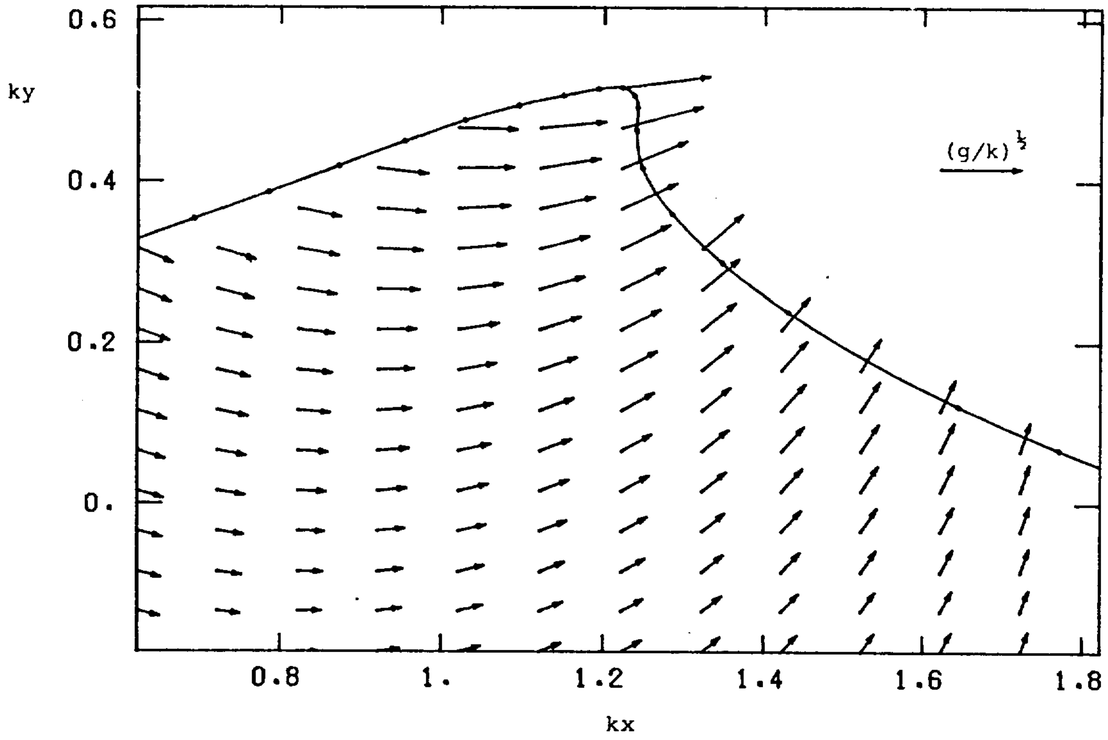

Figure 3a. The velocity field near the wave crest at the moment, $t=t_v$, when the wave face becomes vertical. This wave began as a simple sinusoid with $E \approx 2/3\ E_{max}$.

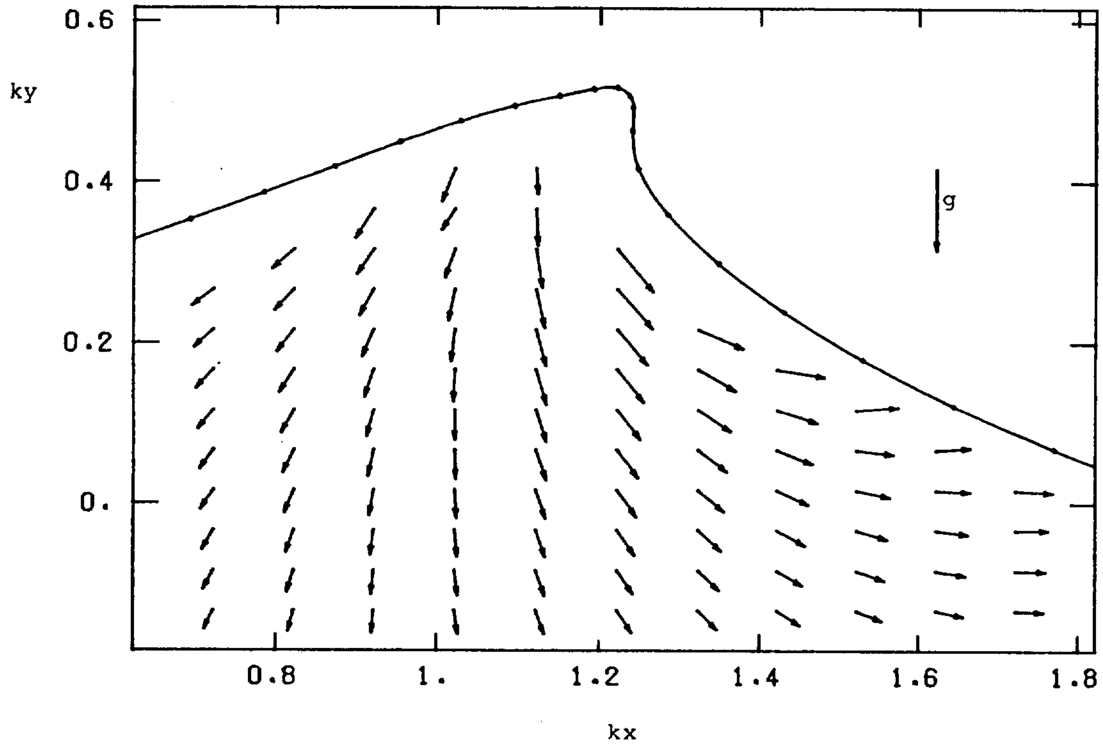

Figure 3b. The acceleration field near the wave crest at $t=t_v$.

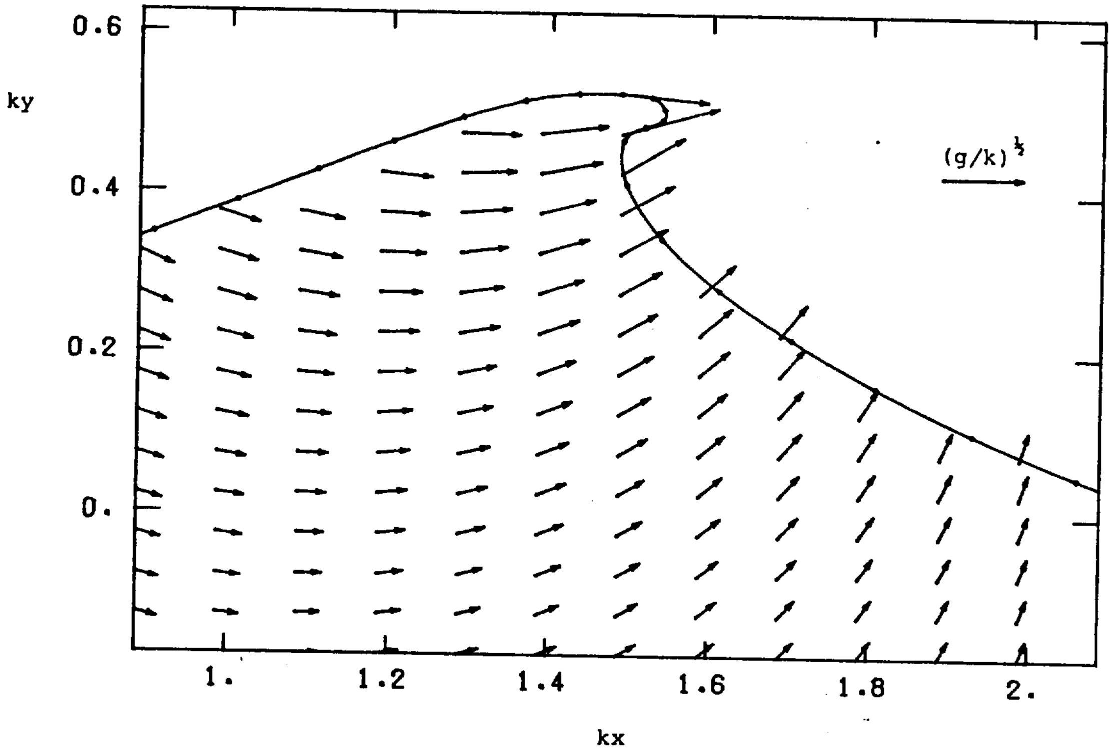

Figure 4a. As for figure 3a but with $t = t_v + 1/4$.

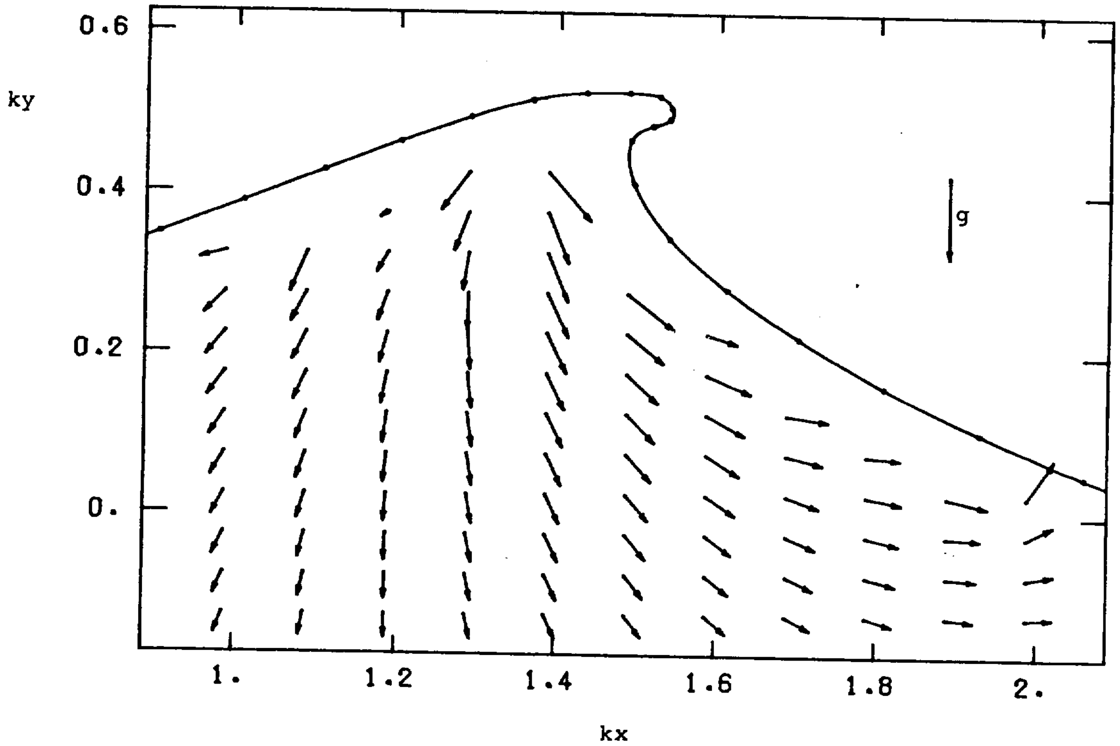

Figure 4b. As for figure 3b but with $t = t_v + 1/4$.

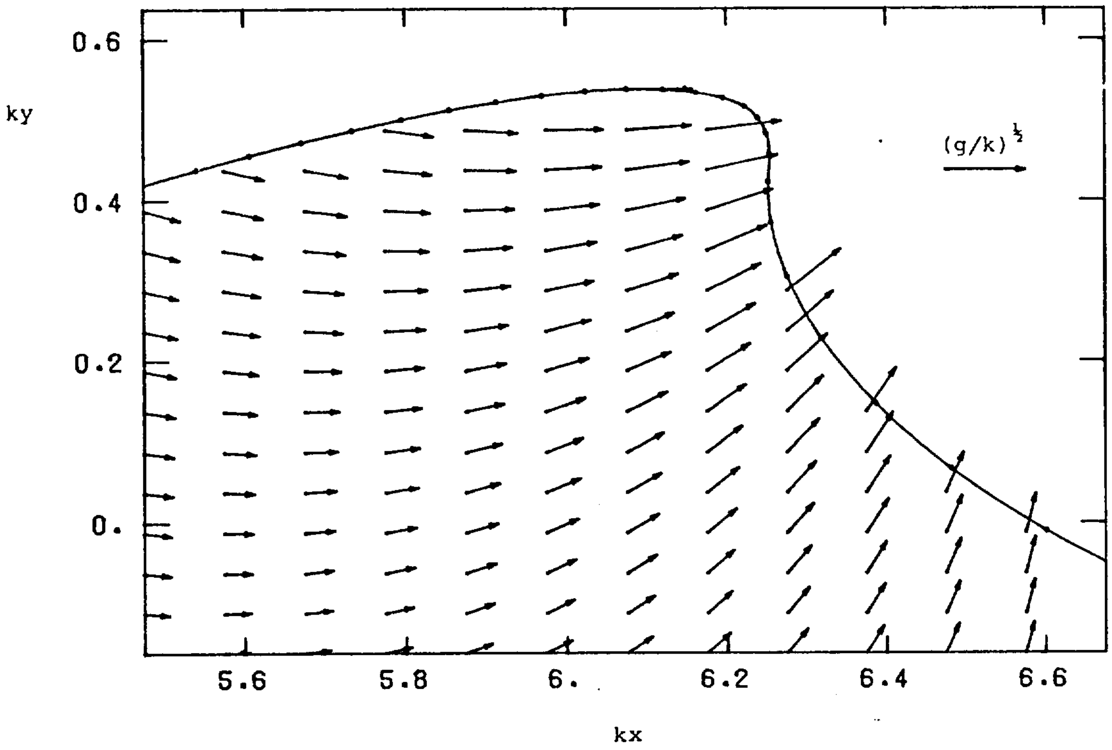

Figure 5a. The velocity field near the wave crest at the moment, $t=t_v$, when the wave face becomes vertical. This wave began as a sinusoid with $E \approx E_{max}$.

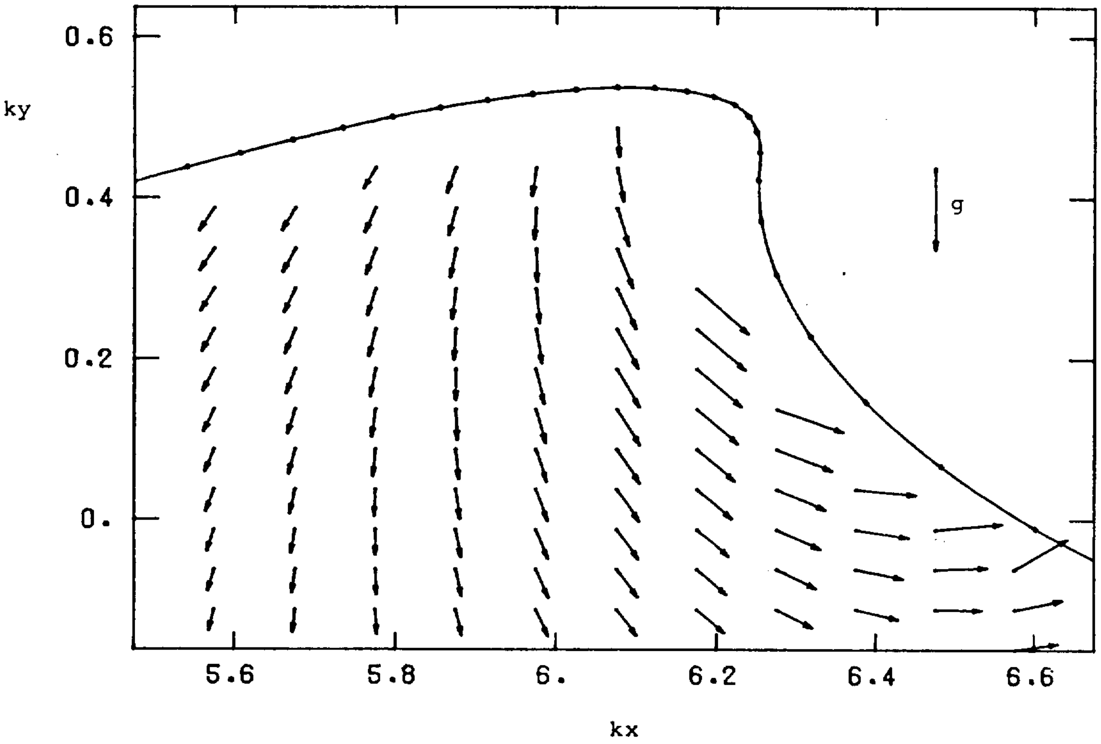

Figure 5b. The acceleration field near the crest at $t=t_v$.

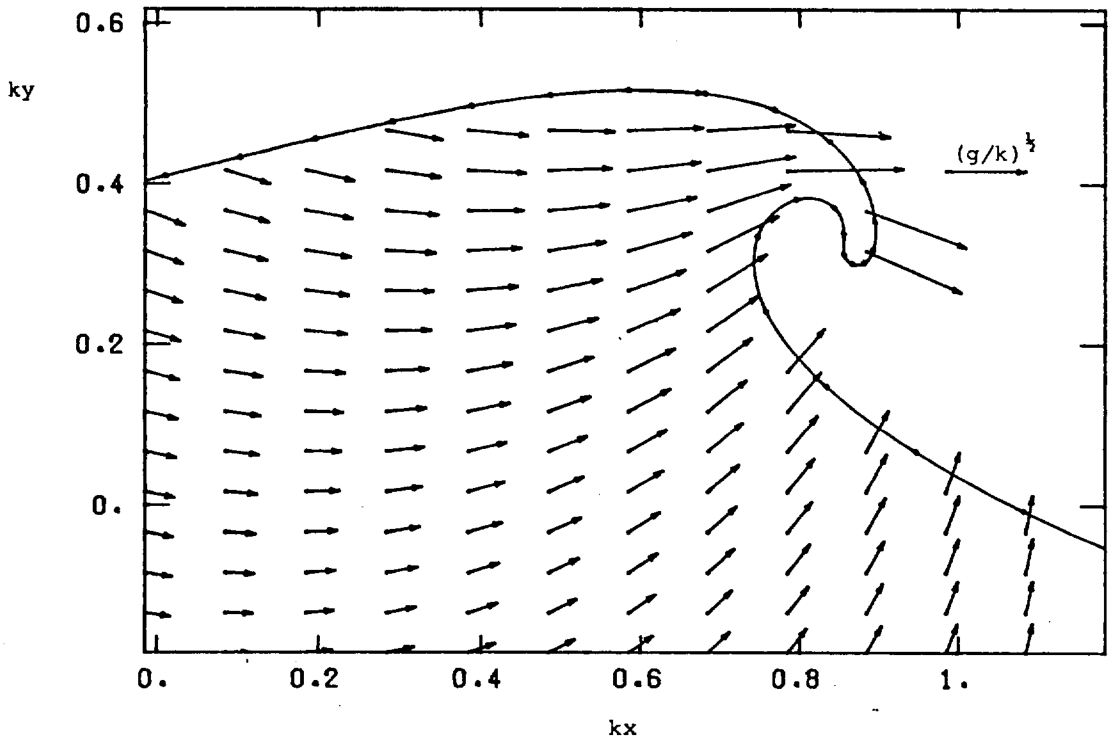

Figure 6a. As for figure 5a but with $t = t_v + 3/4$.

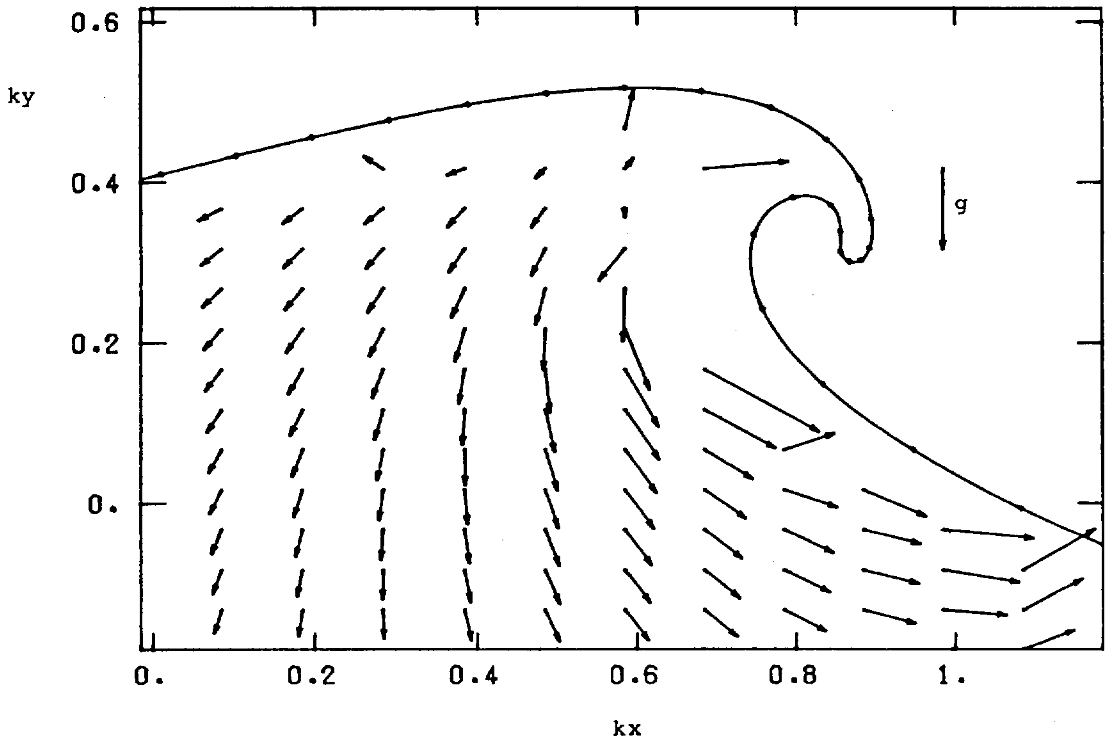

Figure 6b. As for figure 5b but with $t = t_v + 3/4$.

II.C. How INTVEL works

INTVEL works in the following manner. It first reads through the data and locates the time, TIMVER, at which the wave face becomes vertical or overhanging. TIMVER is then the first time, TIMCAL, at which the interior flow field is to be calculated. The program then must calculate

$$\frac{d\psi}{dt} = \frac{D\psi}{Dt}$$

along the surface. This is done by backspacing the data one timestep, calculating

$$\psi(t - \Delta t)$$

using the subroutine CALPSI, spacing forward $2\Delta t$

to find $$\psi(t + \Delta t)$$

and then backspacing to find $\psi(t)$. A centered difference formula gives $\frac{D\psi}{Dt}$.

At t=TIMCAL other simple calculations are performed, and the crest region of size XLEN by YLEN (in non-dimensional units, XLENMM by YLENMM in millimeters) is drawn by CREST. The program then calls WDWDZ which calculates the velocity and pressure within the fluid at discrete points. This is done at intervals of DX and DY, and the velocity vectors, scaled by VELSCL, are drawn by ARROW. Calculated quantities are printed out. The velocities along the boundary can also be drawn, but this is not recommended if an unconfused picture is desired.

The pressure gradients are calculated by a 3-point centered-difference formula. If one of the three points has no pressure value defined because it is outside the fluid, then no pressure gradient or acceleration will be calculated. The accelerations, scaled by ACCSCL, are drawn by ARROW and are printed out. Boundary accelerations can be drawn, but these are very noisy due to the differencing.

This completes one timestep in INTVEL. TIMCAL is incremented by DELT, and the process repeats itself until the data runs out or t exceeds TIMEND.

```
DATE = 20/11/78     TIME =  1.25. 5

FILE = SDMAINDR.INTVEL  (NO)

C
C THIS PROGRAM CALCULATES THE VELOCITY FIELD INTERIOR TO AN IRROTATIONAL
C (TIME-DEPENDENT) DEEP-WATER WAVE GIVEN QUANTITIES ON THE FREE SURFACE.
C
        IMPLICIT REAL*8(A-H, O-Z)
        DIMENSION THETA(61), R(61), PHI(61), PRES(61), DPHIDS(61),
      1  DPHIDN(61), DSDP(61), S(68), DTHDS(61), DRDS(61),
      2  DTHDT(61), DRDT(61), DPHIDT(61), X(61), Y(61), DXDP(61),
      3  DYDP(61), PT(61), W1(61), W2(61), W3(183), W4(61), W5(61),
      4  PSI(61), W6(61), W7(61), DUDTS(61), DVDTS(61)
      5  , D2PDS2(61), D2PDNS(61), D2RDS2(61), D2TDS2(61)
      6  , PHIT(61), PSIT(61), PRES0(15, 19)
        PI = 4.*DATAN(1.D0)
        CALL T4013
        CALL ERRSET(51, -100 000)
        CALL ERRSET(64, -100 000)
        MUNIT = 8
        N = 60
        NP1 = N+1
        XN = DFLOAT(N)
        DO 4 I=1, NP1
        PT(I) = DFLOAT(I)/XN
 4      CONTINUE
C
C LOOK FOR TIME WHEN WAVE FIRST BECOMES VERTICAL.
C
        TIMEND = 10000.D0
        DELT = 0.25D0
 1      READ (MUNIT, END=15) IT, TIME, DELTAT, ENKIN, ENPOT, ENERGY,
      1  YBAR
        READ(MUNIT) THETA, R, PHI, PRES, DPHIDS, DPHIDN, DSDP, S,
      1  DTHDS, DRDS, DPHIDT, DTHDT, DRDT
        DO 2 I=1, N
        IF (DTHDS(I) .GT. 0.) GO TO 2
        TIMVER = TIME
        IT = IT
        GO TO 3
 2      CONTINUE
        GO TO 1
 3      CONTINUE
        TIMCAL = TIMVER
        IF = 4
        GO TO 31
C
C READ THE DATA
C
C FIND THE TIMESTEP NEAREST TO TIMCAL
C
 5      CONTINUE
         READ (MUNIT, END=15) IT, TIME,DELTAT, ENKIN, ENPOT, ENERGY,
      1  YBAR
        IF (DABS(TIME - TIMCAL) .LE. DELTAT/2.) GO TO 30
        IF (TIME - TIMCAL) 20, 30, 10
 10     CONTINUE
         WRITE(6, 1001) IT, TIME
        GO TO 900
 15     CONTINUE
        WRITE(6, 1004) MUNIT, IT, TIME
        GO TO 900
 20     CONTINUE
```

```
      READ (MUNIT)
      GO TO 5
   30 CONTINUE
C
C CALCULATE D(PSI)/DT ALONG THE SURFACE USING A SECOND-ORDER DIFFERENCE
C    FORMULA.
C
C BACKSPACE TO THE TIMESTEP BEFORE TIMCAL.
C
      IE = 3
   31 CONTINUE
      DO 32 I=1, IE
      BACKSPACE MUNIT
   32 CONTINUE
      READ (MUNIT) IT, TIMEM1
      READ(MUNIT) THETA, R, PHI, PRES, DPHIDS, DPHIDN, DSDP, S,
     1  DTHDS, DRDS, DPHIDT, DTHDT, DRDT
      DPHIDS(NP1) = DPHIDS(1)
      DPHIDN(NP1) = DPHIDN(1)
      DRDS(NP1) = DRDS(1)
      DTHDS(NP1) = DTHDS(1)
      DTHDT(NP1) = DTHDT(1)
      DRDT(NP1) = DRDT(1)
      CALL CALPSI(NP1, PHI, DPHIDS, DPHIDN, R, DTHDS, DSDP, PI, W1,
     1  PHINF)
      DO 34 I=1, NP1
      W4(I) = -DTHDT(I)
      W5(I) = DRDT(I)/R(I)
   34 CONTINUE
C
C MOVE TO THE TIMESTEP AFTER TIMCAL.
C
      DO 35 I=1, 2
      READ (MUNIT, END=15)
   35 CONTINUE
      READ (MUNIT) IT, TIMEP1
      READ(MUNIT) THETA, R, PHI, PRES, DPHIDS, DPHIDN, DSDP, S,
     1  DTHDS, DRDS, DPHIDT, DTHDT, DRDT
      DPHIDS(NP1) = DPHIDS(1)
      DPHIDN(NP1) = DPHIDN(1)
      DRDS(NP1) = DRDS(1)
      DTHDS(NP1) = DTHDS(1)
      DTHDT(NP1) = DTHDT(1)
      DRDT(NP1) = DRDT(1)
      CALL CALPSI(NP1, PHI, DPHIDS, DPHIDN, R, DTHDS, DSDP, PI, W2,
     1  PHINF)
      DO 36 I=1, NP1
      W6(I) = -DTHDT(I)
      W7(I) = DRDT(I)/R(I)
   36 CONTINUE
C
C NOW MOVE TO TIME = TIMCAL
C
      DO 40 I=1, 4
      BACKSPACE MUNIT
   40 CONTINUE
      READ (MUNIT) IT, TIME
      READ(MUNIT) THETA, R, PHI, PRES, DPHIDS, DPHIDN, DSDP, S,
     1  DTHDS, DRDS, DPHIDT, DTHDT, DRDT
      DPHIDS(NP1) = DPHIDS(1)
      DPHIDN(NP1) = DPHIDN(1)
      DRDS(NP1) = DRDS(1)
      DTHDS(NP1) = DTHDS(1)
      DPHIDT(NP1) = DPHIDT(1)
      DTHDT(NP1) = DTHDT(1)
      DRDT(NP1) = DRDT(1)
```

```
      WRITE (6, 1002) IT, TIMEM1, TIME, TIMEP1
      CALL CALPSI (NP1, PHI, DPHIDS, DPHIDN, R, DTHDS, DSDP, PT,
     1  PSI, PHINF)
      DELTM1 = TIME - TIMEM1
      DELTP1 = TIMEP1 - TIME
      DELTT2 = TIMEP1 - TIMEM1
      A = DELTM1/DELTP1
      C = DELTP1/DELTM1
      B = C - A
      DO 45 I=1, NP1
      PSIT(I) = (A*W2(I) + B*PSI(I) - C*W1(I))/DELTT2
      DUDTS(I) = (A*W6(I) - B*DTHDT(I) - C*W4(I))/DELTT2
      DVDTS(I) = (A*W7(I) + B*DRDT(I)/R(I) - C*W5(I))/DELTT2
      PHIT(I) = DPHIDT(I) - R(I)*R(I)*(DPHIDS(I)**2 + DPHIDN(I)**2)
   45 CONTINUE
C
C
C DRAW THE CREST
C
      DO 50 I=1, NP1
      X(I) = -THETA(I) + PT(I)*2.*PI
      Y(I) = DLOG(R(I))
   50 CONTINUE
      X(NP1) = X(1)
      CALL TB05AD(NP1, PT, X, DXDP, W3)
      CALL TB05AD(NP1, PT, Y, DYDP, W3)
      DO 60 I=1, NP1
      X(I) = X(I) - PT(I)*2.*PI
      DXDP(I) = DXDP(I) - 2.*PI
   60 CONTINUE
      IBOX = 1
      SPEED = 0.
      XLEN = 1.2D0
      XINC = 0.2D0
      XLENMM = 170.D0
      YLEN = 0.8D0
      YINC = XINC
      YLENMM = YLEN*XLENMM/XLEN
      WVLNTH = 2.*PI
      CALL CREST(N, 0, IBOX, TIME, SPEED, PT, X, Y, DXDP, DYDP,
     1  W1, W2, XMIN, XLEN, XINC, XLENMM, YMIN, YLEN, YINC, YLENMM,
     2  WVLNTH)
      XMAX = XMIN + XLEN
      WRITE(6, 1002)
C
C CALCULATE THE INTERIOR VELOCITIES AND PRESSURES.
C
      C0 = 1.
      VELSCL = 0.1D0/C0
      HEADSZ = VELSCL*C0/10.
      IWRITE = 1
      IEITER = 0
      DX = 0.1D0
      DY = 0.05D0
      X0 = XMIN - DX
      IE = IDINT(XLEN/DX + 1.D-6) + 3
      JE = IDINT(YLEN/DY + 1.D-6) + 3
C
C ......INITIALIZE THE PRESSURE ARRAY.
C
      IEE = IE
      JEE = JE
      DO 61 J=1, JEE
      DO 61 I=1, IEE
      PRESS(I, J) = -1.D-15
   61 CONTINUE
```

```
C
      CALL TB05AD(NP1, PT, DPHIDS, D2PDS2, W3)
      CALL TB05AD(NP1, PT, DPHIDN, D2PDNS, W3)
      CALL TB05AD(NP1, PT, DRDS,   D2RDS2, W3)
      CALL TB05AD(NP1, PT, DTHDS,  D2TDS2, W3)
      DO 62 I=1, NP1
      A = DSDP(I)
      D2PDS2(I) = D2PDS2(I)/A
      D2PDNS(I) = D2PDNS(I)/A
      D2RDS2(I) = D2RDS2(I)/A
      D2TDS2(I) = D2TDS2(I)/A
 62   CONTINUE
      DO 75 I=1, IE
      Y0 = YMIN - DY
      WRITE(6, 1005)
      DO 70 J=1, JE
      THETA0 = -X0
      R0 = DEXP(Y0)
      CALL INSIDC (NP1, THETA0, R0, THETA, R, IENTER, INOUT)
      IF (INOUT .LT. 0) GO TO 65
      CALL WOWDZ( X0, Y0, THETA, R, DTHDS, DRDS, DSDP,
     1  PT, PHI, PSI, DPHIDS, DPHIDN, D2PDS2, D2PDNS, D2RDS2, D2TDS2,
     2  PHIT, PSIT, IWRITE, INOUT, NP1, 1.D-4, 25,
     3  PHI0, PSI0, DPHIDX, DPHIDY, PHIT0, PSIT0, PRES0(I, J),
     4  W1, W2, W3, W4, W5, NITER)
      WRITE(6, 1005)X0, Y0, PHI0, PSI0, DPHIDX, DPHIDY,
     1  PHIT0, PSIT0, PRES0(I, J), NITER, INOUT
 64    CONTINUE
      U = VELSCL*DPHIDX
      V = VELSCL*DPHIDY
      X00 = X0
      CALL ARROW(X00, Y0, U, V, HEADSZ, 1)
 65   CONTINUE
      IENTER = IENTER + 1
      Y0 = Y0 + DY
 70   CONTINUE
      X0 = X0 + DX
 75   CONTINUE
C
C CALCULATE THE VELOCITIES ALONG THE BOUNDARY.
C
      WRITE(6, 1002)
       DO 752 I=1, N, 2
      DPHIDX = -R(I)*(R(I)*DTHDS(I)*DPHIDS(I) - DRDS(I)*DPHIDN(I))
      DPHIDY = R(I)*(DRDS(I)*DPHIDS(I) + R(I)*DTHDS(I)*DPHIDN(I))
      X0 = -THETA(I)
      Y0 = DLOG(R(I))
      CALL MODULO(X0, XMIN, 2.*PI)
      WRITE(6, 1005) X0, Y0, PHI(I), PSI(I), DPHIDX, DPHIDY
      U = VELSCL*DPHIDX
      V = VELSCL*DPHIDY
      CALL ARROW(X0, Y0, U, V, HEADSZ, 1)
 752  CONTINUE
      WRITE(6, 1002)
C
C DRAW THE REFERENCE VELOCITY VECTOR.
C
      X0 = XMIN + XLEN - XINC
      Y0 = YMIN + YLEN - YINC
      U = VELSCL*C0
      CALL ARROW(X0, Y0, U, 0.D0, HEADSZ, 1)
C
C CALCULATE THE PRESSURE GRADIENTS USING SECOND-ORDER DIFFERENCE FORMULAE.
C
      ACCSCL = 0.1D0
      Y0 = XMIN
```

```
      WRITE(6, 1002)
C
      CALL CREST(N, 0, THDX, TIME, SPEED, PT, X, Y, DXDP, DYDP,
     1  W1, W2, XMIN, XLEN, XINC, XLENMM, YMIN, YLEN, YINC, YLENMM,
     2  WVLNTH)
      WRITE(6, 1002)
      IEM1 = IE - 1
      JEM1 = JE - 1
      DO 160 I=2, IEM1
      Y0 = YMIN
      WRITE(6, 1005)
      DO 150 J=2, JEM1
      IF (PRES0(I, J) .EQ. -1.D-15) GO TO 140
      IF (PRES0 (I-1, J) .EQ. -1.D-15) GO TO 140
      IF (PRES0 (I+1, J) .EQ. -1.D-15) GO TO 140
      IF (PRES0 (I, J-1) .EQ. -1.D-15) GO TO 140
      IF (PRES0 (I, J+1) .EQ. -1.D-15) GO TO 140
      DPDX = (PRES0(I+1, J) - PRES0(I-1, J))/(2.*DX)
      DPDY = (PRES0(I, J+1) - PRES0(I, J-1))/(2.*DY)
  130 CONTINUE
      DUDT = - DPDX
      DVDT = - DPDY - 1.
      DPDX = -DPDX
      DPDY = -DPDY
      WRITE(6, 1005) X0, Y0, DPDX, DPDY, DUDT, DVDT
      DUDT = ACCSCL*DUDT
      DVDT = ACCSCL*DVDT
      CALL ARROW(X0, Y0, DUDT, DVDT, HEADSZ, 1)
  140 CONTINUE
      Y0 = Y0 + DY
  150 CONTINUE
      X0 = X0 + DX
  160 CONTINUE
C
C
C DRAW THE ACCELERATIONS ALONG THE BOUNDARY.
C
      WRITE(6, 1002)
       DO 210 I=1, N, 2
      X0 = -THETA(I)
      Y0 = DLOG(R(I))
      CALL MODULO(X0, XMIN, 2.*PI)
      WRITE(6, 1005) X0, Y0, DUDTS(I), DVDTS(I)
      DUDT = ACCSCL*DUDTS(I)
      DVDT = ACCSCL*DVDTS(I)
      CALL ARROW(X0, Y0, DUDT, DVDT, HEADSZ, 1)
  210 CONTINUE
      WRITE(6, 1002)
C
C DRAW THE REFERENCE ACCELERATION VECTOR.
C
      X0 = XMIN + XLEN - XINC
      Y0 = YMIN + YLEN - YINC
      DVDT = -1.*ACCSCL
      CALL ARROW( X0, Y0, 0.D0, DVDT, HEADSZ, 1)
C
C
      TIMCAL = TIMCAL + DELT
      IF (TIMCAL .GT. TIMEND) GO TO 900
      GO TO 5
C
  900  CONTINUE
      CALL GRAEND
      CALL DEVEND
      STOP
 1001 FORMAT(// ' TIMESTEP AND TIME REQUESTED NOT FOUND.  FIRST',
```

```
     1  ' TIMESTEP = ', I6, ' AT TIME = ', F10.6//)
 1002 FORMAT(///I6, 3F10.4//)
 1003 FORMAT(1X, I6, 1P10D11.3)
 1004  FORMAT(//'  END OF DATA ON UNIT ', I4,/
     1  ' LAST TIMESTEP =', I5, '  AT TIME =', F10.6//)
 1005  FORMAT(1X, 1P9D12.2, 2I4)
       END
```

```
DATE = 20/11/78      TIME =  1.25. 6

FILE = SOMAINDP.INTVEL  .ED (YS)

2 $G LN=   0.  1,DG=ONLY( 2)
3 $G LN=   0.  2,CC=ONLY( 2, 3):CALCOMP AND DEFAULT GROUP
5 $G LN=   0.  3,P1=ONLY( 2, 5):120 POINTS ALONG PROFILE
6 $G LN=   0.  4,ST=ONLY( 2, 5, 6):STABILITY EXPER. RESULTS
5 $F LN=   6.  1,C1= 23,C2= 24:121
5 $F LN=   6.  2,C1= 30,C2= 31:121
5 $F LN=   6.  3,C1= 39,C2= 40:121
5 $F LN=   6.  4,C1= 49,C2= 50:121
5 $F LN=   6.  5,C1= 61,C2= 62:121
5 $F LN=   7.  1,C1= 16,C2= 17:121
5 $F LN=   7.  2,C1= 26,C2= 27:121
5 $F LN=   7.  3,C1= 33,C2= 34:128
5 $F LN=   7.  4,C1= 44,C2= 45:121
5 $F LN=   7.  5,C1= 54,C2= 55:121
5 $F LN=   8.  1,C1= 15,C2= 16:121
5 $F LN=   8.  2,C1= 25,C2= 26:121
5 $F LN=   8.  3,C1= 37,C2= 38:121
5 $F LN=   8.  4,C1= 44,C2= 45:121
5 $F LN=   8.  5,C1= 51,C2= 52:121
5 $F LN=   8.  6,C1= 61,C2= 62:121
5 $F LN=   9.  1,C1= 14,C2= 15:121
5 $F LN=   9.  2,C1= 22,C2= 23:121
5 $F LN=   9.  3,C1= 30,C2= 31:121
5 $F LN=   9.  4,C1= 38,C2= 39:121
5 $F LN=   9.  5,C1= 46,C2= 48:363
5 $F LN=   9.  6,C1= 55,C2= 56:121
5 $F LN=   9.  7,C1= 63,C2= 64:121
5 $F LN=  10.  1,C1= 13,C2= 14:121
5 $F LN=  10.  2,C1= 21,C2= 22:121
5 $F LN=  10.  3,C1= 29,C2= 30:121
5 $F LN=  10.  4,C1= 40,C2= 41:121
5 $F LN=  10.  5,C1= 51,C2= 52:121
5 $F LN=  11.  1,C1= 18,C2= 19:121
5 $F LN=  11.  2,C1= 30,C2= 31:121
5 $F LN=  11.  3,C1= 42,C2= 43:121
5 $F LN=  11.  4,C1= 54,C2= 55:121
5 $F LN=  12.  1,C1= 16,C2= 17:121
5 $F LN=  12.  2,C1= 26,C2= 27:121
3 $F LN=  14.  1,C1= 12,C2= 16:CC925
5 $F LN=  18.  1,C1= 11,C2= 12:120
6 $F LN=  28.  1,C1= 20,C2= 19:/DSQRT(2.D0)
6 $F LN= 159.  1,C1= 14,C2= 16:0.6
6 $F LN= 160.  1,C1= 14,C2= 16:0.1
6 $F LN= 162.  1,C1= 14,C2= 16:0.4
6 $F LN= 174.  1,C1= 14,C2= 13:/DSQRT(2.D0)
6 $F LN= 175.  1,C1= 16,C2= 18:0.05
6 $F LN= 179.  1,C1= 12,C2= 14:0.05
6 $F LN= 180.  1,C1= 12,C2= 14:0.025
2 $D LN= 232.  1,L2= 247
6 $F LN= 257.  1,C1= 16,C2= 18:0.05
2 $D LN= 294.  1,L2= 307
```

This is a RHEL ELECTRIC edit file which modifies the file INTVEL. See the ELECTRIC manual for command descriptions

II.D. Example of printed output

The next two pages contain examples of the printed output from INTVEL. On the first page are given the timestep number, the value of t before, during and after the current time, and the coordinates of the box drawn around the crest portion of the new profile. Next are listed the (x,y) coordinates of each point and the values of ϕ, ψ, $\frac{\partial\phi}{\partial x}$, $\frac{\partial\phi}{\partial y}$, $\frac{\partial\phi}{\partial t}$, $\frac{\partial\psi}{\partial t}$ and p (pressure) calculated there. Also listed are the number of iterations, NITER, it took to produce these values and the value of INOUT which is returned by subroutine INSIDC. This should normally be 1 or 0 indicating that the point in question is inside or on the boundary of the fluid. The error return from WDWDZ means that this subroutine failed to converge on at least one of the variables it was trying to calculate. This happens when a grid point at which the velocity, say, is being calculated is very near a computational fluid particle on the free surface. When divergence occurs WDWDZ returns the initial guesses for all of the variables which it is trying to calculate.

The second page lists the output from that part of the program which calculates the pressure gradients and accelerations.

```
TIME-
STEP   t-Δt       t          t+Δt
062    100.6406   100.6445   100.6484



-NEW GRAPH
X AXIS DRAWN FROM    1.225930 TO    1.825930 IN INCREMENTS OF    0.100000
Y AXIS DRAWN FROM   -0.083505 TO    0.316495 IN INCREMENTS OF    0.100000


     x          y          φ          ψ          φx         φy         φt         ψt          p      NITER  INOUT
  1.18D+00  -1.09D-01  -1.42D-01   7.43D-02   1.48D-01  -8.43D-02  -1.22D-01  -6.76D-02   2.16D-01    2    1
  1.18D+00  -8.35D-02  -1.44D-01   7.81D-02   1.56D-01  -9.04D-02  -1.28D-01  -7.21D-02   1.95D-01    2    1
  1.18D+00  -5.85D-02  -1.46D-01   8.21D-02   1.63D-01  -9.71D-02  -1.34D-01  -7.68D-02   1.74D-01    2    1
  1.18D+00  -3.35D-02  -1.49D-01   8.63D-02   1.72D-01  -1.04D-01  -1.40D-01  -8.21D-02   1.54D-01    2    1
  1.18D+00  -8.51D-03  -1.51D-01   9.07D-02   1.80D-01  -1.12D-01  -1.47D-01  -8.77D-02   1.33D-01    3    1
  1.18D+00   1.65D-02  -1.54D-01   9.53D-02   1.89D-01  -1.21D-01  -1.54D-01  -9.39D-02   1.12D-01    3    1
  1.18D+00   4.15D-02  -1.57D-01   1.00D-01   1.97D-01  -1.31D-01  -1.61D-01  -1.01D-01   9.12D-02    3    1
  1.18D+00   6.65D-02  -1.61D-01   1.05D-01   2.06D-01  -1.41D-01  -1.68D-01  -1.08D-01   7.00D-02    3    1
  1.18D+00   9.15D-02  -1.64D-01   1.10D-01   2.15D-01  -1.52D-01  -1.75D-01  -1.16D-01   4.86D-02    4    1
  1.18D+00   1.16D-01  -1.68D-01   1.16D-01   2.24D-01  -1.65D-01  -1.82D-01  -1.25D-01   2.65D-02    6    1

ERROR RETURN FROM WDWDZ:  NO CONVERGENCE AFTER 50 ITERATIONS.

  1.18D+00   1.41D-01  -1.73D-01   1.22D-01   2.33D-01  -1.78D-01  -1.84D-01  -1.38D-01  -5.84D-04   50    1

  1.23D+00  -1.09D-01  -1.34D-01   7.81D-02   1.59D-01  -6.84D-02  -1.30D-01  -5.54D-02   2.23D-01    2    1
  1.23D+00  -8.35D-02  -1.36D-01   8.22D-02   1.68D-01  -7.38D-02  -1.36D-01  -5.92D-02   2.03D-01    2    1
  1.23D+00  -5.85D-02  -1.38D-01   8.65D-02   1.77D-01  -7.98D-02  -1.43D-01  -6.35D-02   1.83D-01    2    1
  1.23D+00  -3.35D-02  -1.40D-01   9.11D-02   1.86D-01  -8.64D-02  -1.51D-01  -6.81D-02   1.63D-01    2    1
  1.23D+00  -8.51D-03  -1.42D-01   9.58D-02   1.96D-01  -9.38D-02  -1.58D-01  -7.33D-02   1.43D-01    2    1
  1.23D+00   1.65D-02  -1.44D-01   1.01D-01   2.06D-01  -1.02D-01  -1.66D-01  -7.90D-02   1.23D-01    3    1
  1.23D+00   4.15D-02  -1.47D-01   1.06D-01   2.17D-01  -1.11D-01  -1.74D-01  -8.52D-02   1.03D-01    3    1
  1.23D+00   6.65D-02  -1.50D-01   1.12D-01   2.28D-01  -1.21D-01  -1.83D-01  -9.22D-02   8.30D-02    3    1
  1.23D+00   9.15D-02  -1.53D-01   1.18D-01   2.39D-01  -1.32D-01  -1.91D-01  -9.98D-02   6.27D-02    3    1
  1.23D+00   1.16D-01  -1.57D-01   1.24D-01   2.50D-01  -1.44D-01  -2.00D-01  -1.08D-01   4.22D-02    3    1
  1.23D+00   1.41D-01  -1.60D-01   1.30D-01   2.62D-01  -1.58D-01  -2.10D-01  -1.17D-01   2.16D-02    4    1

  1.28D+00  -1.09D-01  -1.26D-01   8.11D-02   1.68D-01  -5.05D-02  -1.36D-01  -4.17D-02   2.29D-01    2    1
  1.28D+00  -8.35D-02  -1.27D-01   8.55D-02   1.78D-01  -5.50D-02  -1.44D-01  -4.49D-02   2.10D-01    2    1
  1.28D+00  -5.85D-02  -1.29D-01   9.00D-02   1.88D-01  -6.00D-02  -1.52D-01  -4.84D-02   1.91D-01    2    1
  1.28D+00  -3.35D-02  -1.30D-01   9.49D-02   1.99D-01  -6.56D-02  -1.60D-01  -5.23D-02   1.71D-01    2    1
  1.28D+00  -8.51D-03  -1.32D-01   1.00D-01   2.11D-01  -7.20D-02  -1.68D-01  -5.67D-02   1.52D-01    2    1
  1.28D+00   1.65D-02  -1.34D-01   1.05D-01   2.23D-01  -7.92D-02  -1.77D-01  -6.17D-02   1.33D-01    2    1
  1.28D+00   4.15D-02  -1.36D-01   1.11D-01   2.35D-01  -8.74D-02  -1.87D-01  -6.73D-02   1.14D-01    2    1
```

```
-NEW GRAPH
X AXIS DRAWN FROM    1.225930 TO    1.825930 IN INCREMENTS OF    0.100000
Y AXIS DRAWN FROM   -0.083505 TO    0.316495 IN INCREMENTS OF    0.100000

     X            y           -Px          -Py           Du/Dt        Dv/Dt

 1.23D+00    -8.35D-02    -1.48D-01    8.04D-01    -1.48D-01    -1.96D-01
 1.23D+00    -5.85D-02    -1.62D-01    8.01D-01    -1.62D-01    -1.99D-01
 1.23D+00    -3.35D-02    -1.76D-01    7.99D-01    -1.76D-01    -2.01D-01
 1.23D+00    -8.51D-03    -1.92D-01    7.99D-01    -1.92D-01    -2.01D-01
 1.23D+00     1.65D-02    -2.09D-01    8.00D-01    -2.09D-01    -2.00D-01
 1.23D+00     4.15D-02    -2.27D-01    8.04D-01    -2.27D-01    -1.96D-01
 1.23D+00     6.65D-02    -2.47D-01    8.09D-01    -2.47D-01    -1.91D-01
 1.23D+00     9.15D-02    -2.69D-01    8.16D-01    -2.69D-01    -1.84D-01
 1.23D+00     1.16D-01    -2.94D-01    8.22D-01    -2.94D-01    -1.78D-01
 1.23D+00     1.41D-01    -3.65D-01    9.43D-01    -3.65D-01    -5.71D-02

 1.28D+00    -8.35D-02    -1.19D-01    7.79D-01    -1.19D-01    -2.21D-01
 1.28D+00    -5.85D-02    -1.32D-01    7.73D-01    -1.32D-01    -2.27D-01
 1.28D+00    -3.35D-02    -1.46D-01    7.69D-01    -1.46D-01    -2.31D-01
 1.28D+00    -8.51D-03    -1.62D-01    7.65D-01    -1.62D-01    -2.35D-01
 1.28D+00     1.65D-02    -1.79D-01    7.64D-01    -1.79D-01    -2.36D-01
 1.28D+00     4.15D-02    -1.98D-01    7.65D-01    -1.98D-01    -2.35D-01
 1.28D+00     6.65D-02    -2.19D-01    7.69D-01    -2.19D-01    -2.31D-01
```

II.E. Subroutines called by INTVEL

INTVEL calls subroutines from three different sources: (1) routines written by E. D. Cokelet and described in this manual, (2) IOS grafix, version 0 routines, and (3) Harwell routines. The EDC routines are in ELECTRIC files at RHEL. The last two are in load modules at RHEL. INTVEL calls the following:

EDC Routines: CALPSI, CREST, INSIDC, WDWDZ, ARROW, MODULO.

IOS Grafix: T4013 or CC925, ERRSEL, GRAEND, DEVEND.

Harwell routines: TB05AD.

II.F. Description of EDC subroutines

We give here a brief description of the subroutines associated with INTVEL. They fall into two categories, numerical and graphical. All real variables are DOUBLE PRECISION unless stated otherwise.

II.F.1. Numerical Routines

CALPSI (NP1, PHI, DPHIDS, DPHIDN, R, DTHDS, DSDP, PT, PSI, PHINF)

Calculates the stream function,   , along the free surface of a periodic wave in deep water.

Inputs:

NP1 = N+1,	number of points (must be odd) along curve
PHI (NP1)	ϕ along curve
DPHIDS (NP1)	$\frac{\partial \phi}{\partial s}$
DPHIDN (NP1)	$\frac{\partial \phi}{\partial n}$
R (NP1)	r
DTHDS (NP1)	$\frac{d\theta}{ds}$
DSDP (NP1)	$\frac{ds}{dp}$

Note all the above arrays must be periodic. If F is any array, then F(1) = F(NP1).

PT (NP1)	point number, P

Outputs:

PSI (NP1)	ψ along wave
PHINF	$\left\{\begin{array}{l}\phi \text{ at } y = -\infty. \\ \psi \text{ at } y = -\infty\end{array}\right.$ is chosen to be zero.

Subroutines called: None

```
DATE = 21/11/78     TIME =  1.25. 6

FILE = SOMAINDR.CALPSI  (DG)

      SUBROUTINE CALPSI (NP1, PHI, DPHIDS, DPHIDN, R, DTHDS, DSDP, PT,
     1  PSI, PHINF)
C
C THIS SUBROUTINE CALCULATES PSI ALONG THE FREE SURFACE FOR A PERIODIC,
C INFINITELY DEEP FLUID.
C
C INPUTS:
C  NP1   = NUMBER OF POINTS ALONG CURVE WHICH MUST BE ODD.
C  PHI, D(PHI)/DS, D(PHI)/DN, R, D(THETA)/DS, DS/D(PT), PT  ARRAYS OF VALUES
C    DESCRIBING VELOCITY, BOUNDING CURVE AND POINT NUMBER.
C
C OUTPUTS:
C  PSI     ALONG THE SURFACE
C  PHINF   = PHI AT Y = - INFINITY (R = 0).
C
C  FDC   31 JULY 1978.
C
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION DPHIDN(NP1), R(NP1), DTHDS(NP1), DSDP(NP1), PT(NP1),
     1  PSI(NP1), DPHIDS(NP1), PHI(NP1)
      PI = 4.*DATAN(1.D0)
      N = NP1 - 1
      NM1 = N-1
      IF (2*INT(FLOAT(N)/2. + .001) .EQ. N) GO TO 12
      WRITE(6, 1001) NP1
      STOP
1001 FORMAT(///' ERROR RETURN FROM CALPSI:  NP1 = ', I6,
     1  ' WHICH IS NOT ODD '//)
 12   CONTINUE
C
C CALCULATE PSI + CONSTANT ALONG THE SURFACE.
C
      DELP = PT(2) - PT(1)
      PSI(N) = 0.
      PSI(2) = PSI(N) + (DPHIDN(N)*DSDP(N) + 4.*DPHIDN(1)*DSDP(1) +
     1  DPHIDN(2)*DSDP(2))*DELP/3.
      DO 30 I=4, N, 2
      PSI(I) = PSI(I-2) + (DPHIDN(I-2)*DSDP(I-2) + 4.*DPHIDN(I-1)*
     1  DSDP(I-1) + DPHIDN(I)*DSDP(I))*DELP/3.
 30   CONTINUE
      PSI(N) = 0.
      PSI(1)=PSI(N) + (-DPHIDN(N-1)*DSDP(N-1) + 13.*DPHIDN(N)*DSDP(N)
     1  + 13.*DPHIDN(1)*DSDP(1) - DPHIDN(2)*DSDP(2))*DELP/24.
      DO 40 I=3, N, 2
      PSI(I) = PSI(I-2) + (DPHIDN(I-2)*DSDP(I-2) + 4.*DPHIDN(I-1)
     1  *DSDP(I-1) + DPHIDN(I)*DSDP(I))*DELP/3.
 40   CONTINUE
      PSI(NP1) = PSI(1)
C
C CALCULATE PHI AND PSI AT -INFINITY AND EVALUATE THE CONSTANT.
C THIS USES GREEN'S THIRD IDENTITY.
C
      PHINF = 0.
      PSINF = 0.
      DO 50 I=1, NM1, 2
      DLR = DLOG(R(I))
      PHINF = PHINF + 2.*(PHI(I)*DTHDS(I) - DPHIDN(I)*
     1  DLR)*DSDP(I)
      PSINF = PSINF + 2.*(PSI(I)*DTHDS(I) + DPHIDS(I)*
     1  DLR)*DSDP(I)
```

```
 50   CONTINUE
      DO 60 I=2, N, 2
      DLR = DLOG(R(I))
      PHINF = PHINF + 4.*(PHI(I)*DTHDS(I) - DPHIDN(I)*
     1  DLR)*DSDP(I)
      PSINF = PSINF + 4.*(PSI(I)*DTHDS(I) + DPHIDS(I)*
     1  DLR)*DSDP(I)
 60   CONTINUE
      PHINF = DELP*PHINF/(6.*PI)
      PSINF = DELP*PSINF/(6.*PI)
      DO 70 I=1, NP1
      PSI(I) = PSI(I) - PSINF
 70   CONTINUE
      RETURN
      END
```

INSIDC (NP1, THETAO, RO, THETA, R, IENTER, INOUT)

Determines whether a point whose polar coordinates are (THETAO, RO) is inside (INOUT = 1), on (INOUT = 0), or outside (INOUT=-1) of a closed contour approximated by straight line segments connecting a sequence of NP1 points (THETA, R). If IENTER is greater than 0, the subroutine assumes that it has been entered previously for this contour. For points near the bounding contour the question is answered by calculating $\int_C \frac{1}{\zeta - \zeta_0} \, d\zeta$.

Inputs:

NP1	number of points around curve
THETAO	θ_0
RO	r_0
THETA (NP1)	θ values around curve. THETA (NP1) = THETA(1) + 2 π
R (NP1)	r values around curve. R (NP1) = R (1)
IENTER	parameter to indicate if routine entered earlier for this contour

Outputs:

INOUT	indicates in (+1), on (0) or out (-1)

Subroutines called: None

```
DATE = 24/11/78      TIME =  1.25. 7

FILE = SOMAINDR.INSIDC  (DG)

        SUBROUTINE INSIDC (NP1, THETA0, R0, THETA, R, IENTER, INOUT)
C
C THIS ROUTINE DETERMINES WHETHER OR NOT A POINT (R0, THETA0) IS
C INSIDE (INOUT = 1), ON (INOUT = 0), OR OUTSIDE (INOUT = -1) OF
C A BOUNDING CONTOUR WHOSE COORDINATES ARE GIVEN BY (R(NP1), THETA(NP1)).
C IF IENTER .GT. 0, THE SUBROUTINE ASSUMES THAT IT HAS BEEN ENTERED
C PREVIOUSLY FOR THIS CONTOUR.
C
C FOR POINTS NEAR THE BOUNDING CONTOUR THE QUESTION IS ANSWERED BY
C CALCULATING THE INTEGRAL 1./(Z - Z0) DZ AROUND THE CONTOUR.
C EDC JULY 1978.
C
        IMPLICIT REAL*8(A-H, O-Z)
        DIMENSION R(NP1), THETA(NP1)
        IF (IENTER) 10, 10, 35
C
C SUBROUTINE NOT ENTERED BEFORE FOR THIS BOUNDING CURVE.
C
  10    CONTINUE
        EPS1 = 1.D-5
        EPS2 = 1.D-5
        PI = 4.*DATAN(1.D0)
        EPS3 = 1.D-10
        RMIN = 1.D+50
        RMAX = 0.
        DO 30 I=1, NP1
        IF (R(I) .GE. RMIN) GO TO 20
        RMIN = R(I)
        IMIN = I
  20    IF (R(I) .LE. RMAX) GO TO 30
        RMAX = R(I)
        IMAX = I
  30    CONTINUE
C
C IS R0 .GT. RMAX OR .LE. RMIN?
C
  35    CONTINUE
        IF (R0 .GE. (RMIN - EPS1)) GO TO 40
        INOUT = 1
        RETURN
  40    IF (R0 .LE. (RMAX + EPS1)) GO TO 50
        INOUT = -1
        RETURN
C
C R0 BETWEEN RMIN AND RMAX.
C
  50    CONTINUE
        Y0 = R0*DSIN(THETA0)
        X0 = R0*DCOS(THETA0)
        Y = R(1)*DSIN(THETA(1)) - Y0
        X = R(1)*DCOS(THETA(1)) - X0
        IF ((X*X + Y*Y)/(R0*R0) .GT. EPS3) GO TO 53
        INOUT = 0
        RETURN
  53    CONTINUE
        ALIP1 = DATAN2(Y, X)
        ALI = ALIP1
        DO 100 I=2, NP1
        ALI = ALIP1
        Y = R(I)*DSIN(THETA(I)) - Y0
```

```
      X = R(I)*DCOS(THETA(I)) - X0
      IF ((X*X + Y*Y)/(R0*R0) .GT. EPS3) GO TO 56
      INOUT = 0
      RETURN
 56   CONTINUE
      ALIP1 = DATAN2(Y, X)
      DIFF = ALIP1 - ALT
      IF (DIFF .LE. (PI + EPS2)) GO TO 60
      ALIP1 = ALIP1 - 2.*PI
      DIFF = DIFF - 2.*PI
 60   IF (DIFF .GE. -(PI + EPS2)) GO TO 70
      ALIP1 = ALIP1 + 2.*PI
      DIFF = DIFF + 2.*PI
 70   CONTINUE
      DIFF = DABS(DIFF)
      IF (DIFF .LT. (PI - EPS2)) GO TO 100
      INOUT = 0
      RETURN
 100  CONTINUE
      IF ((ALIP1 - ALT) .GE. PI) GO TO 110
      INOUT = -1
      RETURN
 110  CONTINUE
      INOUT = 1
      RETURN
      END
```

INTERP (ISEEK, N, X, F, DFDX, XN, FN, DFDXN)

Calculates the value of a cubic spline, FN, and its first derivative DFDXN at one point, XN, in the range X (1) to X (N). The X(I)'s are assumed to be in ascending order. The subroutine needs the N knots, X (I), the function values at the knots, F (I), and the first derivatives, DFDX(I), there. If ISEEK is greater than or equal to 0 the routine assumes it has been entered previously with a smaller XN. This reduces the search time.

Inputs:

ISEEK	indicates if entered previously for this routine
N	number of points
X (N)	X
F (N)	f
DFDX (N)	$\frac{df}{dx}$
XN	x_0

Outputs:

FN	f_0
DFDXN	$\frac{df}{dx_0}$

Subroutines called: None

```
      SUBROUTINE INTERP(ISEEK, N, X, F, DFDX, XN, FN, DFDXN)
C   THIS SUBROUTINE CALCULATES THE VALUE OF A CUBIC SPLINE, FN, AND ITS FIRST
C  DERIVATIVE, DFDXN, AT ANY POINT, XN, IN THE RANGE X(1) TO X(N).  THE X(I)'S
C  ARE ASSUMED TO BE IN ASCENDING ORDER.  THE SUBROUTINE NEEDS THE N KNOTS,
C  X(I), THE FUNCTION VALUES AT THE KNOTS, F(I), AND THE DERIVATIVES AT THE
C  KNOTS, DFDX(I).  IF ISEEK IS .GE. 0 THE SUBROUTINE ASSUMES IT HAS BEEN
C  ENTERED PREVIOUSLY WITH A SMALLER XN.  THIS REDUCES THE SEARCH TIME.
      IMPLICIT REAL*8(A-H, O-Z)
 1000 FORMAT(/' THE SPLINE IS TO BE INTERPOLATED AT THE POINT ',
     1  1PD23.15,/' WHICH IS SMALLER THAN THE MINIMUM VALUE OF THE RANGE
     2', D23.15/)
 1001 FORMAT(/' THE SPLINE IS TO BE INTERPOLATED AT THE POINT ',
     1  1PD23.15,/ ' WHICH IS LARGER THAN THE MAXIMUM VALUE OF THE RANGE
     2', D23.15/)
      DIMENSION X(N), F(N), DFDX(N)
      IF (ISEEK .LT. 0) ISAVE=2
      DO 10 I=ISAVE, N
      IF ( XN .GT. (X(I) + 1.D-13*DABS(X(I)))) GO TO 10
      IF (XN .GE. (X(I-1) - 1.D-13*DABS(X(I-1)))) GO TO 15
      J=ISAVE - 1
      WRITE(6, 1000) XN, X(J)
      STOP
 10   CONTINUE
      WRITE(6, 1001) XN, X(N)
      STOP
 15   CONTINUE
      ISAVE = I
      C3 = (DFDX(I-1) + DFDX(I) + 2.*(F(I) - F(I-1))/(X(I-1) - X(I)))/
     1  (X(I-1) - X(I))**2
      C2 = (DFDX(I-1) - DFDX(I))/(2.*(X(I-1) - X(I))) - 3.*C3*(X(I-1) +
     1  X(I))/2.
      C1 = DFDX(I-1) - 2.*C2*X(I-1) - 3.*C3*X(I-1)**2
      C0 = F(I-1) - C1*X(I-1) - C2*X(I-1)**2 - C3*X(I-1)**3
      FN = C0 + C1*XN + C2*XN**2 + C3*XN**3
      DFDXN = C1 + 2.*C2*XN + 3.*C3*XN**2
      RETURN
      END
```

MODULO (X, XMIN, XPER)

Adds or subtracts integral factors of XPER to X until $XMIN \leqslant X \leqslant XMIN + XPER$

Inputs:

X	X
XMIN	minimum value of X
XPER	period in X

Outputs:

X	$XMIN \leqslant X \leqslant XMIN + XPER$

Subroutines called: None

```
DATE = 20/11/78     TIME =  1.25, 9

FILE = SDMAILDR.MODULO  (DG)

      SUBROUTINE MODULO(X, XMIN, XPER)
C
C  MAKES X BETWEEN XMIN AND XMIN + XPER WITH PERIOD XPER.
C
      IMPLICIT REAL*8(A-H, O-Z)
      XMAX = XMIN + XPER
 10   IF (X .LE. XMAX) GO TO 20
      X = X - XPER
      GO TO 10
 20   IF ( X .GE. XMIN) GO TO 30
      X = X + XPER
      GO TO 20
 30    RETURN
      END
```

WDWDZ (XO, YO, THETA, R, DTHDS, DRDS, DSDP, P, PHI, PSI, DPHIDS, DPHIDN, D2PDS2, D2PDNS, D2RDS2, D2TDS2, PHIT, PSIT, IWRITE, INOUT, NP1, EPSLON, NMAX, PHIO, PSIO, DPHIDX, DPHIDY, PHITO, PSITO, PRESO, SIMPCO, CO, SI, C, S, NITER)

calculates ϕ, ψ, $\frac{\partial\phi}{\partial x}$, $\frac{\partial\phi}{\partial y}$, $\frac{\partial\phi}{\partial t}$, $\frac{\partial\psi}{\partial t}$ and p

at (x_0, y_0) for an infinitely - deep fluid. If the point (x_0, y_0) is near the boundary then the solution is found iteratively.

Note: The section which calculates the output variables for points outside of the bounding contour may not be conceptually correct. Also it has not been debugged.

Inputs:

XO, YO	coordinates of point in physical Z-plane
THETA (NP1)	array of θ values along curve in ζ-plane
R (NP1)	r values along curve
DTHDS (NP1)	$\frac{d\theta}{ds}$
DRDS (NP1)	$\frac{dr}{ds}$
DSDP (NP1)	$\frac{ds}{dp}$
P (NP1)	point number
PHI (NP1)	ϕ
PSI (NP1)	ψ

DPHIDS (NP1)	$\frac{\partial \phi}{\partial s}$
DPHIDN (NP1)	$\frac{\partial \phi}{\partial n}$
D2PDS2 (NP1)	$\frac{\partial^2 \phi}{\partial s^2}$
D2PDNS (NP1)	$\frac{\partial^2 \phi}{\partial n \partial s}$
D2RDS2 (NP1)	$\frac{d^2 r}{d s^2}$
D2TDS2 (NP1)	$\frac{d^2 \theta}{d s^2}$
PHIT (NP1)	$\frac{\partial \phi}{\partial t}$
PSIT (NP1)	$\frac{\partial \psi}{\partial t}$

IWRITE	controls writing out of intermediate iterates. If ≤ 0 interative values of ϕ, ψ, $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ are written. If > 0 they are not.
INOUT	tells if (x_0, y_0) is inside, on, or outside of bounding curve =1 inside =0 on =-1 outside
NP1 = N+1	number of points on curve and array size
EPSLON	a small positive number specifying convergence criterion. If successive values of each output variable differ by less than EPSLON, then convergence is assumed. EPSLON should be about 1×10^{-5} for 60 points along the profile
NMAX	maximum number of iterations to convergence. NMAX should be about 50 for N=60, EPSLON = 1×10^{-5}

Outputs:

PHIO	ϕ_0
PSIO	ψ_0
DPHIDX	$\frac{\partial \phi}{\partial x}_0$
DPHIDY	$\frac{\partial \phi}{\partial y}_0$

PHITO	$\frac{\partial \phi}{\partial t}_0$
PSITO	$\frac{\partial \psi}{\partial t}_0$
PRESO	p_0
SIMPCO (NP1)	working array
CO (NP1)	working array
SI (NP1)	working array
C (NP1)	working array
S (NP1)	working array
NITER	number of interations until convergence or divergence. If NITER$>$NMAX then the routine gives an error message, and the output variables are the initial guesses.

Subroutines called: None

```
DATE = 29/11/78      TIME =  1.25. 7

FILE = SOMAILDR.WDWDZ    (DG)

  1.       SUBROUTINE WDWDZ(X0, Y0, THETA, R, DTHDS, DRDS, DSDP,
  2.      1  P, PHI, PSI, DPHIDS, DPHIDN, D2PDS2, D2PDNS,
  3.      2  D2RDS2, D2TDS2, PHIT, PSIT, IWRITE, INOUT, NP1, EPSLON, NMAX,
  4.      3  PHI0, PSI0, DPHIDX, DPHIDY, PHIT0, PSIT0, PRES0,
  5.      4    SIMPCO, CO, SI, C, S, NITER)
  6.C
  7.C   THIS SUBROUTINE CALCULATES PHI, PSI, D(PHI)/DX ,D(PHI)/DY,
  8.C PARTIAL D(PHI)/DT, PARTIAL D(PSI)/DT AND PRESSURE
  9.C AT X0, Y0 FOR AN INFINITELY DEEP FLUID.
 10.C
 11.C
 12.C IF THE POINT X0, Y0 IS NEAR THE BOUNDARY THEN THE SOLUTION IS FOUND
 13.C ITERATIVLLY.  NITER IS THE NUMBER OF ITERATIONS.  IF NITER .GT. NMAX
 14.C THE PROGRAM RETURNS WITH AN ERROR MESSAGE
 15.C AND THE VALUES OF PHI0, PSI0, DPHIDX ,DPHIDY, PHIT0, PSIT0 AND PRES0
 16.C ARE THE INITIAL GUESSES.
 17.C
 18.C NOTE:  NP1 MUST BE ODD BECAUSE SIMPSON'S RULE USED FOR QUADRATURE.
 19.C
 20.C INPUTS:
 21.C  X0, Y0    COORDINATES OF POINT IN PHYSICAL (X,Y) SPACE.
 22.C  THETA, R, D(THETA)/DS, D(R)/DS, D(S)/DP, P, PHI, PSI, D(PHI)/DS,
 23.C     D(PHI)/DN, D2(PHI)/DS2, D2(PHI)/DNS, D2(R)/DS2, D2(THETA)/DS2
 24.C     ARE ARRAYS OF VALUES DESCRIBING CURVE, VELOCITY POTENTIAL
 25.C     AND THEIR DERIVATIVES.
 26.C  PHIT, PSIT    ARE THE PARTIAL DERIVATIVES OF PHI AND PSI WITH RESPECT
 27.C                TO TIME ALONG THE FREE SURFACE.
 28.C  IWRITE    .LE. 0,  ITERATIVE VALUES OF PHI, PSI, DPHIDX AND DPHIDY
 29.C                       ARE WRITTEN OUT.
 30.C           .GT. 0,  VALUES NOT WRITTEN.
 31.C  INOUT   = 1       POINT INSIDE OF CURVE.
 32.C          = 0              ON
 33.C          =-1             OUTSIDE
 34.C  NP1    = SIZE OF ARRAYS.
 35.C EPSLON  =  A SMALL POSITIVE NUMBER.  IT IS THE CONVERGENCE CRITERION.
 36.C              FOR CONVERGENCE IT IS REQUIRED THAT SUCCESSIVE VALUES
 37.C             OF EACH OUTPUT VARIABLE DIFFER BY LESS THAN EPSLON.
 38.C             HINT:  EPSLON SHOULD BE ABOUT 1.D-5 FOR 60 POINTS ALONG
 39.C               THE PROFILE.
 40.C NMAX     =  MAXIMUM NUMBER OF ITERATIONS.
 41.C              NMAX SHOULD BE ABOUT 50 FOR 60 POINTS AND EPSLON=1.D-5.
 42.C
 43.C OUTPUTS:
 44.C  PHI0      PHI                  AT (X0, Y0)
 45.C  PSI0      PSI
 46.C  DPHIDX    D(PHI)/DX
 47.C  DPHIDY    D(PHI)/DY
 48.C  PHIT0     PARTIAL D(PHI)/DT
 49.C  PSIT0     PARTIAL D(PSI)/DT
 50.C  PRES0     PRESSURE
 51.C  SIMPCO, CO, SI, C, S   WORKING ARRAYS
 52.C  NITER   NUMBER OF ITERATIONS UNTIL CONVERGENCE OR DIVERGENCE.
 53.C
 54.C  EDC  7 AUGUST 1978.
 55.C REVISED 20 SEPT. 1978.
 56.C
 57.C NOTE:   THE SECTION WHICH CALCULATES PHI, PSI, D(PHI)/DX AND D(PHI)/DY
 58.C         FOR POINTS OUTSIDE OF THE BOUNDING CONTOUR MAY NOT BE CONCEPTUALLY
 59.C         CORRECT.  IT HAS ALSO TO BE PROPERLY DEBUGGED.
 60.C
```

```
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION THETA(NP1), R(NP1), DTHDS(NP1), DRDS(NP1),
     1  DSDP(NP1), PHI(NP1), DPHIDS(NP1), DPHIDN(NP1), SIMPCO(NP1),
     2  CO(NP1), SI(NP1), P(NP1), PSI(NP1), C(NP1), S(NP1)
      DIMENSION D2PDS2(NP1), D2PDNS(NP1), D2RDS2(NP1), D2TDS2(NP1)
      DIMENSION PHIT(NP1), PSIT(NP1)
C
      PI = 4.*DATAN(1.D0)
      N = NP1 - 1
      NM1 = N-1
      EPSLN2 = EPSLON**2
      IF (2*INT(FLOAT(N)/2. + .001) .EQ. N) GO TO 12
      WRITE(6, 1001) NP1
      STOP
 1001 FORMAT(///' ERROR RETURN FROM WDWDZ:  NP1 = ', I6,
     1  ' WHICH IS NOT ODD '//)
 12   CONTINUE
      SIMPCO(1) = 1.
      SIMPCO(2) = 4.
      SIMPCO(NP1) = 1.
      DO 15 I=3, NM1, 2
      SIMPCO(I) = 2.
      SIMPCO(I+1) = 4.
 15   CONTINUE
      DELP = P(2) - P(1)
      NITER = 0
      R0 = DEXP(Y0)
      THETA0 = -X0
C
C IS R0 NEAR THE BOUNDARY?  I.E. IS ABS((R - R0)/R) .LE. 0.4?
C
      RHOMIN = 100.
      ITER = 0
      DO 110 I=1, NP1
      T0 = THETA(I) - THETA0
      CO(I) = DCOS(T0)
      SI(I) = DSIN(T0)
      C(I) = R(I)*(1. - R0*CO(I)/R(I))
      S(I) = R0*SI(I)
      RHO2 = C(I)*C(I) + S(I)*S(I)
      IF ((RHO2/(R(I)*R(I))) .LE. 0.16D0) ITER = 1
      IF (RHO2 .GE. RHOMIN) GO TO 110
      RHOMIN = RHO2
      IMIN = I
 110  CONTINUE
 120  CONTINUE
      IF (ITER) 150, 130, 150
C
C NO.  CALCULATE PHI, PSI, D(PHI)/DX AND D(PHI)/DY AT X0, Y0 USING
C NORMAL QUADRATURE ON CAUCHY'S FORMULA.
C
 130  CONTINUE
      PHI0 = 0.
      PSI0 = 0.
      PHIT0 = 0.
      PSIT0 = 0.
      DPHIDX = 0.
      DPHIDY = 0.
      DO 140 I=1, NP1
      RHO2 = C(I)*C(I) + S(I)*S(I)
      A = (C(I)*DRDS(I) + S(I)*R(I)*DTHDS(I))*DSDP(I)/RHO2
      B = (S(I)*DRDS(I) - C(I)*R(I)*DTHDS(I))*DSDP(I)/RHO2
      PHI0 = PHI0 - SIMPCO(I)*(PHI(I)*B - PSI(I)*A)
      PSI0 = PSI0 - SIMPCO(I)*(PHI(I)*A + PSI(I)*B)
      PHIT0 = PHIT0 - SIMPCO(I)*(PHIT(I)*B - PSIT(I)*A)
      PSIT0 = PSIT0 - SIMPCO(I)*(PHIT(I)*A + PSIT(I)*B)
```

```
      PR = DRDS(I)*DPHIDS(I) + R(I)*DTHDS(I)*DPHIDN(I)
      PT = R(I)*DTHDS(I)*DPHIDS(I) - DRDS(I)*DPHIDN(I)
      DPHI0Y = DPHIDY - SIMPCO(I)*((PR*CO(I) - PT*SI(I))*B
     1  + (PR*SI(I) + PT*CO(I))*A)
      DPHI0X = DPHIDX - SIMPCO(I)*((PR*SI(I) + PT*CO(I))*B
     1  - (PR*CO(I) - PT*SI(I))*A)
 140  CONTINUE
      PHI0 = PHI0*DELP/(6.*PI)
      PSI0 = PSI0*DELP/(6.*PI)
      PHIT0 = PHIT0*DELP/(6.*PI)
      PSIT0 = PSIT0*DELP/(6.*PI)
      DPHI0Y = DPHIDY*DELP*R0/(6.*PI)
      DPHI0X = -DPHIDX*DELP*R0/(6.*PI)
      PRES0 = -PHIT0 - (DPHIDX**2 + DPHIDY**2)/2. - Y0
      IF (INOUT .GE. 0) RETURN
      PHI0 = -PHI0
      PSI0 = -PSI0
      PHIT0 = -PHIT0
      PSIT0 = -PSIT0
      DPHI0X = -DPHIDX
      DPHI0Y = -DPHIDY
      PRES0 = -PHIT0 - (DPHIDX**2 + DPHIDY**2)/2. - Y0
      RETURN
C
C YES.  CALCULATE PHI, PSI, D(PHI)/DX AND D(PHI)/DY AT X0, Y0 BY
C ITERATION ON CAUCHY'S FORMULA.
C
 150  CONTINUE
C
C     MAKE A FIRST GUESS BY EXTRAPOLATING FROM THE SURFACE.
C
      I = IMIN
      DPHIDR = DRDS(I)*DPHIDS(I) + R(I)*DTHDS(I)*DPHIDN(I)
      DPHIDT = (R(I)*DTHDS(I)*DPHIDS(I) - DRDS(I)*DPHIDN(I))
      D2PDR2 = (2.*DRDS(I)**2 - 1.)*D2PDS2(I) + 2.*R(I)*DRDS(I)*
     1  DTHDS(I)*D2PDNS(I)
     2  + 2.*DRDS(I)*(D2RDS2(I) - R(I)*DTHDS(I)**2)*DPHIDS(I)
     3  + (R(I)*DTHDS(I)*D2RDS2(I) + R(I)*DRDS(I)*D2TDS2(I)
     4  + 2.*DRDS(I)*DRDS(I)*DTHDS(I) - (R(I)*DTHDS(I))**2*DTHDS(I))
     5  *DPHIDN(I)
      D2PDRT = 2.*R(I)*R(I)*DTHDS(I)*DRDS(I)*D2PDS2(I)
     1  + R(I)*(1. - 2.*DRDS(I)**2)*D2PDNS(I)
     2  + (3.*DRDS(I)**2*DTHDS(I) + R(I)*DRDS(I)*
     3  D2TDS2(I) + R(I)*DTHDS(I)*D2RDS2(I))*R(I)*DPHIDS(I)
     4  +((R(I)*DTHDS(I))**2*DRDS(I) - DRDS(I)**3 - 2.*R(I)*DRDS(I)
     5  *D2RDS2(I))*DPHIDN(I)
      CC = DCOS(THETA(I) - THETA0)
      SS = DSIN(THETA(I) - THETA0)
      A = R0*CC/R(I) - 1.
      B = R0*SS
      AA = R0*DCOS(2.*(THETA(I) - THETA0)) - R(I)*CC
      BB = R0*DSIN(2.*(THETA(I) - THETA0)) - R(I)*SS
      PHI0 = PHI(I) + R(I)*DPHIDR*A - B*DPHIDT
      PSI0 = PSI(I) - DPHIDR*B - R(I)*DPHIDT*A
      PHIT0 = PHIT(I)
      PSIT0 = PSIT(I)
      DPHI0X = DPHIDR*SS + DPHIDT*CC + D2PDR2*BB +
     1  (D2PDRT - DPHIDT)*AA/R(I)
      DPHI0Y = DPHIDR*CC - DPHIDT*SS + D2PDR2*AA
     1  - (D2PDRT - DPHIDT)*BB/R(I)
      IF (RHOMIN/(R(IMIN)*R(IMIN)) .LE. EPSLN2) GO TO 190
      DO 160 I=1, NP1
      RHO2 = C(I)*C(I) + S(I)*S(I)
      A = (C(I)*DRDS(I) + S(I)*R(I)*DTHDS(I))*DSDP(I)/RHO2
      B = (S(I)*DRDS(I) - C(I)*R(I)*DTHDS(I))*DSDP(I)/RHO2
      C(I) = A
```

```
      S(I) = B
  160 CONTINUE
      IF (INOUT .LT. 0) GO TO 270
  170 CONTINUE
      IF (NITER .LT. NMAX) GO TO 175
      WRITE(6, 1002) NMAX
      RHOMIN = 0.
      GO TO 150
 1002 FORMAT(/' ERROR RETURN FROM WOWDZ:  NO CONVERGENCE AFTER',
     1  I4, ' ITERATIONS. '/)
  175 CONTINUE
      NITER = NITER + 1
       IF (IWRITE .GT. 0) GO TO 178
      A = -R0*DPHIDX
      B = R0*DPHIDY
      WRITE(6, 1004) NITER, PHI0, PSI0, A, B, PHIT0, PSIT0
 1004  FORMAT( ' ITER. =', I3, '  PHI0 =', 1PD10.2,
     1  '  PSI0 =', D10.2, '  DPHIDX =', D10.2, '  DPHIDY =',
     2  D10.2,  '  PHIT0 =', D10.2, '  PSIT0 =', D10.2)
  178 CONTINUE
      PHIOLD = PHI0
      PSIOLD = PSI0
      PHITOL = PHIT0
      PSITOL = PSIT0
      PHIXO = DPHIDX
      PHIYO = DPHIDY
      PHI0 = 0.
      PSI0 = 0.
      PHIT0 = 0.
      PSIT0 = 0.
      DPHIDX = 0.
      DPHIDY = 0.
      DO 180 I=1, NP1
      PHI0 = PHI0 - SIMPCO(I)*((PHI(I) - PHIOLD)*S(I)
     1  - (PSI(I) - PSIOLD)*C(I))
      PSI0 = PSI0 - SIMPCO(I)*((PHI(I) - PHIOLD)* C(I)
     2  + (PSI(I) - PSIOLD)*S(I))
      PHIT0 = PHIT0 - SIMPCO(I)*((PHIT(I) - PHITOL)*S(I)
     1  - (PSIT(I) - PSITOL)*C(I))
      PSIT0 = PSIT0 - SIMPCO(I)*((PHIT(I) - PHITOL)*C(I)
     1  + (PSIT(I) - PSITOL)*S(I))
      PR = DRDS(I)*DPHIDS(I) + R(I)*DTHDS(I)*DPHIDN(I)
      PT = R(I)*DTHDS(I)*DPHIDS(I) - DRDS(I)*DPHIDN(I)
      DPHIDY = DPHIDY - SIMPCO(I)*((PR*CO(I) - PT*SI(I) - PHIYO)
     1 *S(I) + (PR*SI(I) + PT*CO(I) - PHIXO)*C(I))
      DPHIDX = DPHIDX - SIMPCO(I)*((PR*SI(I) + PT*CO(I) - PHIXO)
     1 *S(I) - (PR*CO(I) - PT*SI(I) - PHIYO)*C(I))
  180 CONTINUE
      PHI0 = PHI0*DELP/(6.*PI)
      PSI0 = PSI0*DELP/(6.*PI)
      PHIT0 = PHIT0*DELP/(6.*PI)
      PSIT0 = PSIT0*DELP/(6.*PI)
      DPHIDX = DPHIDX*DELP/(6.*PI)
      DPHIDY = DPHIDY*DELP/(6.*PI)
      F1 = DABS(PHI0)
      F2 = DABS(PSI0)
      F3 = R0*DABS(DPHIDX)
      F4 = R0*DABS(DPHIDY)
      F5 = DABS(PHIT0)
      F6 = DABS(PSIT0)
      PHI0 = PHI0 + PHIOLD
      PSI0 = PSI0 + PSIOLD
      PHIT0 = PHIT0 + PHITOL
      PSIT0 = PSIT0 + PSITOL
      DPHIDX = DPHIDX + PHIXO
      DPHIDY = DPHIDY + PHIYO
```

```
      IF (E1 .GE. EPSLON) GO TO 170
      IF (E2 .GE. EPSLON) GO TO 170
      IF (E3 .GE. EPSLON) GO TO 170
      IF (E4 .GE. EPSLON) GO TO 170
      IF (E5 .GE. EPSLON) GO TO 170
      IF (E6 .GE. EPSLON) GO TO 170
  190 CONTINUE
      DPHIDX = -DPHIDX*R0
      DPHIDY = DPHIDY*R0
      PRESQ = -PHIT0 - (DPHIDX**2 + DPHIDY**2)/2. - Y0
      RETURN
C
C      ITERATIONS FOR POINTS OUTSIDE OF CURVE.
C
  270 CONTINUE
      IF (NITER .LT. NMAX) GO TO 275
      WRITE(6, 1002) NMAX
      RHOMIN = 0.
      GO TO 150
  275 CONTINUE
      NITER = NITER + 1
       IF (IWRITE .GT. 0) GO TO 278
      A = -R0*DPHIDX
      B = R0*DPHIDY
      WRITE(6, 1004) NITER, PHI0, PSI0, A, B, PHIT0, PSIT0
  278 CONTINUE
      PHIOLD = PHI0
      PSIOLD = PSI0
      PHITOL = PHIT0
      PSITOL = PSIT0
      PHIXO = DPHIDX
      PHIYO = DPHIDY
      PHI0 = 0.
      PSI0 = 0.
      PHIT0 = 0.
      PSIT0 = 0.
      DPHIDX = 0.
      DPHIDY = 0.
      DO 280 I=1, NP1
      PHI0 = PHI0 - SIMPCO(I)*((PHI(I) - PHIOLD)*S(I)
     1  - (PSI(I) - PSIOLD)*C(I))
      PSI0 = PSI0 - SIMPCO(I)*((PHI(I) - PHIOLD)* C(I)
     2  + (PSI(I) - PSIOLD)*S(I))
      PHIT0 = PHIT0 - SIMPCO(I)*((PHIT(I) - PHITOL)*S(I)
     1  - (PSIT(I) - PSITOL)*C(I))
      PSIT0 = PSIT0 - SIMPCO(I)*((PHIT(I) - PHITOL)*C(I)
     1  + (PSIT(I) - PSITOL)*S(I))
      PR = DRDS(I)*DPHIDS(I) + R(I)*DTHDS(I)*DPHIDN(I)
      PT = R(I)*DTHDS(I)*DPHIDS(I) - DRDS(I)*DPHIDN(I)
      DPHIDY = DPHIDY - SIMPCO(I)*((PR*CO(I) - PT*SI(I) - PHIYO)
     1 *S(I) + (PR*SI(I) + PT*CO(I) - PHIXO)*C(I))
      DPHIDX = DPHIDX - SIMPCO(I)*((PR*SI(I) + PT*CO(I) - PHIXO)
     1 *S(I) - (PR*CO(I) - PT*SI(I) - PHIYO)*C(I))
  280 CONTINUE
      PHI0 = -PHI0*DELP/(6.*PI)
      PSI0 = -PSI0*DELP/(6.*PI)
      PHIT0 = -PHIT0*DELP/(6.*PI)
      PSIT0 = -PSIT0*DELP/(6.*PI)
      DPHIDX = -DPHIDX*DELP/(6.*PI)
      DPHIDY = -DPHIDY*DELP/(6.*PI)
      E1 = DABS(PHI0 - PHIOLD)
      E2 = DABS(PSI0 - PSIOLD)
      E3 = R0*DABS(DPHIDX - PHIXO)
      E4 = R0*DABS(DPHIDY - PHIYO)
      E5 = DABS(PHIT0 - PHITOL)
      E6 = DABS(PSIT0 - PSITOL)
```

```
      IF (E1 .GE. EPSLON) GO TO 270
      IF (E2 .GE. EPSLON) GO TO 270
      IF (E3 .GE. EPSLON) GO TO 270
      IF (E4 .GE. EPSLON) GO TO 270
      IF (E5 .GE. EPSLON) GO TO 270
      IF (E6 .GE. EPSLON) GO TO 270
  290 CONTINUE
      DPHIDX = -DPHIDX*R0
      DPHIDY = DPHIDY*R0
      PRES0 = -PHIT0 - (DPHIDX**2 + DPHIDY**2)/2. - Y0
      RETURN
      END
```

II.F.2. Graphical Routines

ADDPLT (N, X, Y, IPEN, ICHAR, CHRSIZ, ILINE)

Adds a graph to axes previously drawn by BOXPLT. The N points of the array Y are plotted against the array X.

Inputs:

N	number of points to be connected by straight lines
X (N)	array of abscissa values
Y (N)	array of ordinate values
IPEN	IOS pen number
ICHAR	IOS character code for character to be drawn at each point (X,Y)
CHRSIZ	character size in millimeters
ILINE	line drawing parameter. 0 is no line; 1 is a straight line

Outputs: A graph of Y versus X

Subroutines called:

IOS Grafix: LINSEL, NIBSEL, CHASEL, MARSEL, PLOLD2

```
DATE = 24/11/78      TIME =  1.25. 8

FILE = SDMAINDR.ADDPLT  (DG)

      SUBROUTINE ADDPLT(N, X, Y, IPEN, ICHAR, CHRSIZ, ILINE)
C   THIS SUBROUTINE ADDS A GRAPH TO AXES AND GRAPHS PREVIOUSLY DRAWN
C BY BOXPLT.  THE N POINTS OF THE ARRAY Y ARE PLOTTED AGAINST THE ARRAY X.
C IPEN IDENTIFIES THE PEN AS EITHER 1, 2 OR 3.
C ICHAR IS THE CHARACTER CODE.
C CHRSIZ IS THE CHARACTER SIZE IN MM.
C ILINE SELECTS THE LINE TYPE TO BE DRAWN BETWEEN POINTS.  0 IS NO LINE.
C 1 IS A STRAIGHT LINE.
C
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION X(1), Y(1)
      CALL LINSEL(ILINE)
      CALL NIBSEL(IPEN, 2)
      CALL NIBSEL(IPEN, 10)
      CALL CHASEL(SNGL(CHRSIZ))
      CALL MARSEL(ICHAR)
      CALL PLOLD2(X, Y, N)
      RETURN
      END
```

ARROW (X, Y, DX, DY, HEADSZ, IPEN)

Draws an arrow from (X,Y) to (X+DX, Y+DY) in user units in a box previously drawn by BOXPLT or a region set up by other IOS Grafix commands.

Inputs:

X	x
Y	y
DX	dx
DY	dy
HEADSZ	size of arrowhead in user units
IPEN	IOS pen number

Outputs:

A graphical arrow

Subroutines called:

IOS Grafix: PUTLA2, NEWPEN, LINLE2

```
DATE = 20/11/76     TIME =  1.25. 6

FILE = SOMAINDR.ARROW   (DG)

      SUBROUTINE ARROW(X, Y, DX, DY, HEADSZ, IPEN)
C
C DRAWS AN ARROW FROM (X, Y) TO (X+DX, Y+DY).
C THE SIZE OF THE ARROWHEAD IN USER UNITS IS HEADSZ.
C IPEN IDENTIFIES THE PEN NUMBER.
C
      IMPLICIT REAL*8(A-H, O-Z)
      DATA CA, SA/0.965925826 D0, 0.258819045 D0/
C
C MOVE TO (X, Y) AND DRAW THE SHAFT.
C
      CALL PUTLA2(SNGL(X), SNGL(Y), IERR)
      IF ( IERR .NE. 0) RETURN
      X2 = X+DX
      Y2 = Y+DY
       CALL NEWPEN(IPEN)
      CALL LINLE2(X, Y, X2, Y2)
C
C DRAW THE ARROWHEAD
C
      CALL PUTLA2(SNGL(X2), SNGL(Y2), IERR)
      IF (IERR .NE. 0) RETURN
      SHAFT = DSQRT(DX*DX + DY*DY)
      CT = DX/SHAFT
      ST = DY/SHAFT
      X3 = X2 - HEADSZ*(CT*CA + ST*SA)
      Y3 = Y2 - HEADSZ*(ST*CA - CT*SA)
      CALL LINLE2(X2, Y2, X3, Y3)
      CALL PUTLA2(SNGL(X2), SNGL(Y2), IERR)
      X4 = X2 - HEADSZ*(CT*CA - ST*SA)
      Y4  = Y2 - HEADSZ*(ST*CA + CT*SA)
      CALL LINLE2(X2, Y2, X4, Y4)
      RETURN
      END
```

BOXPLT (N, X, Y, IPEN, ICHAR, CHRSIZ, ILINE, ITOP, IBOT, ILEFT, IRIGHT, XMIN, XMAX, XINC, XLEN, YMIN, YMAX, YINC, YLEN)

Draws all or part of a box with or without labels and plots the N points of Y versus X which fall within the box. The pen number and plot character type and size are under the user's control. Symbols may be drawn at each point.

Inputs:

N	number of points to be plotted. If N=0, box and labels are drawn with no X-Y plotting
X (N)	abscissa values
Y (N)	ordinate values
IPEN	IOS pen number
ICHAR	IOS character code. A character can be drawn at each data point.
CHRIZ	character size in mm
ILINE	line drawing parameter, 0 is no line, 1 is a straight line
ITOP IBOT ILEFT IRIGHT	control axis drawing and labeling =0, no axis, no labels =1, axis, no labels =2, axis and labels with checks done for label overlapping =3, axis and labels, no overlap checking

XMIN	minimum limit of x-axis
XMAX	maximum limit of X-axis
YMIN	minimum limit of y-axis
YMAX	maximum limit of y-axis
XINC	X - axis increments
YINC	Y - axis increments
XLEN	length of X-axis in mm
YLEN	length of Y-axis in mm

Outputs:

a graphof Y versus X within a box

Subroutines called:

EDC subroutines: ADDPLT

IOS Grafix: PICCIE, GRAFIX, SHIFT2, MOVTO2, LINTO2, DEFLA2, ANMSEL, GRISEL, AXILE2

```
DATE = 24/11/76     TIME =  1.25. 8

FILE = SDMAINDR.BOXPLT  (DG)

      SUBROUTINE BOXPLT(N, X, Y, IPEN, ICHAR, CHRSIZ, ILINE, ITOP,
     1 IBOT, ILEFT, IRIGHT, XMIN, XMAX, XINC, XLEN, YMIN,
     2  YMAX, YINC, YLEN)
C   THIS SUBROUTINE DRAWS ALL OR PART OF A BOX WITH OR WITHOUT LABELS AND
C PLOTS THE N POINTS OF Y VERSUS X WHICH FALL WITHIN THE BOX.  THE PEN
C NUMBER, PLOT CHARACTER TYPE AND SIZE CAN BE ALTERED.
C
C N = NUMBER OF POINTS TO BE PLOTTED (N MAY BE 0).
C X, Y ARE ARRAYS PLOTTED.
C IPEN = 1, 2 OR 3.
C ICHAR IS IOS CHARACTER CODE.
C CHRSIZ IS CHARACTER SIZE IN MM.
C ILINE = 0, NO LINE IS DRAWN BETWEEN POINTS.
C       = 1, A STRAIGHT LINE IS DRAWN.
C ITOP, IBOT, ILEFT AND IRIGHT CONTROL AXIS DRAWING AND LABELING.
C I--- = 0, NO AXIS DRAWN, NO LABELS.
C      = 1, AXIS DRAWN, NO LABELS.
C      = 2, AXIS DRAWN, LABELS BUT CHECKING DONE FOR OVERLAPS.
C      = 3, AXIS DRAWN, LABELS WITH NO OVERLAP CHECKING.
C XMIN AND XMAX = USER LIMITS OF X AXIS.
C YMIN AND YMAX =                Y    .
C XINC AND YINC = INCREMENTS BETWEEN TICK MARKS.
C XLEN AND YLEN = AXIS LENGTHS IN MM.
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XLN, YLN
      DIMENSION X(1), Y(1)
      CALL PICCIE
      XLN = SNGL(XLEN)
      YLN = SNGL(YLEN)
      CALL GRAFIX(XLN, YLN, 0.)
C
C  CENTER THE GRAPH IN THE TEKTRONIX SCREEN.  IF TOO LARGE, ASSUME
C   IS PLOTTER AND DO NOT CENTER.
C
      XL = 200.
      YL = 140.
      IF (ILEFT .LT. 2) GO TO 10
      XL = 170.
      CALL SHIFT2(20., 0.)
   10 IF (IBOT .LT. 2) GO TO 20
      YL = 120.
      CALL SHIFT2(0., 10.)
   20 IF (XLEN .GE. XL) GO TO 30
      CALL SHIFT2((SNGL(XL - XLEN))/2., 0.)
   30 IF (YLEN .GE. YL) GO TO 40
      CALL SHIFT2(0., (SNGL(YL - YLEN))/2.)
   40 CONTINUE
C
C DRAW THE AXES AND ANNOTATE THEM IF DESIRED.
C
      IF (ITOP .EQ. 0) GO TO 50
      CALL MOVTO2(0., YLN)
      CALL LINTO2(XLN, YLN)
   50 CONTINUE
      IF (IBOT .EQ. 0) GO TO 60
      CALL MOVTO2(0., 0.)
      CALL LINTO2(XLN, 0.)
   60 CONTINUE
      IF (ILEFT .EQ. 0) GO TO 70
```

```
      CALL MOVTO2(0., 0.)
      CALL LINTO2(0., YLEN)
70    CONTINUE
      IF (IRIGHT .EQ. 0) GO TO 80
      CALL MOVTO2(XLEN, 0.)
      CALL LINTO2(XLEN, YLEN)
80    CONTINUE
      CALL DEFLA2(SNGL(XMIN), SNGL(XMAX), SNGL(YMIN), SNGL(YMAX))
      CALL ANMSEL(0., SNGL(YLEN/120.), 0., SNGL(YLEN/120.))
      CALL GRISEL(IBOT, ITOP, 0)
      CALL AXILE2(XINC, 0, 1)
      CALL ANMSEL(0., SNGL(XLEN/120.), 0., SNGL(XLEN/120.))
      CALL GRISEL(ILEFT, IRIGHT, 0)
      CALL AXILE2(YINC, 0, 2)
C
C DRAW THE GRAPH.
C
      CALL ADDPLT(N, X, Y, IPEN, ICHAR, CHRSIZ, ILINE)
      RETURN
      END
```

CREST (N, IPOSN, IBOX, TIME, SPEED, PT, XO, YO, DXDP, DYDP, X, Y, XMIN, XLEN, XINC, XLENMM, YMIN, YLEN, YINC, YLENMM, WVLNTH)

Plots that part of the wave crest region at a given time which falls within a box moving with a specified speed. A cubic spline parameterized by the point number is fitted to the curve, and a smooth profile is drawn through it.

Inputs:

N	number of points along entire profile
IPOSN	parameter to specify crest position within box. <0 means crest at left side of box. $=0$ means crest in center of box. >0 means crest at right side of box.
IBOX	box drawing parameter. <0 means no new box, plot profile within old box but moved an amount (SPEED x t). $=0$ means draw new box if crest falls outside box. ≥ 1 means draw new box and plot profile.
TIME	time of plotting
SPEED	speed of box in X-direction
PT(N)	array of point number, p
XO(N)	array of X coordinates
YO(N)	array of Y coordinates
DXDP(N)	$\frac{dx}{dp}$

DYDP(N)	$\frac{dy}{dp}$
XLEN	length of X-axis in user units
XINC	increments of X-axis
XLENMM	length of X-axis in mm
YLEN	length of Y-axis in user units
YINC	increments of Y-axis
YLENMM	length of Y-axis in mm
WVLNTH	wavelength in user units

Outputs:

	A graph of the wave crest within a box
X(N)	working array containing X values
Y(N)	working array containing Y values
XMIN	minimum X coordinate of box
YMIN	minimum Y coordinate of box

Subroutines called:

EDC subroutines: MODULO, BOXPLT, PASPLT

IOS Grafix: NIBSEL, CHASEL

```
DATE = 29/11/78     TIME =  1.25. 6

FILE = SDMAILDR.CREST   (DG)

      SUBROUTINE CREST(N, IPOSN, IBOX, TIME, SPEED, PT, X0, Y0,
     1  DXDP, DYDP, X, Y, XMIN, XLEN, XINC, XLENMM, YMIN, YLEN,
     2  YINC, YLENMM, WVLNTH)
C
C   PLOTS MAGNIFIED WAVE CREST IN BOX MOVING WITH A GIVEN SPEED AT A GIVEN
C TIME.  THE PROFILE IS SPECIFIED AT N POINTS (X0, Y0) IN TERMS OF A
C PARAMETER PT.  GIVEN THE DERIVATIVES, DXDP AND DYDP, AT (X0, Y0)
C WITH RESPECT TO PT, A CUBIC SPLINE IS FITTED.  X AND Y ARE WORK
C ARRAYS.  THE PROGRAM DETERMINES THE COORDINATES (XMIN, YMIN) OF
C THE LOWER LEFT-HAND CORNER OF THE BOX.  THE AXES ARE OF LENGTH
C XLEN AND YLEN IN USER UNITS AND XLENMM AND YLENMM IN MILLIMETERS.
C THE AXIS INCREMENTS ARE XINC AND YINC.
C WVLNTH IS THE WAVELENGTH IN USER UNITS.
C
C IPOSN = -, CREST POSITIONED AT LEFT OF BOX.
C       = 0,                     CENTRE
C       = +,                     RIGHT
C IBOX  = -, DO NOT DRAW NEW BOX, JUST PLOT PART OF PROFILE WITHIN BOX.
C       = 0, DRAW NEW BOX IF CREST OUTSIDE OLD BOX, PLOT PROFILE.
C       = +, DRAW NEW BOX AND PLOT PROFILE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X0(N), Y0(N), X(N), Y(N)
     1  , PT(N), DXDP(N), DYDP(N), P(2), X1(2), DXDP1(2),
     2  Y1(2), DYDP1(2)
      PI = 4.*DATAN(1.D0)
C
C  FIND THE CREST
C
      PER = WVLNTH
      YM = 0.
      DO 120 I=1, N
      Y(I) = Y0(I)
      X(I) = X0(I) - SPEED*TIME
  116 IF (Y(I) .LT. YM) GO TO 120
      YM = Y(I)
      JCREST = I
  120 CONTINUE
C
C  CREST POSITION IN BOX
C
      IF (IBOX .LT. 0) GO TO 190
      IF (IPOSN) 130, 150, 170
C
C  CREST POSITIONED NEAR LEFT OF BOX.
C
  130 CONTINUE
      JS = JCREST
      XJ = X(JS)
      CALL MODULO(XJ, 0.D0, PER)
      XMP = XJ - XINC/2.
      DELX1 = -XINC/2.
      DELX2 =  XINC/2.
      GO TO 180
C
C  CREST POSITIONED NEAR CENTRE OF BOX.
C
  150 CONTINUE
      JS = JCREST
      XJ = X(JS)
```

```
      CALL MODULO(XJ, 0.D0, PER)
      XMP = XJ - XLEN/2.
      DELX1 = XINC/2.
      DELX2 = XINC/2.
      GO TO 180
C
C  CREST POSITIONED NEAR RIGHT OF BOX.
C
  170 CONTINUE
      JS = JCREST
      XJ = X(JS)
      CALL MODULO(XJ, 0.D0, PER)
      XMP = XJ + XINC/2. - XLEN
      DELX1 =  XINC/2.
      DELX2 = -XINC/2.
  180 CONTINUE
C
C  SHOULD A NEW BOX BE DRAWN?
C
      IF (IBOX .GT. 0) GO TO 185
      XJS = X(JS)
      CALL MODULO(XJS, XMIN, PER)
      IF (XJS .LT. (XMIN + DELX1)) GO TO 185
      IF (XJS .GT. (XMAX - DELX2)) GO TO 185
      IF (Y(JCREST) .GT. (YMAX - YINC/4.)) GO TO 185
      IF (Y(JCREST) .LT. (YMIN + YINC/4.)) GO TO 185
      GO TO 190
C
C  DRAW THE BOX.
C
  185 CONTINUE
      XMIN = XMP
      XMAX = XMIN + XLEN
      YMAX = Y(JS) + YINC/2.
      YMIN = YMAX - YLEN
      CALL BOXPLT(0, X, Y, 1, 0, 0.5D0, 1, 1, 2, 2, 1, XMIN, XMAX, XINC,
     1  XLENMM, YMIN, YMAX, YINC, YLENMM)
      WRITE(6, 1001) XMIN, XMAX, XINC
      WRITE(6, 1002) YMIN, YMAX, YINC
C
C  MAKE X BETWEEN XMIN AND XMIN + PER.
C
  190 CONTINUE
      DO 200 I=1, N
      CALL MODULO(X(I), XMIN, PER)
  200 CONTINUE
C
C  PLOT THE PROFILE IN THE BOX.
C
      CALL NIBSEL(1, 2)
      CALL NIBSEL (1, 10)
      CALL CHASEL (0.5)
      INTRVL = 10
      P(1) = PT(1)
      P(2) = PT(2)
      JM1 = N
      DO 260 J=1, N
      X1(1) = X(JM1)
      X1(2) = X(J)
      Y1(1) = Y(JM1)
      Y1(2) = Y(J)
      DXDP1(1) = DXDP(JM1)
      DXDP1(2) = DXDP(J)
      DYDP1(1) = DYDP(JM1)
      DYDP1(2) = DYDP(J)
      IF ((X(J) - X(JM1)) .GT. -PER/2.) GO TO 252
```

```
      X1(2) = X1(2) + PER
      CALL PASPLT(INTRVL, 0, 7, 0, 2, P, X1, DXDP1, Y1, DYDP1)
      X1(1) = X1(1) - PER
      X1(2) = X(J)
      GO TO 254
  252 IF ((X(J) - X(JM1)) .LT. PER/2.) GO TO 254
      X1(2) = X1(2) - PER
      CALL PASPLT(INTRVL, 0, 7, 0, 2, P, X1, DXDP1, Y1, DYDP1)
      X1(1) = X1(1) + PER
      X1(2) = X(J)
  254 CONTINUE
      CALL PASPLT(INTRVL, 0, 7, 0, 2, P, X1, DXDP1, Y1, DYDP1)
      JM1 = J
  260  CONTINUE
      RETURN
 1001 FORMAT(' X AXIS DRAWN FROM ', F10.6, ' TO ', F10.6,
     1  ' IN INCREMENTS OF ', F10.6)
 1002 FORMAT( ' Y AXIS DRAWN FROM ', F10.6, ' TO ', F10.6,
     1  ' IN INCREMENTS OF ', F10.6)
      END
```

PASPLT (INTRVL, PRICH1, PRICHR, SECCHR, N, P, X, DXDP, Y, DYDP)

Given a collection of N points (X, Y) specified in terms of a parameter P which increases monotonically along the curve and given the derivatives of X and Y with respect to P at these points, DXDP and DYDP, this routine draws a cubic spline curve through them. The line between any two data points (or knots) is made up of INTRVL straight lines. At each knot, (X, Y), a plot character can be drawn whose IOS code number if PRICHR. The very first point to be drawn has the character code number PRICH1. At the intermediate points (INTRVL-1 between each two data points) the plot character SECCHR can be drawn.

Inputs:

INTRVL	number of straight line intervals between data points
PRICH1	IOS character code of first data point
PRICHR	IOS character code of remaining data points
SECCHR	IOS character code of interpolated data points
N	number of data points
P(N)	array of parameter values. Must be increasing.
X(N)	abscissa values, X
DXDP(N)	$\frac{dx}{dp}$
Y(N)	ordinate values, Y
DYDP(N)	$\frac{dy}{dp}$

Outputs:

A smooth graph of Y versus X

Subroutines called:

EDC subroutines: INTERP

IOS Grafix: LINSEL, MARSEL, POILD2

```
      SUBROUTINE PASPLT(INTRVL, PRICH1, PRICHR, SECCHR, N,
     1  P, X, DXDP, Y, DYDP)
C
C   GIVEN A COLLECTION OF N POINTS, (X, Y), SPECIFIED IN TERMS OF A PARAMETER P
C AND GIVEN THE DERIVATIVES OF X AND Y WITH RESPECT TO P , DXDP AND DYDP, AT
C THESE POINTS, THIS SUBROUTINE DRAWS A CUBIC SPLINE THROUGH THEM.  THE LINE
C BETWEEN ANY 2 DATA POINTS IS MADE UP OF INTRVL STRAIGHT LINES.  AT EACH
C KNOT, (X, Y), A PLOT CHARACTER CAN BE DRAWN WHOSE CODE NUMBER IS PRICHR.
C AT THE INTERMEDIATE POINTS (OF WHICH THERE ARE INTRVL - 1) THE PLOT
C CHARACTER SECCHR CAN BE DRAWN.  THE VERY FIRST POINT TO BE PLOTTED HAS THE
C PLOT CHARACTER PRICH1 DRAWN.
C
      IMPLICIT REAL*8(A-H, O-Z)
      INTEGER PRICH1, PRICHR, SECCHR
      DIMENSION P(N), X(N), DXDP(N), Y(N), DYDP(N)
      CALL LINSEL(0)
      CALL MARSEL(PRICH1)
      CALL POILD2(X(1), Y(1))
      CALL LINSEL(1)
      NM1 = N-1
      JE = INTRVL - 1
      DO 50 I=1, NM1
      IF (JE .LT. 1) GO TO 30
      IM2 = I - 2
      PP = P(I)
      DP = (P(I+1) - P(I))/FLOAT(INTRVL)
      DO 20 J=1, JE
      PP = PP + DP
      CALL INTERP(IM2, N, P, X, DXDP, PP, XI, DXDPI)
      CALL INTERP(IM2, N, P, Y, DYDP, PP, YI, DYDPI)
      CALL MARSEL(SECCHR)
      CALL POILD2(XI, YI)
 20   CONTINUE
 30   CONTINUE
      CALL MARSEL(PRICHR)
      CALL POILD2(X(I+1), Y(I+1))
 50   CONTINUE
      RETURN
      END
```

II.G. Data Tapes

INTVEL takes as its input data the results of various numerical experiments performed with periodic, deep-water waves. These are on tapes M907 and M908 at IOS. The data files come from numerical experiments beginning with one of 3 types of initial conditions. The first is that studied by Longuet-Higgins and Cokelet (1976). They take as initial conditions a steady wave of steepness ak = 0.399671 corresponding to the value of the expansion parameter $\epsilon^2 = 0.80$ (see Cokelet, E. D. 1977 Steep gravity waves in water of arbitrary uniform depth, Phil. Trans. R. Soc. Lond. A. 286, 183-230). The waves are forced by the surface pressure condition

$$\begin{aligned} p &= p_0 \sin t \ \sin(x - ct), && 0 \le t \le \pi \\ &= 0, && \pi \le t. \end{aligned}$$

Here c = 1.08210, the speed of the initial steady wave.

The second type of initial condition is a simple sinusoidal wave at t = 0 with no pressure forcing

$$\begin{aligned} \eta &= a \sin x \\ \phi &= -a \cos x \\ p &= 0. \end{aligned}$$

Such a wave is steady only when the steepness ak→0. In general the wave profile will distort, and if the amplitude is large enough breaking will ensue.

The third type of initial condition is a pair of slightly perturbed steady waves with no pressure forcing. These were studied by Longuet-Higgins and Cokelet (1979).

We shall give more details of the magnetic tapes, the file structure and the files. Tapes M907 and M908 (duplicate, back-up copy of M907) are 9 track, 6250 bpi, odd parity, standard IBM label magnetic tapes. The data record format is variable blocked, sequential (VBS), and the block size is 2498 bytes. The record length is X since the files were written with unformatted FORTRAN WRITE statements. As a result the files can only be read by a FORTRAN job run on an

IBM 360/195 computer or on some compatible machine.

All of the files are in a single format, and all of the real numbers are in IBM double precision (REAL*8). Typical FORTRAN type specification, DIMENSION and READ statements are as follows:

```
IMPLICIT REAL*8 (A-H, O-Z)

DIMENSION THETA (61), R(61), PHI (61), PRES (61),

1  DPHIDS (61), DPHIDN (61), DSDP (61), S (68),

2  DTHDS (61), DRDS (61), DPHIDT (61), DTHDT (61), DRDT (61)

READ (8) ITIME, TIME, DELTAT, ENKIN, ENPOT, ENERGY, YBAR

READ (8) THETA, R, PHI, PRES, DPHIDS, DPHIDN, DSDP,

     S, DTHDS, DRDS, DPHIDT, DTHDT, DRDT
```

The variables are defined as follows:

ITIME	timestep number	
TIME	t	
DELTAT	Δ t, timestep increment	
ENKIN	average kinetic energy, T.	$T = \frac{1}{2\pi} \int_0^{2\pi} \int_{-\infty}^{\eta} \frac{1}{2} \rho (u^2+v^2)\, dy\, dx$
ENPOT	average potential energy, V	$V = \frac{1}{2\pi} \int_0^{2\pi} \frac{1}{2} \rho g \eta^2\, dx$

ENERGY	total average energy, E = T + V
YBAR	mean free-surface elevation, $\bar{\eta}$. $\bar{\eta} = \frac{1}{2\pi}\int_0^{2\pi} \eta \, dx$. $y = \eta(x,t)$ is the free-surface. $\bar{\eta}$ should be zero, and the fact that it is not provides a check on the numerical results. Note $\bar{\eta} \neq 0$ was not taken into account for calculating V.
THETA	θ
R	r
PHI	ϕ
PRES	pressure
DPHIDS	$\frac{\partial \phi}{\partial s}$
DPHIDN	$\frac{\partial \phi}{\partial n}$
DSDP	$\frac{ds}{dp}$
S	s. Note dimension of this array is N+8.
DTHDS	$\frac{d\theta}{ds}$

DRDS	$\frac{dr}{ds}$
DPHIDT	$\frac{D\phi}{Dt}$
DTHDT	$\frac{D\theta}{Dt}$
DRDT	$\frac{Dr}{Dt}$

There are either N = 60, 90, or 120 points along the wave profile. Point 1 and point N + 1 are the same, but one wavelength ($=2\pi$) apart. The file name has the value of N contained within it. The accompanying tape examination (XTAPE) gives the file labels, names and storage information. Files BREAK60... and SINE60... have N = 60. The rest of the files have N after the last decimal point in the file name.

The first 6 files are of the type with surface pressure forcing. More details are as follows:

Label	Name	Po	Timesteps	Time Span
1	BREAK60.MK1N1	0.100	0-79	0-2.91
2	BREAK60.MK1N6	0.100	194-232	5.04-5.29
3	BREAK60.MK2N5	0.0729	163-259	5.03-5.90
4	BREAK60.MK3N6	0.126	197-273	4.82-5.22
5	BREAK60.MK10	0.206	0-183	0-3.79
6	BREAK60.MK4	0.146	0-368	0-5.38

```
XTAPE   CONTENTS OF TAPE M9027   AT 16.24 ON 19 NOV 78.  OWNER IS

THE TAPE IS 9-TRACK, 6250 BPI, ODD PARITY

       DATA SET NAME          CREATED ON     EXP DATE    BY JOB/STEP        BLOCKS   RECFM   LRECL   BLKSIZE

  1    BREAK60.MK1N1          16 NOV 78     0 JAN 00   XNSDTAPE/            202     VBS      X       2498
  2    BREAK60.MK1N6          16 NOV 78     0 JAN 00   XNSDTAPE/            102     VBS      X       2498
  3    BREAK60.MK2N5          16 NOV 78     0 JAN 00   XNSDTAPE/            261     VBS      X       2498
  4    BREAK60.MK3N6          16 NOV 78     0 JAN 00   XNSDTAPE/            201     VBS      X       2498
  5    BREAK60.MK10           16 NOV 78     0 JAN 00   XNSDTAPE/            480     VBS      X       2498
  6    BREAK60.MK4            17 NOV 78     0 JAN 00   XNSDTAPE/            966     VBS      X       2498
  7    SINE60.MK1             17 NOV 78     0 JAN 00   XNSDTAPE/            346     VBS      X       2498
  8    SINE60.MK2             17 NOV 78     0 JAN 00   XNSDTAPE/            632     VBS      X       2498
  9    SINE60.MK3             17 NOV 78     0 JAN 00   XNSDTAPE/            401     VBS      X       2498
 10    SINE60.MK4             17 NOV 78     0 JAN 00   XNSDTAPE/            596     VBS      X       2498
 11    SINE60.MK5             17 NOV 78     0 JAN 00   XNSDTAPE/            505     VBS      X       2498
 12    SINE60.MK6             17 NOV 78     0 JAN 00   XNSDTAPE/            650     VBS      X       2498
 13    SINE60.MK7             17 NOV 78     0 JAN 00   XNSDTAPE/            710     VBS      X       2498
 14    N12AK10.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            339     VBS      X       2498
 15    N32AK10.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            266     VBS      X       2498
 16    N12AK20.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            336     VBS      X       2498
 17    N32AK20.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            297     VBS      X       2498
 18    G12AK25.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            328     VBS      X       2498
 19    D12AK25.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            328     VBS      X       2498
 20    G12AK32.E025.N90       17 NOV 78     0 JAN 00   XNSDTAPE/           1981     VBS      X       2498
 21    D12AK32.E025.N90       17 NOV 78     0 JAN 00   XNSDTAPE/            640     VBS      X       2498
 22    N12AK38.E0.N90         17 NOV 78     0 JAN 00   XNSDTAPE/            756     VBS      X       2498
 23    N12AK38.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            917     VBS      X       2498
 24    N12AK40.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/           1283     VBS      X       2498
 25    G32AK41.E0125.N90      17 NOV 78     0 JAN 00   XNSDTAPE/            532     VBS      X       2498
 26    G12AK25.E0125.N60      19 NOV 78     0 JAN 00   XNSDTAPE/           2522     VBS      X       2498
 27   G12AK25.E0125.N120      19 NOV 78     0 JAN 00   XNSDTAPE/            909     VBS      X       2498
 28    N12AK32.E0.N90         19 NOV 78     0 JAN 00   XNSDTAPE/            528     VBS      X       2498
 29    N32AK38.E0125.N90      19 NOV 78     0 JAN 00   XNSDTAPE/           1565     VBS      X       2498
 30   G32AK41.E0125.N120      19 NOV 78     0 JAN 00   XNSDTAPE/            976     VBS      X       2498

END
```

Files 7 through 13 have sine wave initial conditions. They are characterized by the initial wave steepness or by the ratio of wave energy (E) to that of the most energetic, steady wave of the same wavelength (Emax).

Label	Name	Initial ak	E/Emax	Timesteps	Time Span
7	SINE60.MK1	$\pi/14$	0.34	0-132	0-19.13
8	SINE60.MK2	$\pi/7$	1.39	0-243	0-4.29
9	SINE60.MK3	$2\pi/7$	5.92	0-153	0-2.01
10	SINE60.MK4	0.3848	1.02	0-229	0-4.51
11	SINE60.MK5	0.3100	0.66	0-193	0-5.63
12	SINE60.MK6	0.4900	1.67	0-249	0-4.17
13	SINE60.MK7	0.5400	2.04	0-273	0-4.11

Files 14 through 30 contain the results of the stability experiments (Longuet-Higgins and Cokelet, 1979). The file names themselves reveal much about the files. The prefix N, G or D signifies theoretically neutral, growing or decaying modes. The numbers 12 or 32 indicate the mode number, n = 1/2 or n = 3/2. The next four characters give the wave steepness. For example, AK10 means AK = 0.10. Following the decimal point is the size of the perturbation parameter ϵ ; E0125 means $\epsilon = 0.0125$. The number of points along the profile is given after the second decimal point; N90 means N=90. For these files the following table is of use.

Label	Name	Timesteps	Time Span
14	N12AK10.E0125.N90	0-87	0-10.88
15	N32AK10.E0125.N90	0-68	0-8.50
16	N12AK20.E0125.N90	0-86	0-10.75
17	N32AK20.E0125.N90	0-76	0-9.50
18	G12AK25.E0125.N90	0-84	0-10.50
19	D12AK25.E0125.N90	0-~84	0-~10.50
20	G12AK32.E025.N90	0-513	0-25.03
21	D12AK32.E025.N90	0-165	0-11.63

Label	Name	Timesteps	Time Span
22	N12AK38.E0.N90	0-194	0-12.50
23	N12AK38.E0125.N90	0-237	0-12.41
24	N12AK40.E0125.N90	0-332	0-12.22
25	G32AK41.E0125.N90	0-137	0-1.46
26	G12AK25.E0125.N60	85-970	10.63-101
27	G12AK25.E0125.N120	870-1047	99.19-100.85
28	N12AK32.E0.N90	0-136	0-11.56
29	N32AK38.E0125.N90	0-405	0-17.00
30	G32AK41.E0125.N120	0-190	0-1.47

II.H. Sample JCL

The following is an example of RHEL IBM 360/195 job control language (JCL) to run INTVEL and produce a graphics tape to make plots on IOS's Calcomp plotter.

The job is to run at priority 8. Its name is XNSD, account number AN01, user SD. It will take about 30 seconds to run and will produce 5 k or less lines of output at 60 lines/page. Its banner heading will have the characters "COKELET-P]06" incorporated within it. Print output will go to REMOTE21, i.e., IOS.

The job uses 2 magnetic tapes M907 from which it reads the data and M800 to which it writes the plot instructions. The SETUP cards tell it to read only standard label tape M907 which has 6250 bpi tape density. It writes to a no label, 800 bpi tape.

The IBM utility IEBGENER is used to transfer the input tape file to disc where it can be more easily manipulated. The original file is called G12Ak25.E0125.N120, and it is number 27 on tape M907 which is at 6250 bpi density. The file is to be "shared" with other programs if need be. The record format is VBS, block size 2498, record length X and density 4 (corresponding to 6250 bpi). The output disc file is called &&TEMP, and it is to be passed along to a later job step. The file will reside on disc unit TEMP14 which is for larger files. This file will require about 400 tracks of disc. Its data control block information (DCB) is as on tape.

Next we run a FORTRAN job on the G compiler. The listing is printed with no map. The compile, link-edit and go step sizes are increased over their default values. The main source deck and subroutines follow next. In the link-edit step the Harwell and IOS graphics libraries are included in object form. FORTRAN unit 8 is the location of the temporary input data file. Unit 15 is the plotter tape. It is 800 bpi, and the plot file is a new one at position 1. The record format is VBS, block size 488 and density 2 (800 bpi). The tape name is M800.

```
/*PRIORITY 8
//XNSD JOB (AN01,SD,0-30,5,,,,,60),COKELET-P106
/*ROUTE  PRINT REMOTE21
/*SETUP M907,R,SL,DEN6250
/*SETUP M800,W,NL,DEN800
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=G12AK25.E0125.N120,LABEL=27,
// VOL=(,RETAIN,(SER=M907)),UNIT=DEN6250,
// DISP=SHR,DCB=(RECFM=VBS,BLKSIZE=2498,LRECL=X,DEN=4)
//SYSUT2 DD DSN=&&TEMP,DISP=(,PASS),UNIT=TEMP14,
// SPACE=(TRK,(400,10)),DCB=(RECFM=VBS,BLKSIZE=2498,LRECL=X)
// EXEC FGCLG,CPRINT=YES,PARM.C='NOMAP',REGION.C=160K,
// PARM.L='SIZE=(208K,24K),NOMAP,LET',REGION.L=210K,REGION.G=240K
//C.SYSIN DD *
      .
      .
      .
 <SOURCE DECK>
      .
      .
      .
//L.SYSLIB DD
//         DD DSN=SYS1.HARLIB,DISP=SHR
//         DD DSN=ULIB.IOS,DISP=SHR
//G.FT08F001 DD DSN=&&TEMP,DISP=SHR,LABEL=(,,,IN)
//G.FT15F001 DD UNIT=(DEN800,,DEFER),DISP=(NEW,KEEP),
// LABEL=(1,NL,,OUT),DCB=(RECFM=VBS,LRECL=84,BLKSIZE=488,DEN=2),
// VOL=SER=M800
```

III. Steady Waves

III.A. Introduction

This section deals with some computer programs at IOS which calculate the properties of steady waves in water of constant depth. This work appears in Cokelet, E.D. 1977 Steep gravity waves in water of arbitrary uniform depth. Phil. Trans. R. Soc. Lond. A 286, 183-230.

The subroutine STDYWV calculates the steady wave expansion. The program STDYWVEX calls STDYWV and prints out the expansion coefficients. No Padé approximant routines are included here, but a good reference is G. A. Baker 1975 Essentials of Padé Approximants, Academic Press, pp. 306.

The Padé-approximated results which appear in the appendix of Cokelet (1977) are held on a file at RHEL. They can be read by a program, TABLE, which prints out the results in tabular form.

```
DATE = 24/11/78      TIME =  1.33.57

FILE = SDMAILDR.STDYWVEX(DG)

C THIS PROGRAM CALCULATES THE STEADY WAVE EXPANSION AND WRITES
C OUT THE COEFFICIENTS.
C
      DOUBLE PRECISION ALPHA(102, 51), GAMMA(51), DELTA(51), D(102),
     1  SIGMA(102),R0
      NORDER = 100
      NROW = 102
      NCOL = 51
      R0 = 0.9D0
      CALL STDYWV(NORDER, NROW, NCOL, R0, ALPHA, GAMMA, DELTA, D, SIGMA)
C
C WRITE OUT THE COEFFICIENTS.
C
      I1E = NORDER + 1
      WRITE(6, 1001)
      DO 10 I1=2, I1E
      I = I1 - 1
      J1E = INT(FLOAT(NORDER - I)/2. + .001) + 1
      WRITE(6, 1002) I, (ALPHA(I1, J1), J1=1, J1E)
      WRITE(6, 1006)
 10   CONTINUE
      WRITE(6, 1006)
      DO 20 I1=1, I1E
      I = I1 - 1
      J1E = INT(FLOAT(NORDER - I)/2. + .001) + 1
      WRITE(6, 1003) I, (ALPHA(NROW - I, NCOL - J1 + 1), J1=1,J1E)
      WRITE(6, 1006)
 20   CONTINUE
      WRITE(6, 1006)
      WRITE(6, 1004) (GAMMA(J1), J1=1, NCOL)
      WRITE(6, 1006)
      WRITE(6, 1005) (DELTA(J1), J1=1, NCOL)
      WRITE(6, 1006)
      WRITE(6, 1007) (D(I1), I1=1, I1E)
      WRITE(6, 1006)
      WRITE(6, 1008) (SIGMA(I1), I1=1, I1E)
      STOP
 1001 FORMAT(1H1)
 1002 FORMAT( ' ALPHA(', I3, ',J)', 1P5D24.16, 30(/, 13X, 5D24.16))
 1003 FORMAT( '  BETA(', I3, ',J)', 1P5D24.16, 30(/, 13X, 5D24.16))
 1004 FORMAT(/, ' GAMMA(J)     ', 1P5D24.16, 30(/, 13X, 5D24.16))
 1005 FORMAT(/, ' DELTA(J)     ', 1P5D24.16, 30(/, 13X, 5D24.16))
 1006 FORMAT(1X)
 1007 FORMAT(/, ' D(I)         ', 1P5D24.16, 30(/, 13X, 5D24.16))
 1008 FORMAT(/, ' SIGMA(I)     ', 1P5D24.16, 30(/, 13X, 5D24.16))
      END
```

III.B. Perturbation Expansion

The steady wave perturbation expansion is done by the subroutine STDYWV. The description is as follows:

STDYWV (NORDER, NROW, NCOL, RO, ALPHA, GAMMA, DELTA, D, SIGMA)

calculates the coefficients in the steady water wave expansion of Cokelet, 1977. The expansion parameter is $\epsilon^2 = 1 - \frac{q^2_{crest}\, q^2_{trough}}{c^4}$

Inputs:

NORDER	order of expansion
NROW	=NORDER+2 if NORDER is even =NORDER+3 is NORDER is odd NROW is the number of rows in the array ALPHA
NCOL	=NROW/2. NCOLis the number of columns in array ALPHA
RO	=Exp (-nondimensional water depth)

Outputs:

ALPHA (NROW, NCOL)	array of Fourier expansion coefficients as defined in Cokelet's equations (3.1a) and (3.1b)
GAMMA (NCOL)	array of phase velocity coefficients in equation (3.1.c)
DELTA (NCOL)	array of Bernoulli constant coefficients in equation (3.1d)
D (NROW)	array of depth-dependent coefficients as defined in equation (2.12)

SIGMA (NROW)	array of depth-dependent coefficients as defined in equation (2.12)

Subroutines called: none

Note: To save array storage space the α_{ij} and β_{ij} arrays of the paper have been packed into the ALPHA array of the program. Also since FORTRAN does not allow zero array indices, the program indices have +1 added to those of the paper. The equivalences are as follows:

Paper	Program
α_{ij}	ALPHA (I + 1, J + 1)
β_{ij}	ALPHA (NROW-I, NCOL-J)
γ_i	GAMMA (I + 1)
Δ_i	DELTA (I + 1)
δ_i	D (I + 1)
σ_i	SIGMA (I + 1)

STDYWVEX is a simple program to call STDYWV and to write out the arrays of coefficients. In the listing which follows the expansion is carried out to order ϵ^{100} with $r_0 = 0.9$.

```
      SUBROUTINE STDYWV(NORDER, NROW, NCOL, R0, ALPHA, GAMMA, DELTA,
     1  D, SIGMA)
C
C   SUBROUTINE TO FIND THE COEFFICIENTS IN THE STEADY WATER WAVE
C EXPANSION OF
C     COKELET, E.D. 1977 STEEP GRAVITY WAVES IN WATER OF ARBITRARY
C            UNIFORM DEPTH.  PHIL. TRANS. R. SOC. LOND. A 286, 183-230.
C EXPANSION PARAMETER IS
C     EPSILON**2 = 1 - (QCREST**2)*(QTROUGH**2)/(C**4)
C
C  INPUTS:
C NORDER = ORDER OF EXPANSION
C NROW = NORDER + 2  IF NORDER IS EVEN.
C      = NORDER + 3  IF NORDER IS ODD.
C        NROW IS NUMBER OF ROWS IN ALPHA ARRAY.
C NCOL = NROW/2.  NCOL IS NUMBER OF COLUMNS IN ALPHA ARRAY.
C R0   = EXP(-NONDIMENSIONAL WATER DEPTH).
C
C  OUTPUTS:
C ALPHA(NROW, NCOL) = ARRAY OF FOURIER EXPANSION COEFFICIENTS ALPHA(I,J)
C                        AND BETA(I, J), AS DEFINED IN COKELET≥S
C                        EQUATIONS (3.1A) AND (3.1B).
C GAMMA(NCOL) = ARRAY OF PHASE VELOCITY COEFFICIENTS IN EQUATION (3.1C).
C DELTA(NCOL) = ARRAY OF BERNOULLI CONSTANT COEFFICIENTS IN EQUATION
C                   (3.1D).
C D(NROW)     = ARRAY OF DEPTH-DEPENDENT COEFFICIENTS AS DEFINED
C                  IN EQUATIONS (2.12).
C SIGMA(NROW) = ARRAY OF DEPTH-DEPENDENT COEFFICIENTS AS DEFINED
C                  IN EQUATIONS (2.12).
C
C TO SAVE ARRAY STORAGE SPACE THE ALPHA AND BETA ARRAYS OF THE PAPER
C HAVE BEEN PACKED INTO THE ALPHA ARRAY OF THE PROGRAM.  ALSO BECAUSE
C FORTRAN DOES NOT ALLOW ZERO ARRAY INDICES, THE PROGRAM INDICES HAVE
C 1 ADDED TO THOSE OF THE PAPER.  THE EQUIVALENCES ARE AS FOLLOWS:
C
C            PAPER                        PROGRAM
C
C         ALPHA(I, J)                   ALPHA(I+1, J+1)
C          BETA(I, J)                   ALPHA(NROW-I, NCOL-J)
C         GAMMA(I)                      GAMMA(I+1)
C (UPPER CASE)DELTA(I)                  DELTA(I+1)
C (LOWER CASE)DELTA(I)                      D(I+1)
C         SIGMA(I)                      SIGMA(I+1)
C
C PROGRAMMED BY E.D. COKELET.
C
      DOUBLE PRECISION A, B, C, D, A1, A2, A5, B1, B2, B3, B4, B5,
     1  C1, C3, C4, C5, D1, D3, D5, R0, S1, XJ, ROE, R02,
     2  ALPHA, DELTA, GAMMA, SIGMA
      INTEGER P, R, S
      DIMENSION ALPHA(NROW, NCOL), GAMMA(NCOL),
     1  DELTA(NCOL), D(NROW), SIGMA(NROW)
      N = NORDER
C
C...COMPUTE THE D(I) AND SIGMA(I).
C
      D(1) = 1.
      SIGMA(1) = 1.
      ROE = 1.
      R02 = R0**2
```

```
      DO 80 I=2,NROW
      ROE = ROE*RO2
      D(I) = 1. - ROE
      SIGMA(I) = 2. - D(I)
   80 CONTINUE
C
C...COMPUTE THE TERMS FROM THE ZERO, FIRST AND SECOND ORDER EQUATIONS.
C
      DO 90 K=1,NCOL
      ALPHA(1,K) = 0.
   90 CONTINUE
      ALPHA(NROW, NCOL) = 1.
      DELTA(1) = D(2)/SIGMA(2)
      ALPHA (2,1) = (SIGMA(2)*D(3)/2. - D(2)*SIGMA(3))/(2.*D(2)*
     1  SIGMA(3)*(SIGMA(2) - D(2) - SIGMA(2)**2) - SIGMA(2)**3*D(3))
      ALPHA(3,1) =  (1. + 2.*SIGMA(2)**2*ALPHA(2,1))/
     1  (4.*SIGMA(3))
      ALPHA(NROW-2, NCOL) = ALPHA(3,1)*SIGMA(3) + (SIGMA(2) - D(2))
     1  *ALPHA(2,1)/2.
      ALPHA(NROW, NCOL-1) = SIGMA(3)*ALPHA(2,1)
      ALPHA(2,1) = DSQRT(ALPHA(2,1))
      ALPHA(NROW-1, NCOL) = SIGMA(2)*ALPHA(2,1)
C
C...COMPUTE THE HIGHER ORDER TERMS
C
      MOE = NORDER + 1
      DO 162 MO=3, MOE
      MX = INT(FLOAT(MO)/2. + .01)
      MY = INT(FLOAT(MO + 1)/2. + .01)
      KE = MX - 1
      IF (MX - MY .EQ. 0) GO TO 510
C
C...CALCULATE THE DELTA(M) IF MO IS ODD.
C
      KE = MY - 1
      M = MY
      DELTA(M)=D(2)*ALPHA(2,1)*ALPHA(NROW, NCOL+1-M) + D(M+1)*ALPHA(M+1,
     1  1)*ALPHA(NROW-M+1, NCOL)/FLOAT(M) + (D(M)*ALPHA(NROW-M,NCOL)
     2  /FLOAT(M-1) - DELTA(1)*SIGMA(2*M)*ALPHA(M+1,1))*ALPHA(M,1)
      LE = M-2
      IF (LE .LE. 0) GO TO 141
      DO 140 L=1,LE
      DELTA(M) = DELTA(M) + D(2)*ALPHA(2,M-L)*ALPHA(NROW,NCOL-L)
     1 - DELTA(L+1)*ALPHA(NROW-1, NCOL+1-M+L)
      S1 = 0.
      ISE = M-L
      DO 120 IS=1,ISE
      S = IS - 1
      S1 = S1 +(D(L+2)*ALPHA(NROW-L,NCOL-S)/FLOAT(L+1) - DELTA(1)*
     1  SIGMA(2*L+2)*ALPHA(L+1,S+1))*ALPHA(L+2,M-L-S)
  120 CONTINUE
      DELTA(M) = DELTA(M) + S1
      S1 = 0.
      ISE = L+1
      DO 130 IS=1, ISE
      S = IS - 1
      S1 = S1 + ALPHA(M-L, L-S+1)*ALPHA(NROW-M+L,NCOL-S)
  130 CONTINUE
      DELTA(M) = DELTA(M) + D(M-L)*S1/FLOAT(M-1-L)
  140 CONTINUE
```

```
  141 CONTINUE
      DELTA(M) = DELTA(M)/ALPHA(NROW-1,NCOL)
      IF (2*M-2 .EQ. NORDER) GO TO 164
C
C... COMPUTE THE ALPHA(J, K) AND ALPHA(NROW+1-J, NCOL+1-K).
C
  510  CONTINUE
      M = MO + 1
  165 DO 295 K=1,KE
      J = M+2-2*K
      IF (J .EQ. 3) GO TO 295
      A = 0.0
      IF (K .LE. 1) GO TO 181
      LE = K-1
      DO 180 L=1,LE
      S1 = 0.0
      IRE = K-L
      DO 170 IR=1,IRE
      R = IR-1
      S1 = S1 + ALPHA(L+1, R+1)*ALPHA(L+J, K-L-R)
  170 CONTINUE
      A = A+ SIGMA(2*L+J)*S1
  180 CONTINUE
  181 CONTINUE
  182 CONTINUE
      LE = J-2
      DO 200 L=1,LE
      S1 = 0.0
      DO 190 IR=1,K
      R = IR-1
      S1 = S1 + ALPHA(L+1,R+1)*ALPHA(J-L,K-R)
  190 CONTINUE
      A = A + (SIGMA(L+1) - D(J-L))*S1/2.
  200 CONTINUE
  202 CONTINUE
      B = 0.0
      LE = J-2
      DO 220 L=1,LE
      S1 = 0.0
      DO 210 IS=1,K
      S = IS-1
      S1 = S1 + ALPHA(L+1, K-S)*ALPHA(NROW+1-J+L, NCOL-S)
  210 CONTINUE
      B = B + D(L+1)*S1/FLOAT(L)
  220 CONTINUE
   222 CONTINUE
      IF (K .LE. 1) GO TO 282
      S1 = 0.0
      ISE = K-1
      DO 230 S=1,ISE
      S1 = S1 + ALPHA(J, K-S )*ALPHA(NROW, NCOL-S)
  230 CONTINUE
      B = B + D(J)*S1/FLOAT(J-1)
  240 CONTINUE
      LE = K-1
      DO 260 L=1,LE
      S1 = 0.0
      ISE = K-L
      DO 250 IS=1,ISE
      S = IS-1
```

```
      S1 = S1 + ALPHA(L+J, K-L-S)*ALPHA(NROW-L, NCOL-S)
  250 CONTINUE
      B = B + D(L+J)*S1/FLOAT(L+J-1)
  260 CONTINUE
      ILE = K-1
      DO 280 IL=1,ILE
      L = IL-1
      S1 = 0.0
      ISE = L+1
      DO 270 IS=1,ISE
      S = IS-1
      S1 = S1 + ALPHA(K-L, L-S+1)*ALPHA(NROW+2-J-K+L, NCOL-S)
  270 CONTINUE
      B = B + D(K-L)*S1/FLOAT(K-L-1)
  280 CONTINUE
  282 CONTINUE
      C = 0.0
      IF (K .LE. 1) GO TO 292
      LE = K-1
      DO 290 L=1,LE
      C = C + DELTA(L+1)*ALPHA(NROW+1-J, NCOL+1-K+L)
  290 CONTINUE
  292 CONTINUE
      ALPHA(J,K)=(C-B+DELTA(1)*A)/(D(J)/FLOAT(J-1)-DELTA(1)*SIGMA(J))
  294 ALPHA(NROW+1-J, NCOL+1-K) = ALPHA(J, K)*SIGMA(J) + A
  295 CONTINUE
C
C...CALCULATE ALPHA(3,MO/2), ALPHA(NROW-2,NCOL+1-MO/2), ALPHA(2,MO/2)
C...  AND ALPHA(NROW-1,NCOL+1-MO/2) IF MO IS EVEN.
C
      IF (MX .NE. MY) GO TO 162
  142 CONTINUE
      M = INT(FLOAT(MO)/2. + .01)
      A1 = -SIGMA(2)
      A2 = 1.
      A5 = 0.
      LE = M-1
      DO 410 L=1, LE
      IRE = M-L
      S1 = 0.
      DO 400 IR=1, IRE
      R = IR - 1
      S1 = S1 + ALPHA(L+1, R+1)*ALPHA(L+2, M-L-R)
  400 CONTINUE
      A5 = A5 + SIGMA(2*L+2)*S1
  410 CONTINUE
      B1 = D(2)*ALPHA(NROW-1, NCOL)
      B2 = D(2)*ALPHA(2,1)
      B3 = D(3)*ALPHA(NROW, NCOL)/2.
      B4 = -DELTA(1)
      B5 = -D(3)*ALPHA(3,1)*ALPHA(NROW, NCOL+1-M)/2. - D(M+2)*
     1  ALPHA(M+2,1)*ALPHA(NROW-M+1, NCOL)/FLOAT(M+1) - D(M)*
     2  ALPHA(M,1)*ALPHA(NROW-M-1, NCOL)/FLOAT(M-1) + DELTA(M)*
     3  ALPHA(NROW-2, NCOL)
      A = 0.
      LE = M-2
      IF (LE .LE. 0) GO TO 442
      DO 440 L=1,LE
      A = A + D(2)*ALPHA(2,M-L)*ALPHA(NROW-1,NCOL-L) + D(3)*
     1  ALPHA(3,M-L)*ALPHA(NROW, NCOL-L)/2. - DELTA(L+1)*
```

```
     2 ALPHA(NROW-2, NCOL+1-M+L)
      S1 = 0.
      ISE = M-L
      DO 420 IS=1,ISE
      S = IS - 1
      S1 = S1 + ALPHA(L+3, M-L-S)*ALPHA(NROW-L,NCOL-S)
  420 CONTINUE
      A = A + D(L+3)*S1/FLOAT(L+2)
      S1 = 0.
      ISE = L+1
      DO 430 IS=1,ISE
      S = IS - 1
      S1 = S1 + ALPHA(M-L, L-S+1)*ALPHA(NROW-M-1+L,NCOL-S)
  430 CONTINUE
      A = A + D(M-L)*S1/FLOAT(M-1-L)
  440 CONTINUE
  442 CONTINUE
      B5 = B5 - A
      C1 = (D(2) - SIGMA(2))*ALPHA(2,1)
      C3 = -SIGMA(3)
      C4 = 1.
      C5 = SIGMA(2*M + 1)*ALPHA(M,1)*ALPHA(M+2,1)
      LE = M-2
      IF (LE .LE. 0) GO TO 462
      DO 460 L=1,LE
      C5 = C5 + (SIGMA(2) - D(2))*ALPHA(2,L+1)*ALPHA(2,M-L)/2.
      S1 = 0.
      IRE = M-L
      DO 450 IR=1, IRE
      R = IR - 1
      S1 = S1 + ALPHA(L+1, R+1)*ALPHA(L+3, M-L-R)
  450 CONTINUE
      C5 = C5 + SIGMA(2*L+3)*S1
  460 CONTINUE
  462 CONTINUE
      D1 = -2.*SIGMA(2)**2*ALPHA(2,1)
      D3 = 2.*SIGMA(3)
      S1 = 1.
      DO 470 J=1,M
      XJ = FLOAT(J-1)
      S1 = S1*(1. + 2.*XJ)/(2.*(FLOAT(M)-XJ))
  470 CONTINUE
      D5 = S1 - SIGMA(3)*SIGMA(2*M-1)*ALPHA(3,1)*ALPHA(2*M-1,1)
     1 - 2.*SIGMA(2*M+1)*ALPHA(2*M+1,1) + 2.*SIGMA(2)*SIGMA(2*M)*
     1 ALPHA(2,1)*ALPHA(2*M,1)
      LE = M-2
      IF (LE .LE. 0) GO TO 502
      DO 500 L=1,LE
      A = 0.
      IRE = M-L
      DO 490 IR=1,IRE
      R = IR - 1
      S1 = 0.
      JE = M-L-R
      DO 480 J1=1,JE
      J = J1 - 1
      IN = (M-L-R-J)*2
      S1 = S1 + SIGMA(IN+1)*SIGMA(2*L+1)*ALPHA(IN+1, R+1)*ALPHA(2*L+1,
     1 J+1) - SIGMA(IN)*SIGMA(2*L+2)*ALPHA(IN,R+1)*ALPHA(2*L+2,J+1)
  480 CONTINUE
```

```
      IN = 2*(M-L-R)
      A = A + SIGMA(IN)*SIGMA(2)*ALPHA(IN,L+1)*ALPHA(2,R+1) - S1
  490 CONTINUE
      IN = 2*(M-L)
      D5 = D5 + SIGMA(IN)*SIGMA(2)*ALPHA(IN,1)*ALPHA(2,L+1) + A
     1 - 2.*SIGMA(IN+1)*ALPHA(IN+1,L+1)
  500 CONTINUE
  502 CONTINUE
      ALPHA(2,M) = (C4*(B5 - A5*B2/A2 - B3*D5/D3) - B4*(C5 - C3*D5/D3
     1 ))/(C4*(B1 - A1*B2/A2 - B3*D1/D3) - B4*(C1 - C3*D1/D3))
      ALPHA(NROW-1, NCOL+1-M) = (A5 - A1*ALPHA(2,M))/A2
      IF (2*M-1 .EQ. NORDER) GO TO 164
      ALPHA(3,M) = (D5 - D1*ALPHA(2,M))/D3
      ALPHA(NROW-2, NCOL+1-M) = (C5 - C3*D5/D3 - (C1 - C3*D1/D3)*
     1 ALPHA(2,M))/C4
C
C...CALCULATE ALPHA(NROW, NCOL-M0/2).
C
      M = MX
      A = 0.0
      DO 530 L=1,M
      IRE = M - L + 1
      S1 = 0.0
      DO 520 IR=1,IRE
      R = IR - 1
      S1 = S1 + ALPHA(L+1, R+1)*ALPHA(L+1, M-L-R+1)
  520 CONTINUE
      A = A + SIGMA(2*L+1)*S1
  530 CONTINUE
      ALPHA(NROW, NCOL-M) = A
C
  162 CONTINUE
  164 CONTINUE
C
C...COMPUTE THE GAMMA(L).
C
      LE = INT(FLOAT(NORDER)/2. + .01) + 1
      DO 324 L=1,LE
      GAMMA(L) = 0.0
      DO 300 KI = 1,L
      K = KI-1
      GAMMA(L) = GAMMA(L) + DELTA(L-K)*ALPHA(NROW, NCOL-K)
  300 CONTINUE
      IF (L .LT. 2) GO TO 322
      IRE = L-1
      DO 320 IR=1,IRE
      R = IR-1
      S1 = 0.0
      IME = R+1
      DO 310 IM=1,IME
      M = IM-1
      S1 = S1 + ALPHA(L-R, R-M+1)*ALPHA(NROW+1-L+R, NCOL-M)
  310 CONTINUE
      GAMMA(L) = GAMMA(L) - 2.*D(L-R)*S1/FLOAT(L-R-1)
  320 CONTINUE
  322 CONTINUE
  324 CONTINUE
      RETURN
      END
```

III.C. Padé-approximated Results

The Padé-approximated results which appear in the appendix of Cokelet (1977) are held in card image form on ELECTRIC file STEADYKD. The first 54 cards are for $r_0 = e^{-d} = 0$. Then the cards come in 9 blocks of 72 cards each corresponding to $r_0 = e^{-d} = 0.1, 0.2, \ldots, 0.9.$

For all values of e^{-d} except $e^{-d} = 0$, each block of cards contains 12 sections with 6 cards in each. These sections correspond to values of wave amplitude (a), phase speed squared (c^2), momentum (I), kinetic energy (T), potential energy (V), mean surface elevation ($\bar{\eta}$), Bernoulli constant (K), radiation stress (Sxx), energy flux (F), mean square bottom velocity ($\overline{u_b^2}$), total head (R) and momentum flux (S).

For $e^{-d} = 0$ only the first 9 sections occur since $\overline{u_b^2} = 0$, and R and S are undefined for infinite depth.

Each section of 6 cards is headed by a blank card (or one containing non-essential alphanumeric data) followed by 4 cards with 6 values on each and 1 card with 4 values. The format of each card is

FORMAT (6 (D11.5, 1X)).

These 28 data points are for $\epsilon^2 = 0.1, 0.2, \ldots, 0.8, 0.81, 0.82, \ldots, 1.$

The ELECTRIC file TABLE is a source deck which can read this data and print the tables in the appendix of Cokelet (1977).

FILE = STEADYKD

```
0001:                                                                         EDC00010
0002: 1.30622D-01 1.86957D-01 2.31791D-01 2.70970D-01 3.06684D-01 3.39924D-01 EDC00020
0003: 3.71029D-01 3.99671D-01 4.02358D-01 4.05006D-01 4.07612D-01 4.10172D-01 EDC00030
0004: 4.12684D-01 4.15145D-01 4.17550D-01 4.19897D-01 4.22181D-01 4.24397D-01 EDC00040
0005: 4.26542D-01 4.28611D-01 4.30602D-01 4.32512D-01 4.34342D-01 4.36098D-01 EDC00050
0006: 4.37795D-01 4.39467D-01 4.41190D-01 4.4313 D-01                         EDC00060
0007:                                                                         EDC00070
0008: 1.01721D 00 1.03557D 00 1.05520D 00 1.07619D 00 1.09858D 00 1.12229D 00 EDC00080
0009: 1.14688D 00 1.17093D 00 1.17319D 00 1.17540D 00 1.17755D 00 1.17963D 00 EDC00090
0010: 1.18164D 00 1.18357D 00 1.18540D 00 1.18711D 00 1.18870D 00 1.19014D 00 EDC00100
0011: 1.19142D 00 1.19251D 00 1.19339D 00 1.19404D 00 1.19443D 00 1.19454D 00 EDC00110
0012: 1.19436D 00 1.19391D 00 1.19329D 00 1.1928 D 00                         EDC00120
0013:                                                                         EDC00130
0014: 8.45446D-03 1.71367D-02 2.60090D-02 3.50007D-02 4.39839D-02 5.27280D-02 EDC00140
0015: 6.08108D-02 6.74309D-02 6.79621D-02 6.84620D-02 6.89285D-02 6.93591D-02 EDC00150
0016: 6.97512D-02 7.01022D-02 7.04094D-02 7.06698D-02 7.08804D-02 7.10386D-02 EDC00160
0017: 7.11415D-02 7.11868D-02 7.11729D-02 7.10996D-02 7.09686D-02 7.07859D-02 EDC00170
0018: 7.0563 D-02 7.0333 D-02 7.016  D-02 7.01   D-02                         EDC00180
0019:                                                                         EDC00190
0020: 4.26345D-03 8.71942D-03 1.33586D-02 1.81548D-02 2.30505D-02 2.79296D-02 EDC00200
0021: 3.25619D-02 3.64834D-02 3.68062D-02 3.71118D-02 3.73988D-02 3.76658D-02 EDC00210
0022: 3.79110D-02 3.81328D-02 3.83294D-02 3.84989D-02 3.86395D-02 3.87492D-02 EDC00220
0023: 3.88262D-02 3.88687D-02 3.88755D-02 3.88459D-02 3.87806D-02 3.86824D-02 EDC00230
0024: 3.8557 D-02 3.8421 D-02 3.8303 D-02 3.827  D-02                         EDC00240
0025:                                                                         EDC00250
0026: 4.22707D-03 8.56686D-03 1.29987D-02 1.74844D-02 2.19555D-02 2.62900D-02 EDC00260
0027: 3.02660D-02 3.34698D-02 3.37220D-02 3.39582D-02 3.41771D-02 3.43776D-02 EDC00270
0028: 3.45585D-02 3.47184D-02 3.48561D-02 3.49701D-02 3.50592D-02 3.51219D-02 EDC00280
0029: 3.51573D-02 3.51642D-02 3.51423D-02 3.50916D-02 3.50137D-02 3.49119D-02 EDC00290
0030: 3.47935D-02 3.4673 D-02 3.457  D-02 3.457  D-02                         EDC00300
0031:                                                                         EDC00310
0032: 8.38264D-03 1.68398D-02 2.53196D-02 3.37390D-02 4.19640D-02 4.97724D-02 EDC00320
0033: 5.67834D-02 6.23151D-02 6.27455D-02 6.31477D-02 6.35198D-02 6.38601D-02 EDC00330
0034: 6.41665D-02 6.44370D-02 6.46694D-02 6.48617D-02 6.50116D-02 6.51172D-02 EDC00340
0035: 6.51765D-02 6.51882D-02 6.51514D-02 6.50665D-02 6.49358D-02 6.4765 D-02 EDC00350
0036: 6.4566 D-02 6.4363 D-02 6.4201 D-02 6.417  D-02                         EDC00360
0037:                                                                         EDC00370
0038: 1.03397D 00 1.06925D 00 1.10584D 00 1.14367D 00 1.18251D 00 1.22184D 00 EDC00380
0039: 1.26045D 00 1.29556D 00 1.29868D 00 1.30169D 00 1.30459D 00 1.30735D 00 EDC00390
0040: 1.30998D 00 1.31244D 00 1.31473D 00 1.31683D 00 1.31872D 00 1.32037D 00 EDC00400
0041: 1.32177D 00 1.32288D 00 1.32369D 00 1.32417D 00 1.32430D 00 1.32407D 00 EDC00410
0042: 1.32349D 00 1.32264D 00 1.32169D 00 1.3212 D 00                         EDC00420
0043:                                                                         EDC00430
0044: 4.37259D-03 9.17713D-03 1.44384D-02 2.01659D-02 2.63353D-02 3.28487D-02 EDC00440
0045: 3.94496D-02 4.55242D-02 4.60586D-02 4.65726D-02 4.70640D-02 4.75301D-02 EDC00450
0046: 4.79684D-02 4.83759D-02 4.87494D-02 4.90855D-02 4.93806D-02 4.96310D-02 EDC00460
0047: 4.98328D-02 4.99820D-02 5.00752D-02 5.01089D-02 5.00818D-02 4.99941D-02 EDC00470
0048: 4.9852 D-02 4.9671 D-02 4.9491 D-02 4.941  D-02                         EDC00480
0049:                                                                         EDC00490
0050: 4.37336D-03 9.18368D-03 1.44618D-02 2.02246D-02 2.64552D-02 3.30623D-02 EDC00500
0051: 3.97888D-02 4.60006D-02 4.65473D-02 4.70731D-02 4.75754D-02 4.80517D-02 EDC00510
0052: 4.84991D-02 4.89145D-02 4.92947D-02 4.96360D-02 4.99348D-02 5.01871D-02 EDC00520
0053: 5.03888D-02 5.05361D-02 5.06250D-02 5.06527D-02 5.06172D-02 5.05199D-02 EDC00530
0054: 5.03671D-02 5.01762D-02 4.99862D-02 4.9903 D-02                         EDC00540
0055:                                                                         EDC00550
0056: 1.27189D-01 1.82094D-01 2.25827D-01 2.64080D-01 2.98985D-01 3.31510D-01 EDC00560
0057: 3.61984D-01 3.90091D-01 3.92730D-01 3.95332D-01 3.97893D-01 4.00410D-01 EDC00570
0058: 4.02880D-01 4.05300D-01 4.07666D-01 4.09975D-01 4.12223D-01 4.14405D-01 EDC00580
0059: 4.16518D-01 4.18557D-01 4.20519D-01 4.22403D-01 4.24208D-01 4.25941D-01 EDC00590
0060: 4.27616D-01 4.2927 D-01 4.3097 D-01 4.329  D-01                         EDC00600
```

```
0061:                                                                           EDC00610
0062: 9.97193D-01 1.01533D 00 1.03472D 00 1.05546D 00 1.07759D 00 1.10103D 00 EDC00620
0063: 1.12534D 00 1.14912D 00 1.15135D 00 1.15354D 00 1.15566D 00 1.15772D 00 EDC00630
0064: 1.15971D 00 1.16162D 00 1.16342D 00 1.16512D 00 1.16668D 00 1.16811D 00 EDC00640
0065: 1.16937D 00 1.17045D 00 1.17132D 00 1.17196D 00 1.17234D 00 1.17245D 00 EDC00650
0066: 1.17227D 00 1.17182D 00 1.1712 D 00 1.1708 D 00                         EDC00660
0067:                                                                           EDC00670
0068: 8.09195D-03 1.64013D-02 2.48923D-02 3.34974D-02 4.20945D-02 5.04632D-02 EDC00680
0069: 5.81986D-02 6.45322D-02 6.50400D-02 6.55179D-02 6.59637D-02 6.63750D-02 EDC00690
0070: 6.67495D-02 6.70845D-02 6.73774D-02 6.76254D-02 6.78257D-02 6.79756D-02 EDC00700
0071: 6.80722D-02 6.81114D-02 6.80991D-02 6.80270D-02 6.78997D-02 6.77229D-02 EDC00710
0072: 6.7508 D-02 6.7286 D-02 6.710  D-02 6.706  D-02                         EDC00720
0073:                                                                           EDC00730
0074: 4.04029D-03 8.26329D-03 1.26604D-02 1.72068D-02 2.18485D-02 2.64755D-02 EDC00740
0075: 3.08691D-02 3.45883D-02 3.48943D-02 3.51840D-02 3.54561D-02 3.57090D-02 EDC00750
0076: 3.59412D-02 3.61513D-02 3.63373D-02 3.64976D-02 3.66304D-02 3.67337D-02 EDC00760
0077: 3.68059D-02 3.68454D-02 3.68510D-02 3.68220D-02 3.67591D-02 3.66650D-02 EDC00770
0078: 3.65463D-02 3.6418 D-02 3.631  D-02 3.630  D-02                         EDC00780
0079:                                                                           EDC00790
0080: 4.00620D-03 8.12016D-03 1.23224D-02 1.65766D-02 2.08180D-02 2.49305D-02 EDC00800
0081: 2.87032D-02 3.17417D-02 3.19808D-02 3.22045D-02 3.24118D-02 3.26016D-02 EDC00810
0082: 3.27727D-02 3.29239D-02 3.30538D-02 3.31613D-02 3.32450D-02 3.33037D-02 EDC00820
0083: 3.33363D-02 3.33418D-02 3.33199D-02 3.32708D-02 3.31957D-02 3.30981D-02 EDC00830
0084: 3.29850D-02 3.28702D-02 3.2783 D-02 3.280  D-02                         EDC00840
0085:                                                                           EDC00850
0086: 8.10334D-03 1.62770D-02 2.44710D-02 3.26054D-02 4.05508D-02 4.80924D-02 EDC00860
0087: 5.48620D-02 6.01996D-02 6.06145D-02 6.10021D-02 6.13606D-02 6.16882D-02 EDC00870
0088: 6.19830D-02 6.22431D-02 6.24663D-02 6.26506D-02 6.27940D-02 6.28943D-02 EDC00880
0089: 6.29500D-02 6.29595D-02 6.29222D-02 6.28384D-02 6.27104D-02 6.25442D-02 EDC00890
0090: 6.2351 D-02 6.2153 D-02 6.200  D-02 6.200  D-02                         EDC00900
0091:                                                                           EDC00910
0092: 1.01371D 00 1.04850D 00 1.08457D 00 1.12185D 00 1.16012D 00 1.19886D 00 EDC00920
0093: 1.23689D 00 1.27146D 00 1.27453D 00 1.27749D 00 1.28034D 00 1.28306D 00 EDC00930
0094: 1.28565D 00 1.28807D 00 1.29032D 00 1.29239D 00 1.29424D 00 1.29586D 00 EDC00940
0095: 1.29723D 00 1.29832D 00 1.29912D 00 1.29958D 00 1.29970D 00 1.29947D 00 EDC00950
0096: 1.29890D 00 1.29805D 00 1.29713D 00 1.29670D 00                         EDC00960
0097:                                                                           EDC00970
0098: 4.86378D-03 1.01187D-02 1.57814D-02 2.18515D-02 2.82923D-02 3.49914D-02 EDC00980
0099: 4.16754D-02 4.77157D-02 4.82400D-02 4.87428D-02 4.92217D-02 4.96744D-02 EDC00990
0100: 5.00981D-02 5.04901D-02 5.08473D-02 5.11663D-02 5.14438D-02 5.16761D-02 EDC01000
0101: 5.18595D-02 5.19906D-02 5.20659D-02 5.20829D-02 5.20403D-02 5.1939 D-02 EDC01010
0102: 5.1789 D-02 5.160  D-02 5.143  D-02 5.14   D-02                         EDC01020
0103:                                                                           EDC01030
0104: 4.46407D-03 9.33834D-03 1.46488D-02 2.04069D-02 2.65898D-02 3.31007D-02 EDC01040
0105: 3.96803D-02 4.57022D-02 4.62285D-02 4.67340D-02 4.72160D-02 4.76721D-02 EDC01050
0106: 4.80996D-02 4.84955D-02 4.88567D-02 4.91797D-02 4.94609D-02 4.96967D-02 EDC01060
0107: 4.98832D-02 5.00166D-02 5.00936D-02 5.01114D-02 5.00685D-02 4.9967 D-02 EDC01070
0108: 4.9812 D-02 4.9623 D-02 4.945  D-02 4.936  D-02                         EDC01080
0109:                                                                           EDC01090
0110: 3.12111D-04 6.14972D-04 9.05476D-04 1.17938D-03 1.43073D-03 1.65101D-03 EDC01100
0111: 1.82761D-03 1.94162D-03 1.94861D-03 1.95467D-03 1.95979D-03 1.96392D-03 EDC01110
0112: 1.96704D-03 1.96913D-03 1.97015D-03 1.97010D-03 1.96896D-03 1.96674D-03 EDC01120
0113: 1.96344D-03 1.95912D-03 1.95384D-03 1.94773D-03 1.94098D-03 1.93390D-03 EDC01130
0114: 1.92699D-03 1.9211 D-03 1.9176 D-03 1.919  D-03                         EDC01140
0115:                                                                           EDC01150
0116: 2.80944D 00 2.82683D 00 2.84487D 00 2.86351D 00 2.88264D 00 2.90202D 00 EDC01160
0117: 2.92103D 00 2.93832D 00 2.93985D 00 2.94133D 00 2.94276D 00 2.94412D 00 EDC01170
0118: 2.94541D 00 2.94662D 00 2.94775D 00 2.94878D 00 2.94971D 00 2.95052D 00 EDC01180
0119: 2.95120D 00 2.95175D 00 2.95214D 00 2.95238D 00 2.95244D 00 2.95232D 00 EDC01190
0120: 2.95203D 00 2.95161D 00 2.95115D 00 2.9509 D 00                         EDC01200
```

```
0121:                                                                          EDC01210
0122: 4.96254D 00 5.02004D 00 5.08059D 00 5.14428D 00 5.21098D 00 5.28009D 00 EDC01220
0123: 5.34990D 00 5.41587D 00 5.42190D 00 5.42776D 00 5.43344D 00 5.43891D 00 EDC01230
0124: 5.44414D 00 5.44911D 00 5.45378D 00 5.45812D 00 5.46208D 00 5.46562D 00 EDC01240
0125: 5.46870D 00 5.47126D 00 5.47324D 00 5.47459D 00 5.47525D 00 5.47519D 00 EDC01250
0126: 5.47440D 00 5.47298D 00 5.47123D 00 5.47016D 00                         EDC01260
0127:                                                                          EDC01270
0128: 1.17156D-01 1.67926D-01 2.08508D-01 2.44136D-01 2.76769D-01 3.07303D-01 EDC01280
0129: 3.36041D-01 3.62685D-01 3.65196D-01 3.67671D-01 3.70110D-01 3.72509D-01 EDC01290
0130: 3.74864D-01 3.77174D-01 3.79434D-01 3.81642D-01 3.83792D-01 3.85882D-01 EDC01300
0131: 3.87906D-01 3.89863D-01 3.91747D-01 3.93558D-01 3.95295D-01 3.96964D-01 EDC01310
0132: 3.98578D-01 4.0017 D-01 4.0181 D-01 4.037  D-01                         EDC01320
0133:                                                                          EDC01330
0134: 9.39572D-01 9.57179D-01 9.76003D-01 9.96136D-01 1.01763D 00 1.04039D 00 EDC01340
0135: 1.06401D 00 1.08713D 00 1.08929D 00 1.09141D 00 1.09348D 00 1.09584D 00 EDC01350
0136: 1.09741D 00 1.09926D 00 1.10101D 00 1.10265D 00 1.10418D 00 1.10556D 00 EDC01360
0137: 1.10678D 00 1.10782D 00 1.10866D 00 1.10927D 00 1.10963D 00 1.10972D 00 EDC01370
0138: 1.10954D 00 1.10910D 00 1.1085 D 00 1.1080 D 00                         EDC01380
0139:                                                                          EDC01390
0140: 7.06286D-03 1.43239D-02 2.17525D-02 2.92906D-02 3.68315D-02 4.41818D-02 EDC01400
0141: 5.09836D-02 5.65549D-02 5.70012D-02 5.74211D-02 5.78125D-02 5.81734D-02 EDC01410
0142: 5.85017D-02 5.87950D-02 5.90510D-02 5.92671D-02 5.94410D-02 5.95701D-02 EDC01420
0143: 5.96521D-02 5.96853D-02 5.96681D-02 5.96003D-02 5.94838D-02 5.9323 D-02 EDC01430
0144: 5.9130 D-02 5.8925 D-02 5.876  D-02 5.872  D-02                         EDC01440
0145:                                                                          EDC01450
0146: 3.42307D-03 7.00692D-03 1.07450D-02 1.46170D-02 1.85774D-02 2.25326D-02 EDC01460
0147: 2.62950D-02 2.94836D-02 2.97459D-02 2.99941D-02 3.02272D-02 3.04437D-02 EDC01470
0148: 3.06425D-02 3.08220D-02 3.09808D-02 3.11174D-02 3.12302D-02 3.13176D-02 EDC01480
0149: 3.13781D-02 3.14102D-02 3.14131D-02 3.13861D-02 3.13300D-02 3.12471D-02 EDC01490
0150: 3.1144 D-02 3.103  D-02 3.09   D-02 3.08   D-02                         EDC01500
0151:                                                                          EDC01510
0152: 3.39470D-03 6.88749D-03 1.04621D-02 1.40880D-02 1.77097D-02 2.12275D-02 EDC01520
0153: 2.44591D-02 2.70626D-02 2.72671D-02 2.74584D-02 2.76355D-02 2.77976D-02 EDC01530
0154: 2.79434D-02 2.80720D-02 2.81823D-02 2.82732D-02 2.83434D-02 2.83920D-02 EDC01540
0155: 2.84181D-02 2.84207D-02 2.83997D-02 2.83551D-02 2.82883D-02 2.8202 D-02 EDC01550
0156: 2.8102 D-02 2.800  D-02 2.79   D-02 2.79   D-02                         EDC01560
0157:                                                                          EDC01570
0158: 7.28645D-03 1.46408D-02 2.20183D-02 2.93473D-02 3.65111D-02 4.33156D-02 EDC01580
0159: 4.94261D-02 5.42413D-02 5.46149D-02 5.49638D-02 5.52862D-02 5.55804D-02 EDC01590
0160: 5.58449D-02 5.60777D-02 5.62770D-02 5.64409D-02 5.65675D-02 5.66550D-02 EDC01600
0161: 5.67018D-02 5.67066D-02 5.66688D-02 5.65888D-02 5.64689D-02 5.6314 D-02 EDC01610
0162: 5.6136 D-02 5.595  D-02 5.581  D-02 5.57   D-02                         EDC01620
0163:                                                                          EDC01630
0164: 9.55202D-01 9.88540D-01 1.02310D 00 1.05880D 00 1.09547D 00 1.13258D 00 EDC01640
0165: 1.16900D 00 1.20212D 00 1.20506D 00 1.20790D 00 1.21062D 00 1.21323D 00 EDC01650
0166: 1.21569D 00 1.21801D 00 1.22015D 00 1.22214D 00 1.22390D 00 1.22545D 00 EDC01660
0167: 1.22675D 00 1.22779D 00 1.22853D 00 1.22896D 00 1.22906D 00 1.22882D 00 EDC01670
0168: 1.22826D 00 1.22744D 00 1.22656D 00 1.2262 D 00                         EDC01680
0169:                                                                          EDC01690
0170: 5.21690D-03 1.07425D-02 1.65818D-02 2.27214D-02 2.91109D-02 3.56251D-02 EDC01700
0171: 4.19851D-02 4.75794D-02 4.80548D-02 4.85083D-02 4.89378D-02 4.93412D-02 EDC01710
0172: 4.97150D-02 5.00597D-02 5.03694D-02 5.06423D-02 5.08754D-02 5.10654D-02 EDC01720
0173: 5.12094D-02 5.13042D-02 5.13474D-02 5.13370D-02 5.12730D-02 5.11579D-02 EDC01730
0174: 5.1000 D-02 5.0820 D-02 5.066  D-02 5.06   D-02                         EDC01740
0175:                                                                          EDC01750
0176: 4.20491D-03 8.75593D-03 1.36714D-02 1.89553D-02 2.45797D-02 3.04487D-02 EDC01760
0177: 3.63192D-02 4.16213D-02 4.20797D-02 4.25187D-02 4.29362D-02 4.33298D-02 EDC01770
0178: 4.36973D-02 4.40360D-02 4.43431D-02 4.46158D-02 4.48508D-02 4.50451D-02 EDC01780
0179: 4.51954D-02 4.52986D-02 4.53518D-02 4.53531D-02 4.53017D-02 4.5199 D-02 EDC01790
0180: 4.5053 D-02 4.488  D-02 4.472  D-02 4.47   D-02                         EDC01800
```

```
0181:                                                                         EDC01810
0182: 1.05690D-03 2.07950D-03 3.05758D-03 3.97706D-03 4.81820D-03 5.55256D-03 EDC01820
0183: 6.13803D-03 6.51137D-03 6.53378D-03 6.55310D-03 6.56920D-03 6.58199D-03 EDC01830
0184: 6.59138D-03 6.59729D-03 6.59964D-03 6.59837D-03 6.59346D-03 6.58492D-03 EDC01840
0185: 6.57282D-03 6.55729D-03 6.53861D-03 6.51719D-03 6.49374D-03 6.46933D-03 EDC01850
0186: 6.4457 D-03 6.4256 D-03 6.4138 D-03 6.418  D-03                         EDC01860
0187:                                                                         EDC01870
0188: 2.08704D 00 2.10371D 00 2.12099D 00 2.13884D 00 2.15717D 00 2.17573D 00 EDC01880
0189: 2.19394D 00 2.21050D 00 2.21197D 00 2.21338D 00 2.21475D 00 2.21605D 00 EDC01890
0190: 2.21728D 00 2.21844D 00 2.21952D 00 2.22051D 00 2.22139D 00 2.22216D 00 EDC01900
0191: 2.22281D 00 2.22333D 00 2.22370D 00 2.22392D 00 2.22397D 00 2.22385D 00 EDC01910
0192: 2.22357D 00 2.22316D 00 2.22272D 00 2.2225 D 00                         EDC01920
0193:                                                                         EDC01930
0194: 2.81745D 00 2.85606D 00 2.89673D 00 2.93952D 00 2.98434D 00 3.03081D 00 EDC01940
0195: 3.07778D 00 3.12219D 00 3.12625D 00 3.13020D 00 3.13403D 00 3.13771D 00 EDC01950
0196: 3.14123D 00 3.14458D 00 3.14772D 00 3.15064D 00 3.15330D 00 3.15568D 00 EDC01960
0197: 3.15775D 00 3.15946D 00 3.16079D 00 3.16168D 00 3.16212D 00 3.16206D 00 EDC01970
0198: 3.16152D 00 3.16055D 00 3.15936D 00 3.15862D 00                         EDC01980
0199: A R0=0.3                                                                EDC01990
0200: 1.01407D-01 1.45806D-01 1.81613D-01 2.13322D-01 2.42619D-01 2.70271D-01 EDC02000
0201: 2.96534D-01 3.21119D-01 3.23450D-01 3.25750D-01 3.28018D-01 3.30252D-01 EDC02010
0202: 3.32448D-01 3.34604D-01 3.36717D-01 3.38782D-01 3.40797D-01 3.42757D-01 EDC02020
0203: 3.44658D-01 3.46498D-01 3.48272D-01 3.49978D-01 3.51617D-01 3.53193D-01 EDC02030
0204: 3.54717D-01 3.56219D-01 3.57760D-01 3.5948 D-01                         EDC02040
0205: C2 R0=0.3                                                               EDC02050
0206: 8.50859D-01 8.67918D-01 8.86141D-01 9.05617D-01 9.26393D-01 9.48390D-01 EDC02060
0207: 9.71196D-01 9.93490D-01 9.95578D-01 9.97621D-01 9.99610D-01 1.00153D 00 EDC02070
0208: 1.00339D 00 1.00517D 00 1.00685D 00 1.00843D 00 1.00989D 00 1.01121D 00 EDC02080
0209: 1.01237D 00 1.01336D 00 1.01416D 00 1.01473D 00 1.01507D 00 1.01514D 00 EDC02090
0210: 1.01494D 00 1.01450D 00 1.01391D 00 1.01349D 00                         EDC02100
0211: I R0=0.3                                                                EDC02110
0212: 5.54793D-03 1.12878D-02 1.71966D-02 2.32284D-02 2.92979D-02 3.52476D-02 EDC02120
0213: 4.07832D-02 4.53378D-02 4.57033D-02 4.60472D-02 4.63676D-02 4.66631D-02 EDC02130
0214: 4.69317D-02 4.71715D-02 4.73804D-02 4.75565D-02 4.76976D-02 4.78016D-02 EDC02140
0215: 4.78667D-02 4.78912D-02 4.78741D-02 4.78150D-02 4.77154D-02 4.75796D-02 EDC02150
0216: 4.74172D-02 4.7246 D-02 4.7105 D-02 4.708  D-02                         EDC02160
0217: T R0=0.3                                                                EDC02170
0218: 2.55876D-03 5.25780D-03 8.09400D-03 1.10525D-02 1.40995D-02 1.71630D-02 EDC02180
0219: 2.00958D-02 2.25950D-02 2.28011D-02 2.29962D-02 2.31793D-02 2.33495D-02 EDC02190
0220: 2.35056D-02 2.36466D-02 2.37712D-02 2.38782D-02 2.39664D-02 2.40344D-02 EDC02200
0221: 2.40810D-02 2.41051D-02 2.41059D-02 2.40830D-02 2.40368D-02 2.39692D-02 EDC02210
0222: 2.38850D-02 2.37937D-02 2.3716 D-02 2.3698 D-02                         EDC02220
0223: V R0=0.3                                                                EDC02230
0224: 2.53722D-03 5.16699D-03 7.87762D-03 1.06642D-02 1.34302D-02 1.61518D-02 EDC02240
0225: 1.86670D-02 2.07209D-02 2.08631D-02 2.10123D-02 2.11515D-02 2.12783D-02 EDC02250
0226: 2.13923D-02 2.14927D-02 2.15786D-02 2.16490D-02 2.17032D-02 2.17403D-02 EDC02260
0227: 2.17593D-02 2.17599D-02 2.17416D-02 2.17046D-02 2.16499D-02 2.15800D-02 EDC02270
0228: 2.14995D-02 2.1418 D-02 2.1353 D-02 2.1346 D-02                         EDC02280
0229:     R0=0.3                                                              EDC02290
0230: 6.01453D-03 1.21163D-02 1.82680D-02 2.44088D-02 3.04396D-02 3.61940D-02 EDC02300
0231: 4.13835D-02 4.54861D-02 4.58048D-02 4.61020D-02 4.63767D-02 4.66273D-02 EDC02310
0232: 4.68522D-02 4.70500D-02 4.72189D-02 4.73573D-02 4.74635D-02 4.75360D-02 EDC02320
0233: 4.75733D-02 4.75744D-02 4.75387D-02 4.74666D-02 4.73600D-02 4.72236D-02 EDC02330
0234: 4.70668D-02 4.69079D-02 4.6784 D-02 4.6772 D-02                         EDC02340
0235: K R0=0.3                                                                EDC02350
0236: 8.64670D-01 8.95656D-01 9.27831D-01 9.61139D-01 9.95392D-01 1.03013D 00 EDC02360
0237: 1.06430D 00 1.09541D 00 1.09818D 00 1.10084D 00 1.10341D 00 1.10586D 00 EDC02370
0238: 1.10818D 00 1.11035D 00 1.11238D 00 1.11423D 00 1.11589D 00 1.11733D 00 EDC02380
0239: 1.11855D 00 1.11952D 00 1.12021D 00 1.12060D 00 1.12068D 00 1.12043D 00 EDC02390
0240: 1.11988D 00 1.11909D 00 1.11822D 00 1.1178 D 00                         EDC02400
```

```
0241: SXX R0=0.3                                                                    EDC02410
0242: 4.77957D-03 9.79464D-03 1.50432D-02 2.05061D-02 2.61307D-02 3.17976D-02 EDC02420
0243: 3.72525D-02 4.19552D-02 4.23477D-02 4.27204D-02 4.30716D-02 4.33993D-02 EDC02430
0244: 4.37016D-02 4.39764D-02 4.42212D-02 4.44337D-02 4.46115D-02 4.47520D-02 EDC02440
0245: 4.48528D-02 4.49116D-02 4.49266D-02 4.48967D-02 4.48224D-02 4.47065D-02 EDC02450
0246: 4.4557 D-02 4.4391 D-02 4.4246 D-02 4.420  D-02                         EDC02460
0247: F R0=0.3                                                                      EDC02470
0248: 3.39944D-03 7.07385D-03 1.10362D-02 1.52874D-02 1.98017D-02 2.44974D-02 EDC02480
0249: 2.91729D-02 3.33611D-02 3.37202D-02 3.40633D-02 3.43887D-02 3.46946D-02 EDC02490
0250: 3.49790D-02 3.52400D-02 3.54752D-02 3.56824D-02 3.58592D-02 3.60031D-02 EDC02500
0251: 3.61115D-02 3.61821D-02 3.62128D-02 3.62022D-02 3.61501D-02 3.60584D-02 EDC02510
0252: 3.59334D-02 3.5789 D-02 3.5657 D-02 3.561  D-02                         EDC02520
0253:    R0=0.3                                                                     EDC02530
0254: 1.78208D-03 3.50601D-03 5.15450D-03 6.70368D-03 8.11989D-03 9.35441D-03 EDC02540
0255: 1.03348D-02 1.09519D-02 1.09880D-02 1.10188D-02 1.10441D-02 1.10637D-02 EDC02550
0256: 1.10774D-02 1.10852D-02 1.10869D-02 1.10825D-02 1.10717D-02 1.10548D-02 EDC02560
0257: 1.10319D-02 1.10031D-02 1.09690D-02 1.09304D-02 1.08885D-02 1.08452D-02 EDC02570
0258: 1.08035D-02 1.07685D-02 1.07484D-02 1.07585D-02                         EDC02580
0259: R R0=0.3                                                                      EDC02590
0260: 1.63631D 00 1.65180D 00 1.66788D 00 1.68454D 00 1.70167D 00 1.71904D 00 EDC02600
0261: 1.73612D 00 1.75168D 00 1.75306D 00 1.75439D 00 1.75568D 00 1.75690D 00 EDC02610
0262: 1.75806D 00 1.75915D 00 1.76016D 00 1.76109D 00 1.76192D 00 1.76264D 00 EDC02620
0263: 1.76325D 00 1.76373D 00 1.76408D 00 1.76427D 00 1.76431D 00 1.76419D 00 EDC02630
0264: 1.76392D 00 1.76352D 00 1.76309D 00 1.76287D 00                         EDC02640
0265: S R0=0.3                                                                      EDC02650
0266: 1.75611D 00 1.78366D 00 1.81268D 00 1.84320D 00 1.87517D 00 1.90831D 00 EDC02660
0267: 1.94181D 00 1.97347D 00 1.97637D 00 1.97918D 00 1.98190D 00 1.98452D 00 EDC02670
0268: 1.98703D 00 1.98941D 00 1.99164D 00 1.99371D 00 1.99560D 00 1.99729D 00 EDC02680
0269: 1.99875D 00 1.99995D 00 2.00088D 00 2.00149D 00 2.00178D 00 2.00172D 00 EDC02690
0270: 2.00131D 00 2.00059D 00 1.99974D 00 1.99923D 00                         EDC02700
0271: A R0=0.4                                                                      EDC02710
0272: 8.16165D-02 1.18154D-01 1.48160D-01 1.75182D-01 2.00545D-01 2.24848D-01 EDC02720
0273: 2.48271D-01 2.70520D-01 2.72647D-01 2.74749D-01 2.76826D-01 2.78873D-01 EDC02730
0274: 2.80889D-01 2.82871D-01 2.84816D-01 2.86720D-01 2.88580D-01 2.90392D-01 EDC02740
0275: 2.92152D-01 2.93856D-01 2.95501D-01 2.97085D-01 2.98606D-01 3.00069D-01 EDC02750
0276: 3.01485D-01 3.02877D-01 3.0431 D-01 3.059  D-01                         EDC02760
0277: C2 R0=0.4                                                                     EDC02770
0278: 7.39745D-01 7.56364D-01 7.74088D-01 7.93000D-01 8.13140D-01 8.34428D-01 EDC02780
0279: 8.56460D-01 8.77946D-01 8.79954D-01 8.81918D-01 8.83828D-01 8.85679D-01 EDC02790
0280: 8.87460D-01 8.89162D-01 8.90772D-01 8.92280D-01 8.93670D-01 8.94927D-01 EDC02800
0281: 8.96033D-01 8.96969D-01 8.97713D-01 8.98244D-01 8.98542D-01 8.98588D-01 EDC02810
0282: 8.98377D-01 8.9793 D-01 8.9735 D-01 8.9693 D-01                         EDC02820
0283: I R0=0.4                                                                      EDC02830
0284: 3.84088D-03 7.88489D-03 1.21161D-02 1.65018D-02 2.09784D-02 2.54266D-02 EDC02840
0285: 2.96191D-02 3.31113D-02 3.33934D-02 3.36592D-02 3.39072D-02 3.41361D-02 EDC02850
0286: 3.43444D-02 3.45305D-02 3.46929D-02 3.48299D-02 3.49397D-02 3.50207D-02 EDC02860
0287: 3.50714D-02 3.50903D-02 3.50766D-02 3.50301D-02 3.49518D-02 3.4845 D-02 EDC02870
0288: 3.4717 D-02 3.4582 D-02 3.4471 D-02 3.445  D-02                         EDC02880
0289: T R0=0.4                                                                      EDC02890
0290: 1.65174D-03 3.42871D-03 5.33003D-03 7.34745D-03 9.45857D-03 1.16132D-02 EDC02900
0291: 1.37055D-02 1.55124D-02 1.56625D-02 1.58048D-02 1.59384D-02 1.60628D-02 EDC02910
0292: 1.61771D-02 1.62803D-02 1.63717D-02 1.64502D-02 1.65150D-02 1.65649D-02 EDC02920
0293: 1.65991D-02 1.66168D-02 1.66172D-02 1.66000D-02 1.65656D-02 1.65155D-02 EDC02930
0294: 1.6453 D-02 1.6385 D-02 1.6327 D-02 1.631  D-02                         EDC02940
0295: V R0=0.4                                                                      EDC02950
0296: 1.63656D-03 3.36430D-03 5.17620D-03 7.05728D-03 8.97835D-03 1.08843D-02 EDC02960
0297: 1.26708D-02 1.41363D-02 1.42524D-02 1.43612D-02 1.44620D-02 1.45543D-02 EDC02970
0298: 1.46374D-02 1.47106D-02 1.47733D-02 1.48249D-02 1.48645D-02 1.48916D-02 EDC02980
0299: 1.49055D-02 1.49058D-02 1.48923D-02 1.48649D-02 1.48244D-02 1.47726D-02 EDC02990
0300: 1.47130D-02 1.4652 D-02 1.4604 D-02 1.460  D-02                         EDC03000
```

```
0301:    R0=0.4                                                                EDC03010
0302: 4.46570D-03 9.06630D-03 1.37711D-02 1.85308D-02 2.32643D-02 2.78352D-02 EDC03020
0303: 3.20050D-02 3.53380D-02 3.55984D-02 3.58418D-02 3.60668D-02 3.62723D-02 EDC03030
0304: 3.64570D-02 3.66195D-02 3.67585D-02 3.68724D-02 3.69599D-02 3.70196D-02 EDC03040
0305: 3.70502D-02 3.70509D-02 3.70211D-02 3.69610D-02 3.68723D-02 3.67588D-02 EDC03050
0306: 3.6628 D-02 3.6495 D-02 3.639  D-02 3.638  D-02                         EDC03060
0307:                                                                          EDC03070
0308: 7.50747D-01 7.78588D-01 8.07671D-01 8.37949D-01 8.69257D-01 9.01180D-01 EDC03080
0309: 9.32742D-01 9.61640D-01 9.64213D-01 9.66699D-01 9.69090D-01 9.71373D-01 EDC03090
0310: 9.73539D-01 9.75573D-01 9.77462D-01 9.79190D-01 9.80739D-01 9.82093D-01 EDC03100
0311: 9.83229D-01 9.84128D-01 9.84768D-01 9.85130D-01 9.85196D-01 9.8496 D-01 EDC03110
0312: 9.8444 D-01 9.8371 D-01 9.830  D-01 9.829  D-01                         EDC03120
0313:                                                                          EDC03130
0314: 3.60328D-03 7.40763D-03 1.14095D-02 1.55193D-02 1.99081D-02 2.42618D-02 EDC03140
0315: 2.84465D-02 3.20295D-02 3.23257D-02 3.26062D-02 3.28696D-02 3.31144D-02 EDC03150
0316: 3.33391D-02 3.35418D-02 3.37210D-02 3.38747D-02 3.40011D-02 3.40981D-02 EDC03160
0317: 3.41641D-02 3.41974D-02 3.41967D-02 3.41614D-02 3.40921D-02 3.39917D-02 EDC03170
0318: 3.3867 D-02 3.3732 D-02 3.3617 D-02 3.359  D-02                         EDC03180
0319:                                                                          EDC03190
0320: 2.27038D-03 4.75628D-03 7.46818D-03 1.04078D-02 1.35577D-02 1.68592D-02 EDC03200
0321: 2.01654D-02 2.31333D-02 2.33873D-02 2.36297D-02 2.38593D-02 2.40748D-02 EDC03210
0322: 2.42747D-02 2.44575D-02 2.46217D-02 2.47656D-02 2.48873D-02 2.49851D-02 EDC03220
0323: 2.50572D-02 2.51019D-02 2.51178D-02 2.51040D-02 2.50604D-02 2.49887D-02 EDC03230
0324: 2.48937D-02 2.4786 D-02 2.4690 D-02 2.465  D-02                         EDC03240
0325:                                                                          EDC03250
0326: 2.07004D-03 4.09106D-03 6.04048D-03 7.88740D-03 9.58845D-03 1.10808D-02 EDC03260
0327: 1.22712D-02 1.30183D-02 1.30613D-02 1.30978D-02 1.31273D-02 1.31498D-02 EDC03270
0328: 1.31649D-02 1.31726D-02 1.31727D-02 1.31650D-02 1.31494D-02 1.31262D-02 EDC03280
0329: 1.30953D-02 1.30571D-02 1.30124D-02 1.29620D-02 1.29075D-02 1.28515D-02 EDC03290
0330: 1.27979D-02 1.27530D-02 1.27276D-02 1.2741 D-02                         EDC03300
0331:                                                                          EDC03310
0332: 1.29166D 00 1.30558D 00 1.32013D 00 1.33526D 00 1.35092D 00 1.36688D 00 EDC03320
0333: 1.38266D 00 1.39711D 00 1.39840D 00 1.39964D 00 1.40084D 00 1.40198D 00 EDC03330
0334: 1.40306D 00 1.40408D 00 1.40502D 00 1.40588D 00 1.40666D 00 1.40734D 00 EDC03340
0335: 1.40791D 00 1.40835D 00 1.40867D 00 1.40886D 00 1.40889D 00 1.40877D 00 EDC03350
0336: 1.40851D 00 1.40815D 00 1.4078 D 00 1.4077 D 00                         EDC03360
0337:                                                                          EDC03370
0338: 1.10202D 00 1.12174D 00 1.14255D 00 1.16446D 00 1.18745D 00 1.21130D 00 EDC03380
0339: 1.23543D 00 1.25826D 00 1.26034D 00 1.26237D 00 1.26433D 00 1.26622D 00 EDC03390
0340: 1.26802D 00 1.26973D 00 1.27134D 00 1.27282D 00 1.27418D 00 1.27538D 00 EDC03400
0341: 1.27642D 00 1.27728D 00 1.27793D 00 1.27836D 00 1.27855D 00 1.27848D 00 EDC03410
0342: 1.27816D 00 1.2776 D 00 1.2770 D 00 1.276  D 00                         EDC03420
0343:                                                                          EDC03430
0344: 6.02076D-02 8.83127D-02 1.12137D-01 1.34191D-01 1.55402D-01 1.76177D-01 EDC03440
0345: 1.96607D-01 2.16384D-01 2.18293D-01 2.20184D-01 2.22054D-01 2.23901D-01 EDC03450
0346: 2.25723D-01 2.27518D-01 2.29281D-01 2.31009D-01 2.32700D-01 2.34349D-01 EDC03460
0347: 2.35952D-01 2.37505D-01 2.39005D-01 2.40448D-01 2.41832D-01 2.4316 D-01 EDC03470
0348: 2.4443 D-01 2.457  D-01 2.469  D-01 2.484  D-01                         EDC03480
0349:                                                                          EDC03490
0350: 6.15059D-01 6.31112D-01 6.48237D-01 6.66501D-01 6.85933D-01 7.06443D-01 EDC03500
0351: 7.27629D-01 7.48230D-01 7.50151D-01 7.52028D-01 7.53853D-01 7.55618D-01 EDC03510
0352: 7.57316D-01 7.58937D-01 7.60469D-01 7.61900D-01 7.63217D-01 7.64403D-01 EDC03520
0353: 7.65444D-01 7.66318D-01 7.67008D-01 7.67492D-01 7.67748D-01 7.67764D-01 EDC03530
0354: 7.6754 D-01 7.6707 D-01 7.6648 D-01 7.660  D-01                         EDC03540
0355:                                                                          EDC03550
0356: 2.27828D-03 4.76432D-03 7.44898D-03 1.03114D-02 1.33093D-02 1.63593D-02 EDC03560
0357: 1.92985D-02 2.18008D-02 2.20058D-02 2.21993D-02 2.23805D-02 2.25481D-02 EDC03570
0358: 2.27012D-02 2.28386D-02 2.29589D-02 2.30609D-02 2.31434D-02 2.32049D-02 EDC03580
0359: 2.32444D-02 2.32605D-02 2.32527D-02 2.32206D-02 2.3165 D-02 2.3087 D-02 EDC03590
0360: 2.2993 D-02 2.290  D-02 2.281  D-02 2.28   D-02                         EDC03600
```

```
0361:                                                                         EDC03610
0362: 8.93380D-04 1.89245D-03 2.99870D-03 4.20911D-03 5.51145D-03 6.87501D-03 EDC03620
0363: 8.23092D-03 9.42888D-03 9.52974D-03 9.62558D-03 9.71559D-03 9.80014D-03 EDC03630
0364: 9.87776D-03 9.94813D-03 1.00106D-02 1.00646D-02 1.01093D-02 1.01440D-02 EDC03640
0365: 1.01682D-02 1.01811D-02 1.01822D-02 1.01714D-02 1.01487D-02 1.0114 D-02 EDC03650
0366: 1.0073 D-02 1.0027 D-02 9.987  D-03 9.97   D-03                         EDC03660
0367:                                                                         EDC03670
0368: 8.83775D-04 1.85131D-03 2.89960D-03 4.02062D-03 5.19710D-03 6.39438D-03 EDC03680
0369: 7.54417D-03 8.51053D-03 8.58833D-03 8.66141D-03 8.72937D-03 8.79180D-03 EDC03690
0370: 8.84823D-03 8.89823D-03 8.94130D-03 8.97694D-03 9.00467D-03 9.02400D-03 EDC03700
0371: 9.03450D-03 9.03579D-03 9.02767D-03 9.0101 D-03 8.9836 D-03 8.9491 D-03 EDC03710
0372: 8.909  D-03 8.868  D-03 8.835  D-03 8.83   D-03                         EDC03720
0373:                                                                         EDC03730
0374: 2.90502D-03 5.99719D-03 9.25188D-03 1.26305D-02 1.60699D-02 1.94637D-02 EDC03740
0375: 2.26240D-02 2.52032D-02 2.54075D-02 2.55990D-02 2.57766D-02 2.59394D-02 EDC03750
0376: 2.60862D-02 2.62160D-02 2.63276D-02 2.64197D-02 2.64913D-02 2.65411D-02 EDC03760
0377: 2.65681D-02 2.65714D-02 2.65505D-02 2.65055D-02 2.64376D-02 2.6349 D-02 EDC03770
0378: 2.6245 D-02 2.614  D-02 2.606  D-02 2.606  D-02                         EDC03780
0379:                                                                         EDC03790
0380: 6.22675D-01 6.45724D-01 6.72150D-01 6.98909D-01 7.26854D-01 7.55613D-01 EDC03800
0381: 7.84307D-01 8.10826D-01 8.13200D-01 8.15497D-01 8.17708D-01 8.19822D-01 EDC03810
0382: 8.21830D-01 8.23718D-01 8.25473D-01 8.27081D-01 8.28525D-01 8.29787D-01 EDC03820
0383: 8.30849D-01 8.31691D-01 8.32292D-01 8.32633D-01 8.32699D-01 8.3248 D-01 EDC03830
0384: 8.3199 D-01 8.313  D-01 8.305  D-01 8.303  D-01                         EDC03840
0385:                                                                         EDC03850
0386: 2.17915D-03 4.54525D-03 7.09579D-03 9.81850D-03 1.26818D-02 1.56158D-02 EDC03860
0387: 1.84728D-02 2.09403D-02 2.11447D-02 2.13381D-02 2.15196D-02 2.16880D-02 EDC03870
0388: 2.18424D-02 2.19813D-02 2.21037D-02 2.22080D-02 2.22930D-02 2.23574D-02 EDC03880
0389: 2.23997D-02 2.24188D-02 2.24140D-02 2.23847D-02 2.23316D-02 2.2256 D-02 EDC03890
0390: 2.2165 D-02 2.206  D-02 2.198  D-02 2.19   D-02                         EDC03900
0391:                                                                         EDC03910
0392: 1.21065D-03 2.58210D-03 4.12374D-03 5.83983D-03 7.72253D-03 9.73756D-03 EDC03920
0393: 1.17925D-02 1.36648D-02 1.38260D-02 1.39800D-02 1.41260D-02 1.42630D-02 EDC03930
0394: 1.43901D-02 1.45063D-02 1.46106D-02 1.47019D-02 1.47788D-02 1.48404D-02 EDC03940
0395: 1.48853D-02 1.49125D-02 1.49210D-02 1.49102D-02 1.48800D-02 1.48318D-02 EDC03950
0396: 1.47686D-02 1.46975D-02 1.4634 D-02 1.4613 D-02                         EDC03960
0397:                                                                         EDC03970
0398: 1.80583D-03 3.61783D-03 5.40970D-03 7.14661D-03 8.78050D-03 1.02425D-02 EDC03980
0399: 1.14306D-02 1.21896D-02 1.22339D-02 1.22714D-02 1.23019D-02 1.23252D-02 EDC03990
0400: 1.23410D-02 1.23491D-02 1.23493D-02 1.23414D-02 1.23255D-02 1.23014D-02 EDC04000
0401: 1.22693D-02 1.22296D-02 1.21829D-02 1.21302D-02 1.20731D-02 1.20143D-02 EDC04010
0402: 1.19577D-02 1.1910 D-02 1.1883 D-02 1.189  D-02                         EDC04020
0403:                                                                         EDC04030
0404: 1.00448D 00 1.01651D 00 1.02922D 00 1.04260D 00 1.05657D 00 1.07095D 00 EDC04040
0405: 1.08530D 00 1.09856D 00 1.09975D 00 1.10090D 00 1.10200D 00 1.10306D 00 EDC04050
0406: 1.10406D 00 1.10501D 00 1.10588D 00 1.10669D 00 1.10741D 00 1.10804D 00 EDC04060
0407: 1.10857D 00 1.10899D 00 1.10929D 00 1.10946D 00 1.10949D 00 1.10939D 00 EDC04070
0408: 1.10914D 00 1.10878D 00 1.1084 D 00 1.1082 D 00                         EDC04080
0409:                                                                         EDC04090
0410: 6.68936D-01 6.82615D-01 6.97104D-01 7.12445D-01 7.28606D-01 7.45442D-01 EDC04100
0411: 7.62529D-01 7.78730D-01 7.80211D-01 7.81651D-01 7.83044D-01 7.84385D-01 EDC04110
0412: 7.85667D-01 7.86882D-01 7.88021D-01 7.89076D-01 7.90036D-01 7.90890D-01 EDC04120
0413: 7.91623D-01 7.92226D-01 7.92681D-01 7.92977D-01 7.93099D-01 7.9304 D-01 EDC04130
0414: 7.9279 D-01 7.924  D-01 7.919  D-01 7.92   D-01                         EDC04140
0415:                                                                         EDC04150
0416: 3.99211D-02 5.99596D-02 7.77954D-02 9.49578D-02 1.12000D-01 1.29152D-01 EDC04160
0417: 1.45424D-01 1.63500D-01 1.65166D-01 1.66818D-01 1.68456D-01 1.70076D-01 EDC04170
0418: 1.71677D-01 1.73256D-01 1.74809D-01 1.76333D-01 1.77825D-01 1.79282D-01 EDC04180
0419: 1.80698D-01 1.82071D-01 1.83395D-01 1.84666D-01 1.8588 D-01 1.8704 D-01 EDC04190
0420: 1.8815 D-01 1.8921 D-01 1.902  D-01 1.914  D-01                         EDC04200
```

```
0421:                                                                     EDC04210
0422: 4.84437D-01 4.99333D-01 5.15320D-01 5.32433D-01 5.50678D-01 5.69948D-01 EDC04220
0423: 5.89843D-01 6.09151D-01 6.10947D-01 6.12701D-01 6.14405D-01 6.16053D-01 EDC04230
0424: 6.17635D-01 6.19144D-01 6.20568D-01 6.21896D-01 6.23115D-01 6.24210D-01 EDC04240
0425: 6.25165D-01 6.25964D-01 6.26585D-01 6.27010D-01 6.2722 D-01 6.2720 D-01 EDC04250
0426: 6.2693 D-01 6.2643 D-01 6.257  D-01 6.249  D-01                         EDC04260
0427:                                                                     EDC04270
0428: 1.11395D-03 2.40760D-03 3.87720D-03 5.51241D-03 7.28956D-03 9.15842D-03 EDC04280
0429: 1.10153D-02 1.26445D-02 1.27805D-02 1.29095D-02 1.30307D-02 1.31434D-02 EDC04290
0430: 1.32469D-02 1.33403D-02 1.34227D-02 1.34932D-02 1.35509D-02 1.35950D-02 EDC04300
0431: 1.36244D-02 1.36383D-02 1.36363D-02 1.36182D-02 1.35845D-02 1.3538 D-02 EDC04310
0432: 1.3485 D-02 1.3449 D-02 1.35   D-02 1.32   D-02                         EDC04320
0433:                                                                     EDC04330
0434: 3.87661D-04 8.50650D-04 1.39163D-03 2.01115D-03 2.70471D-03 3.45707D-03 EDC04340
0435: 4.22993D-03 4.93439D-03 4.99483D-03 5.05247D-03 5.10700D-03 5.15808D-03 EDC04350
0436: 5.20536D-03 5.24845D-03 5.28694D-03 5.32040D-03 5.34840D-03 5.37046D-03 EDC04360
0437: 5.38617D-03 5.39509D-03 5.39689D-03 5.3913 D-03 5.3784 D-03 5.3586 D-03 EDC04370
0438: 5.3333 D-03 5.305  D-03 5.279  D-03 5.26   D-03                         EDC04380
0439:                                                                     EDC04390
0440: 3.82663D-04 8.28720D-04 1.33768D-03 1.90660D-03 2.52746D-03 3.18215D-03 EDC04400
0441: 3.83222D-03 4.39736D-03 4.44389D-03 4.48780D-03 4.52885D-03 4.56676D-03 EDC04410
0442: 4.60127D-03 4.63208D-03 4.65887D-03 4.68134D-03 4.69916D-03 4.71202D-03 EDC04420
0443: 4.71962D-03 4.72172D-03 4.7181 D-03 4.7088 D-03 4.6941 D-03 4.675  D-03 EDC04430
0444: 4.652  D-03 4.631  D-03 4.62   D-03 4.65   D-03                         EDC04440
0445:                                                                     EDC04450
0446: 1.60045D-03 3.40714D-03 5.40107D-03 7.55455D-03 9.82319D-03 1.21312D-02 EDC04460
0447: 1.43426D-02 1.62009D-02 1.63511D-02 1.64924D-02 1.66242D-02 1.67456D-02 EDC04470
0448: 1.68558D-02 1.69539D-02 1.70390D-02 1.71102D-02 1.71666D-02 1.72071D-02 EDC04480
0449: 1.72310D-02 1.7237 D-02 1.7226 D-02 1.7196 D-02 1.715  D-02 1.708  D-02 EDC04490
0450: 1.700  D-02 1.693  D-02 1.68   D-02 1.68   D-02                         EDC04500
0451:                                                                     EDC04510
0452: 4.88845D-01 5.08636D-01 5.29939D-01 5.52703D-01 5.76796D-01 6.01896D-01 EDC04520
0453: 6.27234D-01 6.50937D-01 6.53076D-01 6.55147D-01 6.57145D-01 6.59058D-01 EDC04530
0454: 6.60878D-01 6.62592D-01 6.64190D-01 6.65656D-01 6.66976D-01 6.68133D-01 EDC04540
0455: 6.69110D-01 6.69889D-01 6.70448D-01 6.7077 D-01 6.7083 D-01 6.7063 D-01 EDC04550
0456: 6.7018 D-01 6.694  D-01 6.686  D-01 6.677  D-01                         EDC04560
0457:                                                                     EDC04570
0458: 1.02113D-03 2.19612D-03 3.52432D-03 4.99993D-03 6.60573D-03 8.30086D-03 EDC04580
0459: 9.99494D-03 1.14914D-02 1.16168D-02 1.17357D-02 1.18475D-02 1.19515D-02 EDC04590
0460: 1.20469D-02 1.21330D-02 1.22089D-02 1.22737D-02 1.23266D-02 1.23667D-02 EDC04600
0461: 1.23931D-02 1.2405 D-02 1.2402 D-02 1.2383 D-02 1.2349 D-02 1.230  D-02 EDC04610
0462: 1.224  D-02 1.218  D-02 1.21   D-02 1.21   D-02                         EDC04620
0463:                                                                     EDC04630
0464: 4.92681D-04 1.08722D-03 1.79125D-03 2.61029D-03 3.54388D-03 4.57729D-03 EDC04640
0465: 5.66311D-03 6.67891D-03 6.76763D-03 6.85257D-03 6.93324D-03 7.00915D-03 EDC04650
0466: 7.07977D-03 7.14448D-03 7.20270D-03 7.25374D-03 7.29692D-03 7.33157D-03 EDC04660
0467: 7.3569 D-03 7.3724 D-03 7.3774 D-03 7.3715 D-03 7.354  D-03 7.327  D-03 EDC04670
0468: 7.293  D-03 7.25   D-03 7.21   D-03 7.2    D-03                         EDC04680
0469:                                                                     EDC04690
0470: 1.20695D-03 2.48852D-03 3.81773D-03 5.16057D-03 6.47135D-03 7.68517D-03 EDC04700
0471: 8.70555D-03 9.38466D-03 9.42602D-03 9.46157D-03 9.49108D-03 9.51428D-03 EDC04710
0472: 9.53090D-03 9.54074D-03 9.54358D-03 9.53924D-03 9.52761D-03 9.5086 D-03 EDC04720
0473: 9.4824 D-03 9.4490 D-03 9.4092 D-03 9.3636 D-03 9.314  D-03 9.262  D-03 EDC04730
0474: 9.211  D-03 9.17   D-03 9.14   D-03 9.1    D-03                         EDC04740
0475:                                                                     EDC04750
0476: 7.55248D-01 7.65144D-01 7.75795D-01 7.87177D-01 7.99224D-01 8.11773D-01 EDC04760
0477: 8.24443D-01 8.36294D-01 8.37363D-01 8.38399D-01 8.39398D-01 8.40355D-01 EDC04770
0478: 8.41265D-01 8.42122D-01 8.42920D-01 8.43653D-01 8.44314D-01 8.44892D-01 EDC04780
0479: 8.45380D-01 8.45770D-01 8.46049D-01 8.46209D-01 8.4624 D-01 8.4614 D-01 EDC04790
0480: 8.4591 D-01 8.455  D-01 8.451  D-01 8.447  D-01                         EDC04800
```

```
0481:                                                                         EDC04810
0482: 3.78999D-01 3.87785D-01 3.97224D-01 4.07317D-01 4.18034D-01 4.29273D-01 EDC04820
0483: 4.40743D-01 4.51671D-01 4.52672D-01 4.53646D-01 4.54589D-01 4.55497D-01 EDC04830
0484: 4.56364D-01 4.57186D-01 4.57958D-01 4.58671D-01 4.59321D-01 4.59897D-01 EDC04840
0485: 4.60393D-01 4.60798D-01 4.61103D-01 4.61299D-01 4.6137 D-01 4.6131 D-01 EDC04850
0486: 4.6113 D-01 4.6082 D-01 4.604  D-01 4.600  D-01                         EDC04860
0487:                                                                         EDC04870
0488: 2.30969D-02 3.61956D-02 4.86378D-02 6.11638D-02 7.40387D-02 8.73675D-02 EDC04880
0489: 1.01119D-01 1.15005D-01 1.16374D-01 1.17735D-01 1.19085D-01 1.20423D-01 EDC04890
0490: 1.21747D-01 1.23054D-01 1.24341D-01 1.25606D-01 1.2684 D-01 1.2805 D-01 EDC04900
0491: 1.2923 D-01 1.3036 D-01 1.3146 D-01 1.3250 D-01 1.3349 D-01 1.344  D-01 EDC04910
0492: 1.353  D-01 1.361  D-01 1.368  D-01 1.374  D-01                         EDC04920
0493:                                                                         EDC04930
0494: 3.53910D-01 3.66691D-01 3.80583D-01 3.95576D-01 4.11646D-01 4.28680D-01 EDC04940
0495: 4.46301D-01 4.63400D-01 4.64988D-01 4.66538D-01 4.68044D-01 4.69498D-01 EDC04950
0496: 4.70893D-01 4.72222D-01 4.73474D-01 4.74641D-01 4.7571 D-01 4.7666 D-01 EDC04960
0497: 4.7749 D-01 4.7818 D-01 4.7871 D-01 4.790  D-01 4.792  D-01 4.792  D-01 EDC04970
0498: 4.789  D-01 4.783  D-01 4.775  D-01 4.76   D-01                         EDC04980
0499:                                                                         EDC04990
0500: 4.22398D-04 9.64986D-04 1.62559D-03 2.40086D-03 3.28155D-03 4.24442D-03 EDC05000
0501: 5.23600D-03 6.13736D-03 6.21434D-03 6.28767D-03 6.35692D-03 6.42167D-03 EDC05010
0502: 6.4815 D-03 6.5358 D-03 6.5842 D-03 6.6260 D-03 6.6606 D-03 6.6876 D-03 EDC05020
0503: 6.706  D-03 6.716  D-03 6.717  D-03 6.708  D-03 6.689  D-03 6.66   D-03 EDC05030
0504: 6.62   D-03 6.58   D-03 6.53   D-03 6.48   D-03                         EDC05040
0505:                                                                         EDC05050
0506: 1.25643D-04 2.92174D-04 5.01426D-04 7.53010D-04 1.05272D-03 1.38949D-03 EDC05060
0507: 1.74897D-03 2.08896D-03 2.11878D-03 2.14735D-03 2.17450D-03 2.20006D-03 EDC05070
0508: 2.22385D-03 2.2456 D-03 2.2553 D-03 2.2824 D-03 2.2969 D-03 2.3085 D-03 EDC05080
0509: 2.3171 D-03 2.322  D-03 2.324  D-03 2.321  D-03 2.315  D-03 2.305  D-03 EDC05190
0510: 2.291  D-03 2.274  D-03 2.25   D-03 2.24   D-03                         EDC05100
0511:                                                                         EDC05110
0512: 1.23741D-04 2.83384D-04 4.78885D-04 7.09813D-04 9.73850D-04 1.26413D-03 EDC05120
0513: 1.56388D-03 1.83503D-03 1.85794D-03 1.87969D-03 1.90014D-03 1.91915D-03 EDC05130
0514: 1.93360D-03 1.9523 D-03 1.9661 D-03 1.9778 D-03 1.9873 D-03 1.9944 D-03 EDC05140
0515: 1.9989 D-03 2.0007 D-03 1.999  D-03 1.996  D-03 1.988  D-03 1.978  D-03 EDC05150
0516: 1.966  D-03 1.951  D-03 1.94   D-03 1.92   D-03                         EDC05160
0517:                                                                         EDC05170
0518: 7.10029D-04 1.59357D-03 2.63504D-03 3.81727D-03 5.11467D-03 6.48263D-03 EDC05180
0519: 7.83765D-03 9.01579D-03 9.11328D-03 9.20547D-03 9.29189D-03 9.37200D-03 EDC05190
0520: 9.44524D-03 9.51101D-03 9.56870D-03 9.61767D-03 9.65722D-03 9.68671D-03 EDC05200
0521: 9.70549D-03 9.71296D-03 9.7086 D-03 9.6922 D-03 9.6639 D-03 9.6243 D-03 EDC05210
0522: 9.575  D-03 9.519  D-03 9.464  D-03 9.421  D-03                         EDC05220
0523:                                                                         EDC05230
0524: 3.55936D-01 3.71197D-01 3.87965D-01 4.06171D-01 4.25706D-01 4.46316D-01 EDC05240
0525: 4.67385D-01 4.87360D-01 4.89177D-01 4.90940D-01 4.92644D-01 4.94279D-01 EDC05250
0526: 4.95837D-01 4.97308D-01 4.98682D-01 4.9994 D-01 5.0108 D-01 5.0209 D-01 EDC05260
0527: 5.029  D-01 5.036  D-01 5.041  D-01 5.044  D-01 5.045  D-01 5.04   D-01 EDC05270
0528: 5.04   D-01 5.03   D-01 5.02   D-01 5.0    D-01                         EDC05280
0529:                                                                         EDC05290
0530: 3.48087D-04 7.91006D-04 1.32808D-03 1.95800D-03 2.67496D-03 3.46180D-03 EDC05300
0531: 4.27599D-03 5.01870D-03 5.08211D-03 5.14245D-03 5.19939D-03 5.2525 D-03 EDC05310
0532: 5.3015 D-03 5.3459 D-03 5.3853 D-03 5.4191 D-03 5.4468 D-03 5.468  D-03 EDC05320
0533: 5.482  D-03 5.489  D-03 5.488  D-03 5.479  D-03 5.46   D-03 5.43   D-03 EDC05330
0534: 5.40   D-03 5.37   D-03 5.32   D-03 5.28   D-03                         EDC05340
0535:                                                                         EDC05350
0536: 1.41606D-04 3.31257D-04 5.72994D-04 8.70939D-04 1.22742D-03 1.63911D-03 EDC05360
0537: 2.08850D-03 2.52384D-03 2.55260D-03 2.59985D-03 2.63535D-03 2.66888D-03 EDC05370
0538: 2.70020D-03 2.7290 D-03 2.7550 D-03 2.7779 D-03 2.7974 D-03 2.813  D-03 EDC05380
0539: 2.825  D-03 2.832  D-03 2.834  D-03 2.832  D-03 2.824  D-03 2.81   D-03 EDC05390
0540: 2.79   D-03 2.77   D-03 2.74   D-03 2.72   D-03                         EDC05400
```

```
0541:                                                                         EDC05410
0542: 6.06452D-04 1.31874D-03 2.11247D-03 2.96095D-03 3.82997D-03 4.67079D-03 EDC05420
0543: 5.40924D-03 5.92840D-03 5.96194D-03 5.99134D-03 6.01638D-03 6.03682D-03 EDC05430
0544: 6.0525 D-03 6.0631 D-03 6.0684 D-03 6.0685 D-03 6.0630 D-03 6.0519 D-03 EDC05440
0545: 6.035  D-03 6.013  D-03 5.985  D-03 5.952  D-03 5.916  D-03 5.876  D-03 EDC05450
0546: 5.836  D-03 5.801  D-03 5.77   D-03 5.76   D-03                         EDC05460
0547:                                                                         EDC05470
0548: 5.34643D-01 5.42273D-01 5.50658D-01 5.59761D-01 5.69528D-01 5.79833D-01 EDC05480
0549: 5.90368D-01 6.00355D-01 6.01263D-01 6.02145D-01 6.02997D-01 6.03814D-01 EDC05490
0550: 6.04593D-01 6.05329D-01 6.0501 D-01 6.0665 D-01 6.0722 D-01 6.0772 D-01 EDC05500
0551: 6.0814 D-01 6.0848 D-01 6.087  D-01 6.089  D-01 6.089  D-01 6.088  D-01 EDC05510
0552: 6.086  D-01 6.08   D-01 6.08   D-01 6.07   D-01                         EDC05520
0553:                                                                         EDC05530
0554: 1.90190D-01 1.95174D-01 2.00621D-01 2.06517D-01 2.12839D-01 2.19524D-01 EDC05540
0555: 2.26397D-01 2.32989D-01 2.33595D-01 2.34184D-01 2.34756D-01 2.35306D-01 EDC05550
0556: 2.35832D-01 2.36330D-01 2.36798D-01 2.37232D-01 2.3763 D-01 2.3797 D-01 EDC05560
0557: 2.3828 D-01 2.3852 D-01 2.3871 D-01 2.3882 D-01 2.3886 D-01 2.3883 D-01 EDC05570
0558: 2.386  D-01 2.385  D-01 2.383  D-01 2.380  D-01                         EDC05580
0559:                                                                         EDC05590
0560: 1.10585D-02 1.86852D-02 2.64278D-02 3.45424D-02 4.31421D-02 5.22822D-02 EDC05600
0561: 6.19407D-02 7.1910 D-02 7.2903 D-02 7.3892 D-02 7.4876 D-02 7.5851 D-02 EDC05610
0562: 7.6818 D-02 7.777  D-02 7.871  D-02 7.963  D-02 8.055  D-02 8.143  D-02 EDC05620
0563: 8.228  D-02 8.311  D-02 8.390  D-02 8.464  D-02 8.534  D-02 8.598  D-02 EDC05630
0564: 8.65   D-02 8.70   D-02 8.74   D-02 8.77   D-02                         EDC05640
0565:                                                                         EDC05650
0566: 2.27975D-01 2.37587D-01 2.48167D-01 2.59680D-01 2.72109D-01 2.85359D-01 EDC05660
0567: 2.99161D-01 3.12589D-01 3.13836D-01 3.15053D-01 3.16233D-01 3.17371D-01 EDC05670
0568: 3.18452D-01 3.19498D-01 3.2047 D-01 3.2137 D-01 3.2219 D-01 3.2291 D-01 EDC05680
0569: 3.2352 D-01 3.2402 D-01 3.2437 D-01 3.2454 D-01 3.245  D-01 3.243  D-01 EDC05690
0570: 3.238  D-01 3.230  D-01 3.219  D-01 3.205  D-01                         EDC05700
0571:                                                                         EDC05710
0572: 1.09966D-04 2.74402D-04 4.90670D-04 7.58669D-04 1.07720D-03 1.44002D-03 EDC05720
0573: 1.82848D-03 2.1956 D-03 2.2277 D-03 2.2583 D-03 2.2875 D-03 2.314  D-03 EDC05730
0574: 2.340  D-03 2.363  D-03 2.384  D-03 2.402  D-03 2.417  D-03 2.429  D-03 EDC05740
0575: 2.436  D-03 2.44   D-03 2.43   D-03 2.43   D-03 2.42   D-03 2.40   D-03 EDC05750
0576: 2.38   D-03 2.35   D-03 2.31   D-03 2.27   D-03                         EDC05760
0577:                                                                         EDC05770
0578: 2.62526D-05 6.68759D-05 1.22217D-04 1.93305D-04 2.80956D-04 3.84628D-04 EDC05780
0579: 5.00048D-04 6.1373 D-04 6.239  D-04 6.337  D-04 6.431  D-04 6.519  D-04 EDC05790
0580: 6.602  D-04 6.677  D-04 6.745  D-04 6.805  D-04 6.854  D-04 6.894  D-04 EDC05800
0581: 6.922  D-04 6.935  D-04 6.934  D-04 6.916  D-04 6.87   D-04 6.82   D-04 EDC05810
0582: 6.74   D-04 6.63   D-04 6.50   D-04 6.34   D-04                         EDC05820
0583:                                                                         EDC05830
0584: 2.58047D-05 6.46052D-05 1.16017D-04 1.80270D-04 2.57319D-04 3.45816D-04 EDC05840
0585: 4.41159D-04 5.312  D-04 5.390  D-04 5.465  D-04 5.535  D-04 5.601  D-04 EDC05850
0586: 5.661  D-04 5.716  D-04 5.764  D-04 5.805  D-04 5.84   D-04 5.86   D-04 EDC05860
0587: 5.87   D-04 5.88   D-04 5.87   D-04 5.85   D-04 5.81   D-04 5.76   D-04 EDC05870
0588: 5.70   D-04 5.6    D-04 5.5    D-04 5.4    D-04                         EDC05880
0589:                                                                         EDC05890
0590: 2.30316D-04 5.62959D-04 9.84957D-04 1.48789D-03 2.06503D-03 2.69566D-03 EDC05900
0591: 3.34302D-03 3.927  D-03 3.976  D-03 4.023  D-03 4.067  D-03 4.108  D-03 EDC05910
0592: 4.146  D-03 4.181  D-03 4.212  D-03 4.238  D-03 4.26   D-03 4.27   D-03 EDC05920
0593: 4.28   D-03 4.29   D-03 4.28   D-03 4.27   D-03 4.26   D-03 4.2    D-03 EDC05930
0594: 4.2    D-03 4.1    D-03 4.1    D-03 4.0    D-03                         EDC05940
0595:                                                                         EDC05950
0596: 2.28648D-01 2.39218D-01 2.50996D-01 2.63920D-01 2.77934D-01 2.92894D-01 EDC05960
0597: 3.08387D-01 3.23286D-01 3.24653D-01 3.25982D-01 3.27268D-01 3.28504D-01 EDC05970
0598: 3.29682D-01 3.30796D-01 3.31836D-01 3.3279 D-01 3.3365 D-01 3.3441 D-01 EDC05980
0599: 3.3504 D-01 3.3554 D-01 3.3587 D-01 3.3603 D-01 3.3598 D-01 3.3570 D-01 EDC05990
0600: 3.3515 D-01 3.3429 D-01 3.3309 D-01 3.315  D-01                         EDC06000
```

```
0601:                                                                       EDC06010
0602: 7.60472D-05 1.86659D-04 3.33424D-04 5.15787D-04 7.33549D-04 9.83053D-04 EDC06020
0603: 1.2518 D-03 1.506  D-03 1.528  D-03 1.55   D-03 1.57   D-03 1.59   D-03 EDC06030
0604: 1.60   D-03 1.62   D-03 1.63   D-03 1.65   D-03 1.66   D-03 1.67   D-03 EDC06040
0605: 1.67   D-03 1.67   D-03 1.67   D-03 1.6    D-03 1.6    D-03 1.6    D-03 EDC06050
0606: 1.6    D-03 1.6    D-03 1.5    D-03 1.     D-03                      EDC06060
0607:                                                                       EDC06070
0608: 2.43023D-05 6.24129D-05 1.15247D-04 1.84472D-04 2.71682D-04 3.77210D-04 EDC06080
0609: 4.9751 D-04 6.1888 D-04 6.2992 D-04 6.4056 D-04 6.5073 D-04 6.603  D-04 EDC06090
0610: 6.693  D-04 6.776  D-04 6.851  D-04 6.916  D-04 6.971  D-04 7.014  D-04 EDC06100
0611: 7.044  D-04 7.059  D-04 7.057  D-04 7.037  D-04 6.996  D-04 6.93   D-04 EDC06110
0612: 6.84   D-04 6.72   D-04 6.57   D-04 6.38   D-04                      EDC06120
0613:                                                                       EDC06130
0614: 2.12428D-04 5.04997D-04 8.59365D-04 1.26152D-03 1.69478D-03 2.13422D-03 EDC06140
0615: 2.5391 D-03 2.841  D-03 2.862  D-03 2.880  D-03 2.896  D-03 2.91   D-03 EDC06150
0616: 2.92   D-03 2.93   D-03 2.93   D-03 2.93   D-03 2.93   D-03 2.92   D-03 EDC06160
0617: 2.92   D-03 2.91   D-03 2.9    D-03 2.9    D-03 2.8    D-03 2.8    D-03 EDC06170
0618: 2.7    D-03 2.7    D-03 2.6    D-03 2.     D-03                      EDC06180
0619:                                                                       EDC06190
0620: 3.37468D-01 3.42753D-01 3.48642D-01 3.55103D-01 3.62111D-01 3.69591D-01 EDC06200
0621: 3.77337D-01 3.84786D-01 3.85470D-01 3.86135D-01 3.86777D-01 3.87395D-01 EDC06210
0622: 3.87984D-01 3.88542D-01 3.89062D-01 3.89540D-01 3.89971D-01 3.9035 D-01 EDC06220
0623: 3.9066 D-01 3.9091 D-01 3.9108 D-01 3.9116 D-01 3.9113 D-01 3.9099 D-01 EDC06230
0624: 3.9071 D-01 3.9029 D-01 3.8969 D-01 3.889  D-01                      EDC06240
0625:                                                                       EDC06250
0626: 7.58416D-02 7.80912D-02 8.05827D-02 8.33050D-02 8.62505D-02 8.93937D-02 EDC06260
0627: 9.26557D-02 9.58120D-02 9.61034D-02 9.63871D-02 9.66620D-02 9.69268D-02 EDC06270
0628: 9.7180 D-02 9.7420 D-02 9.7644 D-02 9.7851 D-02 9.8039 D-02 9.8204 D-02 EDC06280
0629: 9.8343 D-02 9.8453 D-02 9.8530 D-02 9.857  D-02 9.856  D-02 9.851  D-02 EDC06290
0630: 9.839  D-02 9.821  D-02 9.796  D-02 9.761  D-02                      EDC06300
0631:                                                                       EDC06310
0632: 3.80037D-03 7.16574D-03 1.07501D-02 1.463  D-02 1.88   D-02 2.36   D-02 EDC06320
0633: 2.87   D-02 3.44   D-02 3.50   D-02 3.56   D-02 3.62   D-02 3.68   D-02 EDC06330
0634: 3.76   D-02 3.8    D-02 3.9    D-02 3.9    D-02 4.0    D-02 4.1    D-02 EDC06340
0635: 4.1    D-02 4.2    D-02 4.3    D-02 4.3    D-02 4.4    D-02 4.5    D-02 EDC06350
0636: 4.6    D-02 4.6    D-02 4.7    D-02 4.8    D-02                      EDC06360
0637:                                                                       EDC06370
0638: 1.09612D-01 1.14960D-01 1.20861D-01 1.27330D-01 1.34381D-01 1.41983D-01 EDC06380
0639: 1.4998 D-01 1.578  D-01 1.585  D-01 1.592  D-01 1.599  D-01 1.606  D-01 EDC06390
0640: 1.613  D-01 1.619  D-01 1.624  D-01 1.630  D-01 1.634  D-01 1.638  D-01 EDC06400
0641: 1.642  D-01 1.645  D-01 1.64   D-01 1.64   D-01 1.64   D-01 1.64   D-01 EDC06410
0642: 1.64   D-01 1.63   D-01 1.62   D-01 1.61   D-01                      EDC06420
0643:                                                                       EDC06430
0644: 1.34267D-05 3.75001D-05 7.1385 D-05 1.9552 D-04 1.703  D-04 2.36   D-04 EDC06440
0645: 3.1    D-04 3.9    D-04 4.0    D-04 4.1    D-04 4.1    D-04 4.2    D-04 EDC06450
0646: 4.4    D-04 4.4    D-04 4.5    D-04 4.6    D-04 4.7    D-04 4.8    D-04 EDC06460
0647: 5.     D-04 5.     D-04 5.     D-04 5.     D-04 5.     D-04 5.     D-04 EDC06470
0648: 5.     D-04 5.     D-04 6.     D-04 6.     D-04                      EDC06480
0649:                                                                       EDC06490
0650: 2.22264D-06 6.35734D-06 1.24086D-05 2.061  D-05 3.123  D-05 4.44   D-05 EDC06500
0651: 6.0    D-05 7.8    D-05 8.0    D-05 8.2    D-05 8.4    D-05 8.6    D-05 EDC06510
0652: 9.     D-05 9.     D-05 9.     D-05 9.     D-05 9.     D-05 1.0    D-04 EDC06520
0653: 1.0    D-04 1.0    D-04 1.0    D-04 1.0    D-04 1.1    D-04 1.1    D-04 EDC06530
0654: 1.1    D-04 1.1    D-04 1.1    D-04 1.1    D-04                      EDC06540
0655:                                                                       EDC06550
0656: 2.18126D-06 6.12048D-06 1.1716 D-05 1.908  D-05 2.83   D-05 3.98   D-05 EDC06560
0657: 5.32   D-05 6.8    D-05 7.0    D-05 7.2    D-05 7.3    D-05 7.5    D-05 EDC06570
0658: 7.7    D-05 7.8    D-05 8.     D-05 8.     D-05 8.     D-05 8.     D-05 EDC06580
0659: 9.     D-05 9.     D-05 9.     D-05 9.     D-05 9.     D-05 1.0    D-04 EDC06590
0660: 1.0    D-04 1.0    D-04 1.0    D-04 1.0    D-04                      EDC06600
```

```
0661:                                                                         EDC06610
0662: 4.05547D-05 1.10601D-04 2.05336D-04 3.238  D-04 4.65   D-04 6.26   D-04 EDC06620
0663: 8.0    D-04 9.9    D-04 9.9    D-04 1.0    D-03 1.0    D-03 1.0    D-03 EDC06630
0664: 1.0    D-03 1.1    D-03 1.1    D-03 1.1    D-03 1.1    D-03 1.1    D-03 EDC06640
0665: 1.2    D-03 1.2    D-03 1.2    D-03 1.2    D-03 1.2    D-03 1.3    D-03 EDC06650
0666: 1.3    D-03 1.3    D-03 1.3    D-03 1.3    D-03                         EDC06660
0667:                                                                         EDC06670
0668: 1.09732D-01 1.15285D-01 1.21462D-01 1.28270D-01 1.35720D-01 1.4378 D-01 EDC06680
0669: 1.5230 D-01 1.609  D-01 1.617  D-01 1.625  D-01 1.633  D-01 1.641  D-01 EDC06690
0670: 1.649  D-01 1.657  D-01 1.664  D-01 1.671  D-01 1.678  D-01 1.686  D-01 EDC06700
0671: 1.691  D-01 1.69   D-01 1.70   D-01 1.71   D-01 1.71   D-01 1.71   D-01 EDC06710
0672: 1.72   D-01 1.72   D-01 1.73   D-01 1.73   D-01                         EDC06720
0673:                                                                         EDC06730
0674: 6.47033D-06 1.80889D-05 3.4526 D-05 5.609  D-05 8.32   D-05 1.159  D-04 EDC06740
0675: 1.54   D-04 1.96   D-04 2.00   D-04 2.05   D-04 2.09   D-04 2.14   D-04 EDC06750
0676: 2.18   D-04 2.23   D-04 2.27   D-04 2.33   D-04 2.36   D-04 2.4    D-04 EDC06760
0677: 2.4    D-04 2.5    D-04 2.5    D-04 2.6    D-04 2.6    D-04 2.7    D-04 EDC06770
0678: 2.7    D-04 2.8    D-04 2.8    D-04 2.8    D-04                         EDC06780
0679:                                                                         EDC06790
0680: 1.44613D-06 4.18645D-06 8.2856 D-06 1.3976 D-05 2.153  D-05 3.12   D-05 EDC06800
0681: 4.32   D-05 5.7    D-05 5.9    D-05 6.0    D-05 6.2    D-05 6.3    D-05 EDC06810
0682: 6.5    D-05 6.6    D-05 6.8    D-05 6.9    D-05 7.1    D-05 7.3    D-05 EDC06820
0683: 7.4    D-05 7.6    D-05 7.7    D-05 7.9    D-05 8.0    D-05 8.3    D-05 EDC06830
0684: 8.4    D-05 8.5    D-05 8.7    D-05 8.9    D-05                         EDC06840
0685:                                                                         EDC06850
0686: 3.91227D-05 1.04493D-04 1.8984 D-04 2.923  D-04 4.08   D-04 5.34   D-04 EDC06860
0687: 6.6    D-04 7.8    D-04 7.9    D-04 8.0    D-04 8.1    D-04 8.2    D-04 EDC06870
0688: 8.3    D-04 8.4    D-04 8.5    D-04 8.7    D-04 9.     D-04 9.     D-04 EDC06880
0689: 9.     D-04 9.     D-04 9.     D-04 9.     D-04 9.     D-04 9.     D-04 EDC06890
0690: 9.     D-04 9.     D-04 9.     D-04 9.     D-04                         EDC06900
0691:                                                                         EDC06910
0692: 1.60225D-01 1.63003D-01 1.66092D-01 1.69496D-01 1.73221D-01 1.7725 D-01 EDC06920
0693: 1.8152 D-01 1.858  D-01 1.862  D-01 1.866  D-01 1.871  D-01 1.875  D-01 EDC06930
0694: 1.878  D-01 1.882  D-01 1.886  D-01 1.890  D-01 1.893  D-01 1.897  D-01 EDC06940
0695: 1.900  D-01 1.90   D-01 1.91   D-01 1.91   D-01 1.91   D-01 1.91   D-01 EDC06950
0696: 1.91   D-01 1.92   D-01 1.92   D-01 1.92   D-01                         EDC06960
0697:                                                                         E4306970
0698: 1.71055D-02 1.76796D-02 1.83158D-02 1.90150D-02 1.97787D-02 2.06039D-02 EDC06980
0699: 2.275  D-02 2.282  D-02 2.289  D-02 2.296  D-02 2.302  D-02 2.309  D-02 EDC07000
0700: 2.1475 D-02 2.2346 D-02 2.243  D-02 2.251  D-02 2.259  D-02 2.267  D-02 EDC06990
0701: 2.315  D-02 2.320  D-02 2.325  D-02 2.329  D-02 2.332  D-02 2.33   D-02 EDC07010
0702: 2.336  D-02 2.34   D-02 2.34   D-02 2.33   D-02                         EDC07020
```

```
DATE = 21/11/78       TIME =  1.25. 5

FILE = SOPATIDE.TABLE    (DG)

C PROGRAM TO PRINT OUT TABLE OF STEADY WAVE PROPERTIES AS APPEARS
C IN CORLETT, 1977.
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 DATA2(29,12)
      DIMENSION DATA(29, 12)
      DO 40 K=1, 10
      JE = 12
      IF (K .EQ. 1) JE=6
      DO 10 J=1, JE
      READ (5, 1001)
      READ(5, 1002) (DATA(I,J), DATA2(I,J),I=2, 29)
      DATA(1,J) = 0.
   10 CONTINUE
      R0 = DFLOAT(K-1)/10.
      DATA(1,2) =( 1.-R0*R0)/(1.+R0*R0)
       DATA(1,7) = DATA(1,2)
      IF (K .EQ. 1) GO TO 12
      D = -DLOG(R0)
      DATA(1,11) = DATA(1,2)/2. + D
      DATA(1,12) = D*(DATA(1,2) + D/2.)
   12 CONTINUE
      WRITE(6, 1003)
      IF (K .EQ. 1) WRITE(6, 1011)
      KKK = K-1
      IF ( K .NE. 1) WRITE(6, 1012) KKK,KKK
      DELEPS = 0.1
      EPS = 0.
      WRITE(6, 1006) EPS, (DATA(1,J),J=1, 6)
      EPS = 0.1
      DO 20 I=2, 29
      WRITE(6, 1004) EPS, (DATA(I,J),DATA2(I,J),J=1, 6)
      IF (I .EQ. 9) DELEPS = 0.01
      EPS = EPS + DELEPS
   20 CONTINUE
      WRITE(6, 1005)
      WRITE(6, 1013) KKK
      DELEPS = 0.1
      EPS = 0.
      WRITE(6, 1006) EPS, (DATA(1,J),J=7,JE)
      EPS = 0.1
      DO 30 I=2, 29
      WRITE(6, 1004) EPS, (DATA(I,J),DATA2(I,J),J=7, JE)
      IF (I .EQ. 9) DELEPS = 0.01
      EPS =EPS + DELEPS
   30 CONTINUE
   40 CONTINUE
      WRITE(6, 1003)
      STOP
 1001 FORMAT(1X)
 1002 FORMAT(6(A8,A4))
 1003 FORMAT(1H1)
 1004 FORMAT(25X, F4.2, 2X, 6(A8,A4,2X))
 1005 FORMAT(1H1)
 1006 FORMAT(25X, F4.2, 1PD13.5,5D14.5)
 1010 FORMAT(1X, 1PD11.5)
 1011 FORMAT(//////26X, 'TABLE A.0.  THE PROPERTIES OF THE STEADY WAVE A
     1S A FUNCTION OF THE EXPANSION'//38X,'PARAMETER',5X,'FOR',5X,
     2  '=0.'///)
 1012 FORMAT(//////26X, 'TABLE A.', I1, '.  THE PROPERTIES OF THE STEADY
```

```
     1 WAVE AS A FUNCTION OF THE EXPANSION'//38X,'PARAMETER',
     2  5X,'FOR', 5X, 'EQ.', I1, '.'///)
1013 FORMAT(//////26X, 'TABLE A.',I1,'.  (CONTINUED)'//////)
      END
```

IV. Other Useful Subroutines

IV.A. Introduction

We include here some subroutines not used by INTVEL or by the steady wave programs but which might be useful to computer users at IOS. The numerical routine INTERP of section II.F.1. goes with SPLINE here. The graphical routines go together with those of section II.F.2. to form one complete package.

IV.B. Numerical Routines

SPLINE (N, IX1, IXN, X, F, DFDX, W)

calculates a cubic spline fit to the function F(X) through the N knots X(I), I = 1, 2,...N.

Inputs:

N	number of data points
IX1	parameter to specify what should be done at the endpoint, X(1), of the spline. if IX1 = 1, $\frac{df}{dx}$ specified at X(1). =-1, $\frac{df}{dx}$ equal at X(1) and X(2). =2, $\frac{d^2f}{dx^2}$ specified at X(1). =-2, $\frac{d^2f}{dx^2}$ equal at X(1) and X(2). =3, $\frac{d^3f}{dx^3}$ specified at X(1). =-3, $\frac{d^3f}{dx^3}$ equal over interval X(1) to X(3).
IXN	parameter to specify what should be done at the endpoint X(N). Similar to IX1.
X	array of abscissa values, dimension N
F	array of ordinate values, dimension N

DFDX (1)	specified first, second or third derivative at X (1) if IX1 = 1, 2 or 3.
DFDX (N)	as above but for X (N)

Outputs:

DFDX	array of $\frac{df}{dx}$ values at each knot, dimension N
W	working array of dimensions 3*N

Subroutines called: none

```
      SUBROUTINE SPLINE(N, IX1, IXN, X, F, DFDX, W)
      IMPLICIT REAL*8 (A-H, O-Z)
C   SUBROUTINE CALCULATES CUBIC SPLINE FIT TO FUNCTION F AT THE
C KNOTS X(I), I=1, N.  IT RETURNS THE VALUES OF THE FIRST DERIVATIVE,
C DFDX, AT THE KNOTS.  W IS A WORK ARRAY OF SIZE 3*N.  IF THE
C DERIVATIVES ARE SPECIFIED THEY ARE INPUT BY DFDX(1) AND DFDX(N).
C IF IX1 = 1, F' SPECIFIED AT X(1).
C         -1, F' EQUAL AT X(1) AND X(2).
C          2, F'' SPECIFIED AT X(1).
C         -2, F'' EQUAL AT X(1) AND X(2).
C          3, F''' SPECIFIED AT X(1).
C         -3, F''' EQUAL OVER FIRST 2 INTERVALS, X(1) TO X(3).
C SAME FOR IXN BUT AT X(N) OR OVER LAST 2 INTERVALS.
      DIMENSION X(N), F(N), DFDX(N), W(1)
      DO 5 I=2, N
      IF (X(I) - X(I-1)) 1, 1, 5
 1    WRITE(6, 1003) I
      STOP
 5    CONTINUE
C SET UP THE SYSTEM OF EQUATIONS.
        H2 = X(2) - X(1)
      NM1 = N-1
      DO 10 I=2, NM1
      H1 = H2
      H2 = X(I+1) - X(I)
      XLAM = H2/(H1 + H2)
      XMU = 1. - XLAM
      W(3*I - 2) = XLAM
      W(3*I - 1) = 2.
      W(3*I) = XMU
      DFDX(I) = 3.*XLAM*(F(I) - F(I-1))/H1 + 3.*XMU*(F(I+1) - F(I))/H2
 10   CONTINUE
C TAKE CARE OF SPECIAL CONDITIONS AT X(1).
      W(1) = 0.
      W(2) = 2.
      JS = 1
      H1 = X(2) - X(1)
      H2 = X(3) - X(2)
      JX1 = IABS(IX1)
      IF (JX1 .EQ. 0) GO TO 15
      IF (JX1 .LE. 3) GO TO 20
 15   WRITE(6, 1001) IX1
      STOP
 20   GO TO (30, 60, 90),JX1
 30   CONTINUE
      IF (IX1) 50, 15, 40
 40   CONTINUE
C F' SPECIFIED AT X(1).
      W(3) = 0.
      DFDX(1) = 2.*DFDX(1)
      GO TO 120
 50   CONTINUE
C F' EQUAL AT X(1) AND X(2).
      W(3) = -2.
      DFDX(1) =0.
      GO TO 120
 60   IF(IX1) 80, 15, 70
 70   CONTINUE
C F'' SPECIFIED AT X(1).
      W(3) = 1.
```

```
      DFDX(1) = -H1*DFDX(1)/2. + 3.*(F(2) - F(1))/H1
      GO TO 120
   80 CONTINUE
C F'' EQUAL AT X(1) AND X(2).
      W(3) = 2.
      DFDX(1) = 4.*(F(2) - F(1))/H1
      GO TO 120
   90 IF (IX1) 110, 15, 100
  100 CONTINUE
C F'''  SPECIFIED AT X(1).
      W(3) = 2.
      DFDX(1) = 4.*(F(2) - F(1))/H1 + (H1**2)*DFDX(1)/3.
      GO TO 120
  110 CONTINUE
C F''' EQUAL OVER FIRST 2 INTERVALS, X(1) TO X(3).
      JS = 2
      W(5) = 2.
      W(6) = 2.*H1/(H1 + H2)
      DFDX(2) = H1*(6.*H2 + 4.*H1)*F(3)/(H2*(H1 + H2)**2)
     1  + 2.*(H2 - 2.*H1)*F(2)/(H1*H2) - 2.*H2*H2*F(1)/(H1*(H1 + H2)**2)
  120 CONTINUE
C TAKE CARE OF SPECIAL CONDITIONS AT X(N)
      W(3*N-1) = 2.
      W(3*N) = 0.
      H1 = X(N-1) - X(N-2)
      H2 = X(N) - X(N-1)
      JF = N
      JXN = IABS(IXN)
      IF (JXN .EQ. 0) GO TO 125
      IF (JXN .LE. 3) GO TO 130
  125 WRITE(6, 1002) IXN
      STOP
  130 GO TO (140, 170, 200),JXN
  140 CONTINUE
      IF (IXN) 160, 125, 150
  150 CONTINUE
C F' SPECIFIED AT X(N).
      W(3*N-2) = 0.
      DFDX(N) = 2.*DFDX(N)
      GO TO 230
  160 CONTINUE
C F' EQUAL AT X(N-1) AND X(N).
      W(3*N-2) = -2.
      DFDX(N) = 0.
      GO TO 230
  170 IF (IXN) 190, 125, 180
  180  CONTINUE
C F'' SPECIFIED AT X(N).
      W(3*N-2) = 1.
      DFDX(N) = H2*DFDX(N)/2. + 3.*(F(N) - F(N-1))/H2
      GO TO 230
  190 CONTINUE
C F'' EQUAL AT X(N-1) AND X(N).
      W(3*N-2) = 2.
      DFDX(N) = 4.*(F(N) - F(N-1))/H2
      GO TO 230
  200 IF (IXN) 220, 125, 210
  210 CONTINUE
C F''' SPECIFIED AT X(N).
      W(3*N-2) = 2.
```

```
      DFDX(N) = 4.*(F(N) - F(N-1))/H2 + DFDX(N)*H2**2/3.
      GO TO 230
  220 CONTINUE
C F''' EQUAL OVER LAST 2 INTERVALS, X(N-2) TO X(N).
      JF = N-1
      W(3*N-4) = 2.
      W(3*N-5) = 2.*H2/(H1 + H2)
      DFDX(N-1) = -H2*(6.*H1 + 4.*H2)*F(N-2)/(H1*(H1+H2)**2)
     1 -2.*(H1 - 2.*H2)*F(N-1)/(H2*H1)
     2 +2.*H1*H1*F(N)/(H2*(H1 + H2)**2)
  230 CONTINUE
C SOLVE THE SYSTEM OF EQUATIONS.
      U = 0.
      Q = 0.
      DO 250 K=JS, JF
      A = W(3*K-2)
      B = W(3*K-1)
      C = W(3*K)
      D = DFDX(K)
      P = A*Q + B
      Q = -C/P
      U = (D - A*U)/P
      W(3*K-2) = P
      W(3*K-1) = Q
      W(3*K) = U
  250 CONTINUE
      DFDX(JF) = W(3*JF)
      JE = JF - 1
      K = JE
      DO 260 J=JS, JE
      DFDX(K) = W(3*K-1)*DFDX(K+1) + W(3*K)
      K = K - 1
  260 CONTINUE
      IF (JS .EQ. 1) GO TO 270
      H1 = X(2) - X(1)
      H2 = X(3) - X(2)
      DFDX(1) = (H1*H1 - H2*H2)*DFDX(2)/(H2*H2) + H1*H1*DFDX(3)/(H2*H2)
     1 - 2.*F(1)/H1 + 2.*(H2**3 + H1**3)*F(2)/(H1*H2**3)
     2 - 2.*H1*H1*F(3)/(H2**3)
  270 CONTINUE
      IF (JF .EQ. N) GO TO 280
      H1 = X(N-1) - X(N-2)
      H2 = X(N) - X(N-1)
      DFDX(N) = H2*H2*DFDX(N-2)/(H1*H1) + (H2*H2 - H1*H1)*DFDX(N-1)
     1 /(H1*H1) + 2.*H2*H2*F(N-2)/(H1**3)
     2  - 2.*(H2**3 + H1**3)*F(N-1)/(H1**3*H2)
     3 +2.*F(N)/H2
  280 RETURN
 1001 FORMAT(//' ERROR IN SUBROUTINE SPLINE.'//' PARAMETER IX1 = ',I5,
     1 ' WHICH IS OUT OF PERMITTED RANGE.')
 1002 FORMAT (//' ERROR IN SUBROUTINE SPLINE.'//' PARAMETER IXN = ',I5,
     2 ' WHICH IS OUT OF PERMITTED RANGE. )
 1003 FORMAT(//' ERROR IN SUBROUTINE SPLINE.'//' X(', I5,
     1 ') OUT OF ORDER.')
      END
```

IV.C. Graphical Routine

ADDPNT (X, Y, IPEN, ICHAR, CHRSIZ, ILINE)

adds a point to a graph previously drawn by BOXPLT.

Inputs:

X	abscissa value
Y	ordinate value
IPEN	IOS pen number
ICHAR	IOS code for character to be drawn at (X,Y)
CHRSIZ	character size in mm
ILINE	line drawing parameter. If =1, draw a line to the last point added by ADDPNT. =0, no line is drawn.

Outputs:

A graphical data point, possibly connected to another by a straight line.

Subroutines called:

IOS Grafix library: LINSEL, NIBSEL, CHASEL, MARSEL, POILD2

```
DATE = 20/11/78      TIME =  1.25. 8

FILE = SOHATLDR.ADDPNT  (DG)

      SUBROUTINE ADDPNT(X, Y, IPEN, ICHAR, CHRSIZ, ILINE)
C   THIS SUBROUTINE ADDS A POINT AND CONNECTS IT WITH A STRAIGHT LINE
C (ILINE=1) OR NO LINE (ILINE=0) TO THE LAST POINT ADDED BY ADDPNT
C ON AXES PREVIOUSLY DRAWN BY BOXPLT.
C IPEN IDENTIFIES THE PEN AS EITHER 1, 2 OR 3.
C ICHAR IS THE CHARACTER CODE.
C CHRSIZ IS THE CHARACTER SIZE IN MM.
      IMPLICIT REAL*8(A-H,O-Z)
      CALL LINSEL(ILINE)
      CALL NIBSEL(IPEN, 2)
      CALL NIBSEL(IPEN, 10)
      CALL CHASEL(SNGL(CHRSIZ))
      CALL MARSEL(ICHAR)
       CALL POILD2(X, Y)
      RETURN
      END
```

PERADD (N, X, Y, XP, IPEN, ICHAR, CHRSIZ, ILINE)

plots the N data points (X, Y) on axes already drawn by PERPLT. Y is periodic in X with period XPER. XMIN and XPER are supplied by PERPLT through a COMMON statement.

Inputs:

N	number of points to be plotted
X	array of abscissa values, dimension N
Y	array of ordinate values, dimension N. Y(1) = Y(N) by assumption
XP	working array, dimension N
IPEN	IOS pen number for line to be drawn
ICHAR	IOS character code for character to be output at each point (X,Y)
CHRSIZ	character size in mm
ILINE	line drawing parameter. if =0, no line is drawn. If =1, a straight line is drawn between points

Outputs:

XP	array of X values (dimension N) but with $\mathrm{XMIN} \leqslant X \leqslant \mathrm{XMIN} + \mathrm{XPER}$

A graph of Y versus X.

Subroutines called:

EDC library: ADDPNT

```
DATE = 20/11/78      TIME =  1.25. 8

FILE = SOMAILDE.PERADD  (DG)

      SUBROUTINE PERADD(N, X, Y, XP, IPEN, ICHAR, CHRSIZ, ILINE)
C   THIS SUBROUTINE PLOTS THE PERIODIC ARRAY Y VERSUS THE ARRAY X EACH
C OF SIZE N ON AXES ALREADY DRAWN BY PERPLT.
C Y IS ASSUMED TO BE PERIODIC OF PERIOD XPER.
C (XMIN AND XPER ARE SUPPLIED BY PERPLT THROUGH A COMMON STATEMENT.)
C THE X AXIS IS DRAWN FROM XMIN TO (XMIN + XPER).
C XP IS A WORK ARRAY OF LENGTH
C N AND RETURNS THE VALUES OF X BUT ADJUSTED TO BE BETWEEN XMIN AND
C (XMIN + XPER).
C
C N = NUMBER OF POINTS TO BE PLOTTED.  (N MAY BE 0.)
C X, Y ARE ARRAYS PLOTTED.
C XP IS ARRAY X BUT NORMALIZED TO BE BETWEEN XMIN AND (XMIN + XPER).
C IPEN = 1, 2 OR 3.
C ICHAR IS IOS CHARACTER CODE.
C CHRSIZ IS CHARACTER SIZE IN MM.
C ILINE = 0, NO LINE IS DRAWN BETWEEN POINTS.
C        = 1, A STRAIGHT LINE IS DRAWN.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /PERAD/XMIN,XPER
      DIMENSION X(1), Y(1), XP(1)
      IF (N .LE. 0) RETURN
C
C REDEFINE THE X(I) TO BE BETWEEN XMIN AND (XMIN + XPER).
C
      DO 10 I=1, N
      XP(I) = X(I)
 10   CONTINUE
      DO 50 I=1, N
 12   IF (XP(I) .GE. XMIN) GO TO 44
      XP(I) = XP(I) + XPER
      GO TO 42
 14   IF (XP(I) .LE. (XMIN + XPER)) GO TO 50
      XP(I) = XP(I) -XPER
      GO TO 44
 50   CONTINUE
C
C PLOT Y VERSUS X.
C
      CALL ADDPNT(XP(N), Y(N), IPEN, 0, CHRSIZ, 0)
      DO 60 J=1, N
      JM1 = J - 1
      IF(JM1 .EQ. 0) JM1 = N
      IF((XP(J) - XP(JM1)) .GT. -XPER/2.) GO TO 52
      XX = XP(J) + XPER
      CALL ADDPNT(XX, Y(J), IPEN, ICHAR, CHRSIZ, ILINE)
      XX = XP(JM1) - XPER
      CALL ADDPNT(XX, Y(JM1), IPEN, ICHAR, CHRSIZ, 0)
      GO TO 54
 52   IF ((XP(J) - XP(JM1)) .LT. XPER/2.) GO TO 54
      XX = XP(J) - XPER
      CALL ADDPNT(XX, Y(J), IPEN, ICHAR, CHRSIZ, ILINE)
      XX = XP(JM1) + XPER
      CALL ADDPNT(XX, Y(JM1), IPEN, ICHAR, CHRSIZ, 0)
 54   CONTINUE
      CALL ADDPNT(XP(J), Y(J), IPEN, ICHAR, CHRSIZ, ILINE)
 60   CONTINUE
      RETURN
      END
```

PERPLT (N, X, Y, XP, IPEN, ICHAR, CHRSIZ, ILINE, ITOP, IBOT, ILEFT, IRIGHT, XMIN, XPER, XINC, XLEN, YMIN, YMAX, YINC, YLEN)

Plots the periodic array Y as a function of X. It draws all or part of a box with or without labels and plots the N points Y versus X which fall within the box. The pen number, plot character type and size can be altered.

Inputs:

N	number of points to be plotted
X	array of abscissa values, dimension N
Y	array of ordinate values, dimension N. Y(1) = Y(N) by assumption
XP	working array, dimension N
IPEN	IOS pen number for line to be drawn
ICHAR	IOS character code for character to be output at each point (X,Y)
CHRSIZ	character size in mm
ILINE	line drawing parameter If = 0 no line between points If = 1 straight line
ITOP	controls drawing of top of box = 0, no axis drawn, no labels = 1, axis, no labels = 2, axis, labels checked for overlaps = 3, axis, labels with no overlap checking

IBOT	as for ITOP, but bottom of box
ILEFT	as for ITOP, but left side of box
IRIGHT	as for ITOP, but right side of box
XMIN	minimum value of X
XPER	period of X. Graph will be drawn such that XMIN $\leq$ X $\leq$ XMIN + XPER
XINC	increments of X-axis
XLEN	length of X-axis in mm
YMIN	minimum value of y-axis
YMAX	maximum value of y-axis
YINC	increments of y-axis
YLEN	length of y-axis in mm

Outputs:

XP	array of X values but with XMIN $\leq$ X $\leq$ XMIN + XPER

A graph of Y versus X within a box.

Subroutines called:

EDC library: BOXPLT, PERADD

```
DATE = 29/11/78      TIME =  1.25. 8

FILE = SOMATIDOF.PERPLT  (DG)

      SUBROUTINE PERPLT(N, X, Y, XP, IPEN, ICHAR, CHRSIZ, ILINE,
     1  ITOP, IBOT, ILEFT, IRIGHT, XMIN, XPER, XINC, XLEN, YMIN,
     2  YMAX, YINC, YLEN)
C   THIS SUBROUTINE PLOTS THE PERIODIC ARRAY Y VERSUS THE ARRAY X EACH
C OF SIZE N.  IT DRAWS ALL OR PART OF A BOX WITH OR WITHOUT LABELS AND
C PLOTS THE N POINTS OF Y VERSUS X WHICH FALL WITHIN THE BOX.  THE PEN
C NUMBER, PLOT CHARACTER TYPE AND SIZE CAN BE ALTERED.
C Y IS ASSUMED TO BE PERIODIC OF PERIOD XPER.
C THE X AXIS IS DRAWN FROM XMIN TO (XMIN + XPER) IN INCREMENTS OF
C XINC.  THE Y AXIS IS FROM YMIN TO YMAX WITH INCREMENTS OF YINC.  THE
C AXES ARE OF LENGTH XLEN AND YLEN MM.  XP IS A WORK ARRAY OF LENGTH
C N AND RETURNS THE VALUES OF X BUT ADJUSTED TO BE BETWEEN XMIN AND
C (XMIN + XPER).
C
C N = NUMBER OF POINTS TO BE PLOTTED.  (N MAY BE 0.)
C X, Y ARE ARRAYS PLOTTED.
C XP IS ARRAY X BUT NORMALIZED TO BE BETWEEN XMIN AND (XMIN + XPER).
C IPEN = 1, 2 OR 3.
C ICHAR IS IOS CHARACTER CODE.
C CHRSIZ IS CHARACTER SIZE IN MM.
C ILINE = 0, NO LINE IS DRAWN BETWEEN POINTS.
C       = 1, A STRAIGHT LINE IS DRAWN.
C ITOP, IBOT, ILEFT AND IRIGHT CONTROL AXIS DRAWING AND LABELING.
C I--- = 0, NO AXIS DRAWN, NO LABELS.
C      = 1, AXIS DRAWN, NO LABELS.
C      = 2, AXIS DRAWN, LABELS BUT CHECKING DONE FOR OVERLAPS.
C      = 3, AXIS DRAWN, LABELS WITH NO OVERLAP CHECKING.
C XMIN AND (XMIN + XPER) = USER LIMITS OF X AXIS.
C YMIN AND YMAX =                          Y     .
C XINC AND YINC = INCREMENTS BETWEEN TICK MARKS.
C XLEN AND YLEN = AXIS LENGTHS IN MM.
C
      IMPLICIT REAL*8(A-H, O-Z)
      COMMON /PERAD/XMINN,XPERR
      DIMENSION X(1), Y(1), XP(1)
      XMAX = XMIN + XPER
      CALL BOXPLT(0, 0.D0, 0.D0, IPEN, 0, 0.D0, 0, ITOP, IBOT, ILEFT,
     1  IRIGHT, XMIN, XMAX, XINC, XLEN, YMIN, YMAX, YINC, YLEN)
      XMINN = XMIN
      XPERR = XPER
      CALL PERADD(N, X, Y, XP, IPEN, ICHAR, CHRSIZ, ILINE)
      RETURN
      END
```

Acknowledgements

The research on which the computer programs in this manual are based was first begun while I was a research student under the supervision of Professor Michael S. Longuet-Higgins in the Department of Applied Mathematics and Theoretical Physics, Cambridge. From November 1975 until November 1978 this research was funded at IOS by the U.K. Departments of Energy and Industry. I should like to thank my colleagues at IOS, especially Michael Longuet-Higgins and James Crease, for their help and encouragement. This manuscript has been prepared while I have been an oceanographer at the Pacific Marine Environmental Laboratory, Seattle, Washington, U.S.A.