ANALYSIS OF SHIP FLOW IN AN IDEAL FLUID USING GUILLOTON'S METHOD AND SPLINE FUNCTIONS

JIUNN-MING CHUANG

ANALYSIS OF SHIP FLOW IN AN IDEAL FLUID USING GUILLOTON'S METHOD AND SPLINE FUNCTIONS

BY

© Jiunn-Ming Chuang, M.Sc.(Eng.)

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

January, 1984

St. John's Newfoundland

ABSTRACT

In this thesis, the numerical evaluation is formulated for computing the linearized disturbance velocity of a steady, inviscid free surface gravity flow past a ship hull. The hull is represented by a system of source panels with uniformly distributed strengths on the centerplane. The imporvement of the results on the boundaries, i.e. free surface and hull surface, by Guilloton's method is investigated. Based on Guilloton's method, thin-ship-panel approximation and cubic spline curve fitting, a scheme has been developed for setting up a computer program to compute the ship wave-making resistance, flow around the ship hull and wave elevation along the ship side. The results of sample calculations for standard hull forms of Wigley model 3012 and Series 60 block 60 have shown good agreement with the experimental results for Froude numbers between 0.25 and 0.35 which are just in the speed range of the conventional merchant ships.

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to Professor G. C. Hsiung for his valuable guidance during the course of studies and Professor Don Bass for his suggestions and comments. Sincere thanks are also due to Dr. G. R. Peters, Dean of Engineering and Applied Science, and Dr. T. R. Chari, Associate Dean of Engineering and Applied Science, for their encouragement. The author is also grateful to Mr. Weizhen Ge and Mr. C. C. Tse for devoting much of their time in discussions and especially to his wife, Bih-Chyun, for her continuous encouragement throughout this work. Acknowledgement is also made for University Fellowship by Memorial University of Newfoundland and Financial Support by NSERC Operating Grant on Computational Ship Hydrodynamics.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

Linearized space (LS)	=	The domain (x_o, y_o, z_o) of the linearized solution.
Real space (RS)	=	The domain (x, y, z) of the Guilloton's solution.
$f(x,z)$, $f^o(x_o,z_o)$	=	Ship hull functions in RS and LS, respectively.
$\zeta(x,y)$, $\zeta^o(x_o,y_o)$	=	Wave elevations in RS and LS respectively.
R, R_o	=	Wave-making resistances in RS and LS, respectively.
g	=	Gravitational acceleration.
$\vec{q}(x,y,z)$	=	(u,v,w), Disturbance velocity vector.
ε	=	Perturbation parameter.
$G(x,y,z;x',0,z')$	=	Green's function of linearized problem.
Re	=	Real part of complex function.
Im	=	Imaginary part of complex function.
C_1, C_2	=	Integrating paths in complex plane.
$m(x',z')$	=	Strength of source distribution on the centerplane.
u_{ij}, v_{ij}, w_{ij}	=	Element velocity components induced by a source panel (i,j) on the centerplane.
R_{ij}	=	Element wave-making resistance induced by a source panel (i,j) on the centerplane.
ζ_{ij}	=	Element wave-elevation induced by a source panel (i,j) on the centerplane.

I_1, I_2, I_S, I_R = Complex integrals

Res(a) = Residue of complex function, f(k), at pole k = a

sig ω = -1 for ω < 0, 1 for ω > 0

H(x − x′) = Heaviside "unit step function", 0 for x < x′, 1 for x > x′

$$\{f(x',z')\}\Big|_{x'_i}^{x'_{i+1}}\Big|_{z'_j}^{z'_{j+1}} = \{f(x',z')\}_{\substack{x'=x'_{i+1}\\ z'=z'_{j+1}}} - \{f(x',z')\}_{\substack{x'=x'_{i+1}\\ z'=z'_j}} - \{f(x',z')\}_{\substack{x'=x'_i\\ z'=z'_{j+1}}} + \{f(x',z')\}_{\substack{x'=x'_i\\ z'=z'_j}}$$

$E_1(\lambda)$ = $\int_{\lambda}^{\infty} \frac{e^{-K}}{K}\, dK$, (λ is a complex number), or called complex exponential integral.

η or ETA = Nondimensional wave elevation.

$A(x_o,z_o), B(x_o,z_o), C(x_o,z_o)$ = Displacements of Guilloton's transformation in x_o-, y_o- and z_o- directions, respectively.

F.P. = Forward perpendicular of a ship (-1)

A.P. = After perpendicular of a ship (1)

C_w = $R_w/(\frac{1}{2}\rho S U^2)$, Wave resistance coefficient

L = 1/2 LBP

CHAPTER 1

INTRODUCTION

Since Michell [1] developed the thin ship theory to solve the linearized problem of the waves produced by a ship of given form moving with uniform velocity in the free surface of unbounded water which is considered to be inviscid, many researchers have tried to modify the thin ship theory to obtain more reasonable results by including the nonlinear effects. From a practical point of view, one of the notable methods was developed by Guilloton [2] based on geometrical and intuitive physical reasoning and was formulated in mathematical form by Gadd [10]. As a matter of fact, Guilloton's basic idea is the same as that of the well-known "strained coordinates method" which was developed by Poincaré and successfully used in some singular perturbation problems [3]. The main idea of this kind of method is that the linearized solution of the nonlinear problem may have the right form, but not quite at the right place, so that the remedy is to slightly strain the coordinates or set up a transformation between the "linearized space" and the "real space". Following the method of strained coordinates, a perturbation analysis can be carried out to rationalize Guilloton's method, such as in [4] and [5]. They have shown that Guilloton's solution is essentially equivalent to an inconsistent second order approximation, in which the field equation is satisfied to first order and the boundary conditions are satisfied to second order.

From the computational point of view, Guilloton's method includes the following three sub-procedures:

1. Find the "linearized hull" corresponding to a given "real hull" for a given Froude number by an "inverse Guilloton's transformation". This is an iterative process.

2. Calculate the flow quantities around the "linearized hull" by the thin ship theory.

3. Transform the calculated flow quantities to that around the "real hull" by Guilloton's transformation.

Based on these sub-procedures, a computer program has been developed to analyse the flow and isobar around the ship hull, the wave elevation along the ship side and the ship wave-making resistance. In mathematical principle, Guilloton's solution is still only of first-order accuracy. But from the results of numerical experiments, the prediction of the ship flow with Guilloton's method is much better than that with the thin ship theory for Froude numbers of about 0.25 to 0.35. The use of Guilloton's method for preliminary design of conventional ships has a good potential.

CHAPTER 2

THEORETICAL BACKGROUND AND NUMERICAL FORMULATION

2.1 Exact and linearized steady ship-flow problem

It is assumed that the fluid is incompressible and inviscid and that the flow is irrotational. Let Oxyz be a moving coordinate system fixed on the ship, with velocity $\vec{U} = (-U,0,0)$ with respect to the obvious inertial frame and the origin O is at midship. The Oxy plane is taken to coincide with the undisturbed free surface and the z-axis is positive upwards as in Figure 1. The effects of sinkage and trim are not considered. The hull equation is $y = \pm f(x,z)$ for $-L \leq x \leq L$, $-b(x) \leq z \leq \zeta(x,y)$, where 2L is the length of the ship, $z = \zeta(x,y)$ denotes the elevation of the disturbed free surface and $z = -b(x)$ is the equation of the keel line. Now, the problem of evaluating the ship flow is reduced to determine the disturbance velocity potential, say $\phi(x,y,z)$, which satisfies Laplace's equation, $\nabla^2 \phi = 0$, in the flow field with the following exact boundary conditions as given in [6]:

(a) The kinematic boundary condition on the hull surface:

$$\phi_x(x,\pm f(x,z),z)\, f_x(x,z) \mp \phi_y + \phi_z f_z = -Uf_x \qquad (1.1)$$

(b) The kinematic boundary condition on the free surface:

$$\phi_x(x,y,\zeta(x,y))\,\zeta_x(x,y) + \phi_y \zeta_y - \phi_z = -U\zeta_x \qquad (1.2)$$

(c) The dynamic boundary condition on the free surface:

$$U\phi_x(x,y,\zeta(x,y)) - 1/2(\phi_x^2 + \phi_y^2 + \phi_z^2) + g\,\zeta(x,y) = 0 \qquad (1.3)$$

(d) The kinematic boundary condition on the ocean bottom, (it is assumed to be infinite):

$$\phi_z(x,y,z) = 0 \quad \text{as} \quad z \longrightarrow -\infty \tag{1.4}$$

(e) The radiation condition specifying that waves are not propagated upstream from the ship but only downstream.

If the disturbance velocity potential ϕ can be found, then the disturbance velocity, $\vec{q}\,(x,y,z)$, the wave-making resistance, R, and the wave elevation, $\zeta(x,y)$, can be calculated as follows:

$$\vec{q}\,(x,y,z) = \nabla\,\phi(x,y,z) \tag{1.5}$$

$$R = \iint_{hull} p\,\cos(n,x)\,ds = 2\iint_{S_o} p(x,f(x,z),z)\,f_x(x,z)\,dxdz$$

$$= -2\rho\iint_{S_o} [U\phi_x(x,f(x,z),z) + 1/2(\phi_x^2 + \phi_y^2 + \phi_z^2) + gz]\,f_x(x,z)\,dxdz \tag{1.6}$$

and $$\zeta(x,y) = -\frac{1}{g}\,[U\phi_x(x,y,\zeta(x,y)) + 1/2(\phi_x^2 + \phi_y^2 + \phi_z^2)] \tag{1.7}$$

where S_o is the projection of the wetted surface on the centerplane and p is the pressure on the ship hull.

The formula for the wave elevation is an implicit form since the right hand side of (1.7) is also a function of wave elevation.

The difficulty of this "exact" problem stems from the fact that the position of the free surface and the extent of the wetted area of the hull surface are initially unknown and are to be determined as part of the solution; also the boundary conditions are non-linear. One of the procedures for linearizing the problem begins by writing the equation of the ship hull in the form $y = \pm\varepsilon f(x,z)$ where ε is a beam-length ratio. It is assumed that the disturbance velocity potential, wave elevation and wave-making resistance can be expanded in power series of ε as follows:

$$\phi(x,y,z;\varepsilon) = \varepsilon\phi^{(1)}(x,y,z) + \varepsilon^2\phi^{(2)} + \ldots\ldots$$

$$\zeta(x,y;\varepsilon) = \varepsilon\zeta^{(1)}(x,y,z) + \varepsilon^2\zeta^{(2)} + \ldots\ldots \qquad (1.8)$$

$$R(\varepsilon) = \varepsilon^2 R^{(1)} + \varepsilon^3 R^{(2)} + \ldots\ldots$$

The expansions (1.8) are now substituted into (1.1) to (1.4). After they have been expanded as a power series in ε, it is found that the velocity potential ϕ of the first order approximation must satisfy Laplace's equation together with the following linearized boundary conditions:

(a) $\phi_y(x,\pm 0,z) = \pm U f_x(x,z)$ (1.9)

(b) $\phi_z(x,y,0) = U\zeta_x(x,y)$ (1.10)

(c) $\zeta(x,y) = -\frac{U}{g}\phi_x(x,y,0)$ (1.11)

(d) $\phi_z(x,y,z) = 0$ as $z \longrightarrow -\infty$ (1.12)

(e) The radiation condition mentioned before.

The wave-making resistance is changed from (1.6) to the following form

$$R = -2\rho U \iint_{S_o} \phi_x(x,0,z) f_x(x,z)\, dxdz \qquad (1.13)$$

where S_o is the ship centerplane for $-L \leq x \leq L$ and $-b(x) \leq z \leq 0$. The free-surface boundary conditions (1.10) and (1.11), are combined to give

$$\phi_{xx}(x,y,0) + K_o\, \phi_z = 0 \quad \text{where} \quad K_o = \frac{g}{U^2} \qquad (1.14)$$

2.2 Solution of linearized ship-flow problem

The linearized ship flow problem is a mathematical boundary-value problem which can be solved by the Green's function method [7], i.e. representing the body by a distribution of singularities. The linearized disturbance velocity potential can be expressed in the form:

$$\phi(x,y,z) = \iint_{S_o} 2Uf_x(x',z') \cdot G(x,y,z;x',0,z')dx'dz' \qquad (2.1)$$

where

$G(x,y,z;x',0,z')$ is Green's function or the unit source function of the linearized problem.

$f_{x'}(x',z')$ is the longitudinal slope of ship hull. The "prime" system also denotes coordinates on the body.

In this linearized solution, equation (2.1) shows that the sources are distributed on the ship centerplane and the source strength is only dependent on the longitudinal slope of the ship hull. We define the source strength $m(x',z') = 2Uf_{x'}(x',z')$ and equation (2.1) becomes

$$\phi(x,y,z) = \iint_{S_o} m(x',z')\, G(x,y,z;x',0,z')\, dx'dz' \qquad (2.2)$$

Green's function of the linearized problem developed by Havelock [8] with the image method is given in the following form

$$4\pi G(x,y,z;x',0,z') = -\frac{1}{r_1} + \frac{1}{r_2}$$

$$+ \frac{K_o}{\pi} \operatorname{Re}\left\{ \int_{-\pi}^{\pi} \sec^2\theta\, d\theta \int_o^\infty \frac{e^{K[(z+z')+i\omega]}}{K-K_o\sec^2\theta}\, dK \right\} \qquad (2.3)$$

where

$$r_1 = [(x-x')^2 + y^2 + (z-z')^2]^{1/2}$$

$$r_2 = [(x-x')^2 + y^2 + (z+z')^2]^{1/2}$$

$$\omega = (x-x')\cos\theta + y\sin\theta$$

and $K_o = g/U^2$

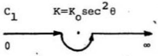
The double integral of Green's function cannot be determined without a statement on the procedure of integration around the singularity, $K = K_o \sec^2\theta$. That is to say that a way (as in [9]) should be chosen such that the free waves only trail behind the ship. The radiation condition mentioned above makes the solution to this mathematical problem unique.

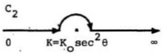
Let

$$I = R_e \left\{ \int_{-\pi}^{\pi} \sec^2 \theta d\theta \int_{o}^{\infty} \frac{e^{K[(z+z')+i\omega]}}{K-K_o \sec^2\theta} dK \right\}$$

$$= R_e \left\{ \int_{-\pi/2}^{\pi/2} \sec^2\theta d\theta \left[\int_{\substack{o \\ (C_1)}}^{\infty} \frac{e^{K[(z+z')+i\omega]}}{K-K_o \sec^2\theta} dK + \int_{\substack{o \\ (C_2)}}^{\infty} \frac{e^{K[(z+z')-i\omega]}}{K-K_o \sec^2\theta} dK \right] \right\} \tag{2.4}$$

The integration paths, C_1 and C_2, have to be chosen as follows:





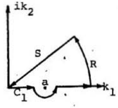
we define

$$I_1 = \int_{C_1} \frac{e^{K[(z+z')+i\omega]}}{K-K_o \sec^2\theta} dK \quad \text{and} \quad I_2 = \int_{C_2} \frac{e^{K[(z+z')-i\omega]}}{K-K_o \sec^2\theta} dK \tag{2.5}$$

I_1 and I_2 can be treated by contour integration in the complex plane to obtain a nonoscillatory integrand.

(1) When $\omega > 0$

(a) Integral I_1



$a = K_o \sec^2\theta$

$$I_1 = 2\pi i \text{ Res}(a) - I_S \qquad (\lim_{R\to\infty} I_R = 0) \qquad (2.6)$$

$$2\pi i\text{Res}(a) = 2\pi\{e^{K_o\sec^2\theta(z+z')}[-\sin(K_o\sec^2\theta\cdot\omega)+i\cos(K_o\sec^2\theta.\omega)]\} \qquad (2.7)$$

$$I_S = \int_S \frac{e^{K[(z+z') + i\omega]}}{K-K_o\sec^2\theta}\, dK \qquad (2.8)$$

The path S can be chosen so as to make the argument of the exponential real along the path and hence eliminating the oscillatory behaviour of the integrand in I_S. So

$$\text{Im}\,\{K[(z+z') + i\omega]\} = 0 \quad \text{and} \quad K = k_1 + ik_2 \qquad (2.9)$$

$$k_2 = -\omega k_1/(z+z') \,, \quad K = k_1\,[(z+z')-i\omega]/(z+z') \qquad (2.10)$$

Substituting (2.10) into (2.8)

$$I_S = \int_{\infty}^{0} \frac{e^{[(z+z')^2+\omega^2]\, k_1/(z+z')}}{[(z+z')^2+\omega^2]\; k_1/(z+z') - K_o \sec^2\theta[(z+z')+i\omega]} \cdot$$

$$\frac{[(z+z')^2+\omega^2]}{(z+z')} \, dK_1 \qquad (2.11)$$

since $z+z' < 0$

changing variable let $-K = [(z+z')^2 + \omega^2]\, k_1/(z+z')$

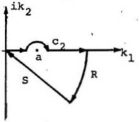
$$I_S = -\int_0^{\infty} \frac{e^{-K}}{[K+K_o \sec^2\theta \cdot (z+z')]+i(K_o \sec^2\theta.\omega)}\, dK \qquad (2.12)$$

Substituting (2.7) and (2.12) into (2.6) and taking the real part.

$$\mathrm{Re}\{I_1\} = \int_0^{\infty} \frac{[K+K_o \sec^2\theta \cdot (z+z')]e^{-K}}{[K+K_o \sec^2\theta \cdot (z+z')]^2+(K_o \sec^2\theta.\omega)^2} dK$$

$$- 2\pi\, e^{K_o \sec^2\theta \cdot (z+z')} \cdot \sin(K_o \sec^2\theta.\omega) \qquad (2.13)$$

(b) Integral I_2



$a = K_o \sec^2\theta$

$$I_2 = -2\pi i \text{ Res}(a) - I_S \qquad (\lim_{R\to\infty} I_R = 0) \qquad (2.14)$$

$$-2\pi i\text{Res}(a) = 2\pi\{e^{K_o\sec^2\theta.(z+z')}[-\sin(K_o\sec^2\theta.\omega)-i\cos(K_o\sec^2\theta.\omega)]\}$$

(2.15)

$$I_S = \int_S \frac{e^{K[(z+z')-i\omega]}}{K-K_o\sec^2\theta}\,dK \qquad (2.16)$$

Similarly,

$$I_m\{K[(z+z')-i\omega]\} = 0 \quad \text{and} \quad K = k_1 + ik_2 \qquad (2.17)$$

$$k_2 = \omega k_1/(z+z') \quad , \quad K = k_1\ [(z+z')+ i\omega]/(z+z') \qquad (2.18)$$

Substituting (2.18) into (2.16).

$$I_S = \int_\infty^o \frac{e^{[(z+z')^2+\omega^2]k_1/(z+z')}}{[(z+z')^2+\omega^2]k_1/(z+z') - K_o\sec^2\theta[(z+z')-i\omega]}\cdot$$

$$\frac{[(z+z')^2+\omega^2]}{(z+z')}\,dk_1 \qquad (2.19)$$

since $z+z' < 0$

changing variable let $-K = [(z+z')^2+\omega^2]k_1/(z+z')$

$$I_S = -\int_o^\infty \frac{e^{-K}}{[K+K_o\sec^2\theta.(z+z')]-i(K_o\sec^2\theta\cdot\omega)}\,dK \qquad (2.20)$$

Substituting (2.15) and (2.20) into (2.14) and taking the real part

$$\text{Re}\ \{I_2\} = \int_0^\infty \frac{[K+K_o\sec^2\theta\cdot(z+z')]e^{-K}}{[K+K_o\sec^2\theta.(z+z')]^2+(K_o\sec^2\theta\cdot\omega)^2}\,dK$$

$$-\ 2\pi\ .\ e^{K_o\sec^2\theta.(z+z')}\ .\ \sin(K_o\sec^2\theta\cdot\omega) \qquad (2.21)$$

Substituting (2.13) and (2.21) into (2.4), the integral I for $\omega > 0$ is

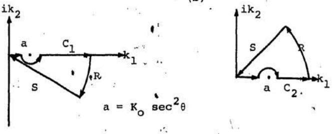
$$I = 2\int_{-\pi/2}^{\pi/2}\sec^2\theta d\theta\int_0^\infty \frac{[K+K_o\sec^2\theta.(z+z')]e^{-K}}{[K+K_o\sec^2\theta.(z+z')]^2+(K_o\sec^2\theta.\omega)^2}\,dK$$

$$-\ 4\pi\int_{-\pi/2}^{\pi/2}\sec^2\theta\cdot e^{K_o\sec^2\theta\cdot(z+z')}\ .\ \sin(K_o\sec^2\theta\cdot\omega)d\theta \qquad (2.22)$$

(2) When $\omega < 0$

Similarly, the contour integrations are chosen as

(a) (b)



The following result for $\omega < 0$ can be obtained

$$I = 2 \int_{-\pi/2}^{\pi/2} \sec^2\theta d\theta \int_0^\infty \frac{[K+K_o \sec^2\theta.(z+z')]e^{-K}}{[K+K_o\sec^2\theta.(z+z')]^2+(K_o\sec^2\theta\cdot\omega)^2} dK$$

(2.23)

Substituting (2.22) and (2.23) into (2.3). Green's function becomes

$$4\pi G(x,y,z;x',0,z') = -\frac{1}{r_1} + \frac{1}{r_2}$$

$$+ \frac{2K_o}{\pi} \int_{-\pi/2}^{\pi/2} \sec^2\theta d\theta \int_0^\infty \frac{[K+K_o\sec^2\theta.(z+z')]e^{-K}}{[K+K_o\sec^2\theta.(z+z')]^2+(K_o\sec^2\theta\cdot\omega)^2} dK$$

$$-4K_o \int_{-\pi/2}^{\pi/2} (\frac{1+\text{sig}\omega}{2})\sec^2\theta\cdot e^{K_o\sec^2\theta.(z+z')} \sin(K_o\sec^2\theta\cdot\omega)d\theta$$

(2.24)

where

$$\text{sig}\ \omega = \begin{cases} -1 & \text{for } \omega < 0 \\ 1 & \text{for } \omega > 0 \end{cases}$$

Since most of the quantities of practical interests, such as wave-making resistance, wave elevation along the ship hull and flow around the ship hull can be obtained from flow variables evaluated at the centerplane, the linearized disturbance velocity potential on the centerplane, $\phi(x,0,z)$, associated with a given source distribution $m(x',z')$, is the most important quantity to be defined in the computational procedure:

$$\phi(x,0,z) = \iint_{S_o} m(x',z') \cdot G(x,0,z;x',0,z')\, dx'dz' \qquad (2.25)$$

with

$$4\pi G(x,0,z;x',0,z') = -\frac{1}{r_1} + \frac{1}{r_2}$$

$$+ \frac{4K_o}{\pi} \int_0^{\pi/2} \sec^2\theta d\theta \int_0^{\infty} \frac{[K+K_o\sec^2\theta.(z+z')]e^{-K}}{[K+K_o\sec^2\theta.(z+z')]^2+(K_o\sec^2\theta\cdot\omega)^2}\, dK$$

$$-H(x-x')\, 8K_o \int_0^{\pi/2} \sec^2\theta \cdot e^{K_o\sec^2\theta.(z+z')} \cdot \sin(K_o\sec^2\theta\cdot\omega)d\theta$$

(2.26)

where

$$r_1 = [(x-x')^2+(z-z')^2]^{1/2}$$

$$r_2 = [(x-x')^2+(z+z')^2]^{1/2}$$

$$\omega = (x-x')\cos\theta$$

and $H(x-x')$ is the Heaviside "unit step function" which is defined as

$$H(x-x') = \begin{cases} 0 & \text{for } x < x' \\ 1 & \text{for } x > x' \end{cases}$$

Substituting (2.25) and (2.26) into the resistance formula (1.13), only the last integral in the expression for G leads to a nonzero term. This gives the well-known Michell's integral of ship wave-making resistance:

$$R = \frac{\rho K_o^2}{\pi} \int_0^{\pi/2} [P(\theta)^2 + Q(\theta)^2] \sec^3\theta d\theta \qquad (2.27)$$

with $P(\theta) = \iint_{S_o} m(x',z') e^{K_o \sec^2\theta \cdot z'} \cdot \cos(K_o \sec\theta \cdot x') dx' dz'$

and $Q(\theta) = \iint_{S_o} m(x',z') e^{K_o \sec^2\theta \cdot z'} \cdot \sin(K_o \sec\theta \cdot x') dx' dz'$

Similarly, substituting (2.25) and (2.26) into the wave elevation formula (1.11), the wave elevation along the ship hull can be written as

$$\zeta(x,0) = -\frac{U}{g} \iint_{S_o} m(x',z') G_x(x,0,0;x',0,z') dx' dz' \qquad (2.28)$$

The disturbance velocity components around the ship hull can also be obtained in terms of Green's function and source distribution:

$$u(x,0,z) = \phi_x(x,0,z) = \iint_{S_o} m(x',z') G_x(x,0,z;x',0,z') dx' dz' \qquad (2.29)$$

$$v(x,0,z) = U f_x(x,z) = \frac{1}{2} m(x,z) \qquad (2.30)$$

$$w(x,0,z) = \phi_z(x,0,z) = \iint_{S_o} m(x',z') G_z(x,0,z;x',0,z') dx' dz' \qquad (2.31)$$

2.3 Thin-ship-panel approximation

In order to evaluate the numerical value of wave-making resistance (2.27), wave elevation along the ship hull (2.28) and disturbance velocity components around the ship hull (2.29), (2.30) and (2.31), the ship centerplane is discretized into a system of source panels with associated strengths, m_{ij}, $i = 1,2,\ldots M$, $j = 1,2,\ldots N$. It is assumed that the strength is uniformly distributed over each source panel, so that the strength in the above equations can be taken out of the integral sign for each source panel. Furthermore, Green's function (2.26) can be separated into three parts as

$$G(x,0,z;x',0,z') = G_1(x,0,z;x',0,z') + G_2(x,0,z;x',0,z') + G_3(x,0,z;x',0,z'), \tag{3.1}$$

where

$$G_1(x,0,z;x',0,z') = \frac{1}{4\pi}\left(-\frac{1}{r_1} + \frac{1}{r_2}\right) \tag{3.2}$$

$$G_2(x,0,z;x',0,z') = \frac{K_o}{\pi^2}\int_0^{\pi/2}\sec^2\theta d\theta \int_0^{\infty}\frac{[K+K_o\sec^2\theta\cdot(z+z')]e^{-K}}{[(K+K_o\sec^2\theta\cdot(z+z')]^2+(K_o\sec^2\theta\cdot\omega)^2}dK \tag{3.3}$$

$$G_3(x,0,z;x',0,z') = -H(x-x')\frac{2K_o}{\pi}\int_0^{\pi/2}\sec^2\theta\cdot e^{K_o\sec^2\theta\cdot(z+z')} \cdot \sin(K_o\sec^2\theta\,\omega)d\theta \tag{3.4}$$

for G_1: contribution of the radical term

G_2: contribution of the double integral term

G_3: contribution of the single integral term

Let us first assume there is a single source panel (i,j) with unit strength below the undisturbed free surface in a uniform stream of velocity U in the direction of the positive x-axis and the coordinates of four corner points of this source panel are (x_i', z_j'), (x_{i+1}', z_j'), (x_{i+1}', z_{j+1}') and (x_i', z_{j+1}') respectively. Then the elemental quantities induced by this source panel are as follows:

(1) The elemental disturbance velocity components, u_{ij}, v_{ij} and w_{ij}, on the centerplane.

From linearized boundary condition (1.9), once the source strength is given, the v-component velocity on the centerplane is determined. Only u-component and w-component velocities should be computed. We separate velocity components into three parts corresponding to the three parts of Green's function as in (3.1):

$$u_{ij}(x,0,z) = u_1(x,0,z) + u_2(x,0,z) + u_3(x,0,z) \tag{3.5}$$

$$w_{ij}(x,0,z) = w_1(x,0,z) + w_2(x,0,z) + w_3(x,0,z) \tag{3.6}$$

Equation (3.9) can be simplified as follows:

$$\text{Let } I_1 = \int_0^\infty e^{-K}\{\log[(K+K_o\sec^2\theta\cdot(z+z'))^2+(K_o\sec^2\theta\cdot\omega)^2]\}dK \quad (3.10)$$

Integrating by parts

$$I_1 = 2\log(K_o\sec^2\theta) + \log[(z+z')^2+\omega^2]$$

$$+ 2\int_0^\infty \frac{[K+K_o\sec^2\theta\cdot(z+z')]e^{-K}}{[K+K_o\sec^2\theta\cdot(z+z')]^2+(K_o\sec^2\theta\cdot\omega)^2}dK \quad (3.11)$$

The integral part of I_1 can be written in the complex form as

$$\text{Re}\{\int_0^\infty \frac{e^{-K}}{[K+K_o\sec^2\theta\cdot(z+z')]+i(K_o\sec^2\theta\cdot\omega)}dK\}$$

$$= \text{Re}\{e^{\lambda}\int_\lambda^\infty \frac{e^{-K}}{K}dK\}$$

$$= \text{Re}\{e^{\lambda}E_1(\lambda)\} \quad (3.12)$$

where

$E_1(\lambda)$ is a complex exponential integral

and $\lambda = K_o\sec^2\theta[(z+z')+i\omega]$

(a) The contribution of the radical term G_1

$$u_1(x,0,z) = \int_{z_j'}^{z_{j+1}'} \int_{x_i'}^{x_{i+1}'} \frac{\partial G_1}{\partial x}\, dx'\, dz'$$

$$= \frac{1}{4\pi} \{\log[(z'-z)+\sqrt{(x'-x)^2+(z'-z)^2}\,]- \log[(z'+z)+\sqrt{(x'-x)^2+(z'+z)^2}\,]\} \Big|_{x_i'}^{x_{i+1}'} \Big|_{z_j'}^{z_{j+1}'} \qquad (3.7)$$

$$w_1(x,0,z) = \int_{x_i'}^{x_{i+1}'} \int_{z_j'}^{z_{j+1}'} \frac{\partial G_1}{\partial z}\, dz'\, dx'$$

$$= \frac{1}{4\pi} \{\log[(x'-x)+\sqrt{(x'-x)^2+(z'-z)^2}\,] + \log[(x'-x)+\sqrt{(x'-x)^2+(z'+z)^2}\,]\} \Big|_{z_j'}^{z_{j+1}'} \Big|_{x_i'}^{x_{i+1}'} \qquad (3.8)$$

(b) The contribution of the double integral term G_2

$$u_2(x,0,z) = \int_{z_j'}^{z_{j+1}'} \int_{x_i'}^{x_{i+1}'} \frac{\partial G_2}{\partial x}\, dx'\, dz'$$

$$= \frac{-1}{2\pi^2} \int_0^{\pi/2} d\theta \int_0^{\infty} e^{-K}\{\log[(K+K_o\sec^2\theta\,(z+z'))^2+(K_o\sec^2\theta\cdot\omega)^2]\} \Big|_{x_i'}^{x_{i+1}'} \Big|_{z_j'}^{z_{j+1}'} dK \qquad (3.9)$$

Substituting (3.11) and (3.12) into (3.9)

$$u_2(x,0,z) = \frac{-1}{2\pi^2} \int_0^{\pi/2} \{2 \log(K_o \sec^2\theta) + \log[(z+z')^2+\omega^2] + 2 \operatorname{Re}[e^{\lambda} E_1(\lambda)]\} \Bigg|_{x_i'}^{x_{i+1}'} \Bigg|_{z_j'}^{z_{j+1}'} dK \quad (3.13)$$

The first term of the integral is independent of x´ and z´, so it can be eliminated. Equation (3.13) becomes

$$u_2(x,0,z) = \frac{-1}{2\pi^2} \int_0^{\pi/2} \{\log[(z+z')^2+\omega^2] + 2 \operatorname{Re}[e^{\lambda} E_1(\lambda)]\} \Bigg|_{x_i'}^{x_{i+1}'} \Bigg|_{z_j'}^{z_{j+1}'} dK \quad (3.14)$$

$$w_2(x,0,z) = \int_{x_i'}^{x_{i+1}'} \int_{z_j'}^{z_{j+1}'} \frac{\partial G_2}{\partial z} \, dz' dx'$$

$$= \frac{-1}{\pi^2} \int_0^{\pi/2} \sec\theta d\theta \int_0^{\infty} e^{-K} \{\tan^{-1}[\frac{K_o \sec^2\theta \cdot \omega}{K+K_o \sec^2\theta \cdot (z+z')}]\} \Bigg|_{z_j'}^{z_{j+1}'} \Bigg|_{x_i'}^{x_{i+1}'} dK \quad (3.15)$$

Similarly, let

$$I_1 = \int_0^{\infty} e^{-K} \{\tan^{-1}[\frac{K_o \sec^2\theta \cdot \omega}{K+K_o \sec^2\theta \cdot (z+z')}]\} \, dK \quad (3.16)$$

Since $z+z' < 0$, there is a discontinuity at $K = -K_o \sec^2\theta \cdot (z+z')$ in the integrand, the integral (3.16) has to be separated into two parts:

$$I_1 = \int_o^a e^{-K} \cdot \tan^{-1}\left[\frac{K_o \sec^2\theta \cdot \omega}{K-a}\right] dK + \int_a^\infty e^{-K} \cdot \tan^{-1}\left[\frac{K_o \sec^2\theta \cdot \omega}{K-a}\right] dK \tag{3.17}$$

where $a = -K_o \sec^2\theta \cdot (z+z')$

Integrating by parts

$$I_1 = (\text{sig}\ \omega)\ \pi \cdot e^{K_o \sec^2\theta \cdot (z+z')} + \tan^{-1}\left(\frac{\omega}{z+z'}\right)$$

$$+ \int_o^\infty \frac{(-K_o \sec^2\theta \cdot \omega)\, e^{-K}}{[K+K_o \sec^2\theta \cdot (z+z')]^2 + (K_o \sec^2\theta \cdot \omega)^2}\, dK \tag{3.18}$$

where $\text{sig}\ \omega = \begin{cases} -1 \text{ for } \omega < 0 \\ 1 \text{ for } \omega > 0 \end{cases}$

The integral part of I_1 can also be written in the complex form as

$$\text{Im}\left\{ \int_o^\infty \frac{e^{-K}}{[K+K_o \sec^2\theta \cdot (z+z')] + i\,(K_o \sec^2\theta \cdot \omega)}\, dK \right\}$$

$$= \text{Im}\left\{ e^\lambda \int_\lambda^\infty \frac{e^{-K}}{K}\, dK \right\}$$

$$= \text{Im}\{ e^\lambda E_1(\lambda) \} \tag{3.19}$$

where

$$\lambda = K_o \sec^2\theta\, [(z+z') + i\omega]$$

Substituting (3.18) and (3.19) into (3.15)

$$w_2(x,0,z) = \frac{-1}{\pi^2} \int_0^{\pi/2} \sec\theta\{(\text{sig}\omega)\pi . e^{K_o \sec^2\theta(z+z')}$$

$$+ \tan^{-1}\left(\frac{\omega}{z+z'}\right) + \text{Im}[e^{\lambda}E_1(\lambda)\ ]\} \Bigg|_{z'_j}^{z'_{j+1}} \Bigg|_{x'_i}^{x'_{i+1}} d\theta \qquad (3.20)$$

(c) The contribution of the single integral term G_3

$$u_3(x,0,z) = \int_{z'_j}^{z'_{j+1}} \int_{x'_i}^{x'_{i+1}} \frac{\partial G_3}{\partial x} dx'dz'$$

$$= H(x-x') . \frac{2}{\pi} \int_0^{\pi/2} e^{K_o \sec^2\theta\cdot(z+z')} . \sin[K_o \sec\theta\cdot(x-x')]$$

$$\Bigg|_{x'_i}^{x'_{i+1}} \Bigg|_{z'_j}^{z'_{j+1}} d\theta \qquad (3.21)$$

$$w_3(x,0,z) = \int_{x'_i}^{x'_{i+1}} \int_{z'_j}^{z'_{j+1}} \frac{\partial G_3}{\partial z} dx'dz'$$

$$= -H(x-x') . \frac{2}{\pi} \int_0^{\pi/2} e^{K_o \sec^2\theta\cdot(z+z')} . \cos[K_o \sec\theta\cdot(x-x')] . \sec\theta$$

$$\Bigg|_{x'_i}^{x'_{i+1}} \Bigg|_{z'_j}^{z'_{j+1}} d\theta \qquad (3.22)$$

(2) The elemental wave-making resistance R_{ij}

$$R_{ij} = \frac{\rho K_o^2}{\pi} \int_o^{\pi/2} [P_{ij}^2(\theta) + Q_{ij}^2(\theta)] \sec^3\theta d\theta \quad (3.23)$$

where

$$P_{ij}(\theta) = \int_{z'_j}^{z'_{j+1}} \int_{x'_i}^{x'_{i+1}} e^{K_o \sec^2\theta . z'} \cdot \cos(K_o \sec\theta \cdot x') dx' dz'$$

$$= \frac{1}{K_o^2 \sec^3\theta} \{(e^{K_o \sec^2\theta \cdot z'_{j+1}} - e^{K_o \sec^2\theta \cdot z'_j})$$

$$\cdot [\sin(K_o \sec\theta \cdot x'_{i+1}) - \sin(K_o \sec\theta \cdot x'_i)]\}$$

$$Q_{ij}(\theta) = \int_{z'_j}^{z'_{j+1}} \int_{x'_i}^{x'_{i+1}} e^{K_o \sec^2\theta \cdot z'} \cdot \sin(K_o \sec\theta \cdot x') dx' dz'$$

$$= \frac{-1}{K_o^2 \sec^3\theta} \{(e^{K_o \sec^2\theta \cdot z'_{j+1}} - e^{K_o \sec^2\theta \cdot z'_j})$$

$$\cdot [\cos(K_o \sec\theta \cdot x'_{i+1}) - \cos(K_o \sec\theta \cdot x'_i)]\}$$

Let

$$P_{ij}(\theta) = \frac{1}{K_o^2 \sec^3\theta} \overline{P}_{ij}(\theta) \quad \text{and} \quad Q_{ij}(\theta) = \frac{-1}{K_o^2 \sec^3\theta} \overline{Q}_{ij}(\theta)$$

The elemental wave-making resistance becomes

$$R_{ij} = \frac{\rho}{\pi K_o^2} \int_o^{\pi/2} [\overline{P}_{ij}^2(\theta) + \overline{Q}_{ij}^2(\theta)] \cos^3\theta d\theta \quad (3.24)$$

where

$$\overline{P}_{ij}(\theta) = (e^{K_o \sec^2\theta \cdot z'_{j+1}} - e^{K_o \sec^2\theta \cdot z'_j}) [\sin(K_o \sec\theta \cdot x'_{i+1})$$

$$- \sin(K_o \sec\theta \cdot x'_i)]$$

$$\overline{Q}_{ij}(\theta) = (e^{K_o \sec^2\theta \cdot z'_{j+1}} - e^{K_o \sec^2\theta \cdot z'_j}) [\cos(K_o \sec\theta \cdot x'_{i+1})$$

$$- \cos(K_o \sec\theta \cdot x'_i)]$$

(3) The elemental wave elevation $\zeta_{ij}(x,0,0)$

$$\zeta_{ij}(x,0,0) = - \frac{U}{g} \int_{z'_j}^{z'_{j+1}} \int_{x'_i}^{x'_{i+1}} \frac{\partial G}{\partial x} \, dx' dz' = - \frac{U}{g} u_{ij}(x,0,0) \quad (3.25)$$

The total quantities induced by a system of source panels (i,j) i=1,2,...M, J=1,2,...N, associated with a system of strength m_{ij}, i=1,2,...M, j=1,2,...N are the summation of the elemental quantities induced by each source panel as follows:

(a) The disturbance velocity components u(x,0,z), v(x,0,z) and w(x,0,z)

$$u(x,0,z) = \sum_{j=1}^{N} \sum_{i=1}^{M} m_{ij} \, u_{ij}(x,0,z) \quad (3.26)$$

$$v(x,0,z) = Uf_x(x,z) = m(x,z)/2 \tag{3.27}$$

$$w(x,0,z) = \sum_{j=1}^{N} \sum_{i=1}^{M} m_{ij}\, w_{ij}(x,0,z) \tag{3.28}$$

(b) The wave-making resistance R

$$R = \frac{\rho}{\pi K_o^2} \int_o^{\pi/2} [\overline{P}^2(\theta) + \overline{Q}^2(\theta)] \cos^3\theta d\theta \tag{3.29}$$

where

$$\overline{P}(\theta) = \sum_{j=1}^{N} \sum_{i=1}^{M} m_{ij}\, \overline{P}_{ij}(\theta)$$

$$\overline{Q}(\theta) = \sum_{j=1}^{N} \sum_{i=1}^{M} m_{ij}\, \overline{Q}_{ij}(\theta)$$

(c) The wave elevation $\zeta(x,0,0)$

$$\zeta(x,0,0) = -\frac{U}{g} \sum_{j=1}^{N} \sum_{i=1}^{M} m_{ij}\, u_{ij}(x,0,0) = -\frac{U}{g}\, u(x,0,0) \tag{3.30}$$

Defining the nondimensional wave elevation $\eta = \zeta/(\frac{U^2}{2g})$, then

$$\eta(x,0,0) = -2\frac{u(x,0,0)}{U} \tag{3.31}$$

2.4 Guilloton's Transformation:

Since in the thin ship theory, flow conditions on the hull surface, for example at a point P_1 in Fig. 2, are actually evaluated at P_o on the centerplane, Guilloton argues that if a transformation is to be applied to the y_o-direction in this way, similar transformations may be applied to the x_o- and z_o-directions. It is assumed that the time taken by a fluid particle to traverse the isobar from the forward perpendicular (F.P.) to a point P(x,y,z) on the ship hull in the real space (RS) is approximately equal to x_o/U, the linearized estimate of flow time from F.P. to the corresponding point $P_o(x_o,0,z_o)$ in the linearized space (LS), where the flow velocity evaluated at the point P_o should apply at the point P(as in Fig. 2)[10]. So

$$dt = dx_o/U \text{ in LS}$$

$$dt = ds/(U + u(x,y,z)) = ds/(U + u_o(x_o,0,z_o)) \text{ in RS}$$

for $ds = [1 + (\frac{\partial y}{\partial x})^2 + (\frac{\partial z}{\partial x})^2]^{1/2} dx \simeq (1+ \alpha^2/2)dx$

where $\alpha = \frac{\partial y}{\partial x}$, s is the distance along the isobar.

Hence, the transformation in the x_o-direction is

$$x = x_o + \int_{F.P.}^{x_o} \frac{u_o(x_o,0,z_o)/U - \alpha^2/2}{1 + \alpha^2/2} dx_o \qquad (4.1)$$

The transformation in the y_o-direction is the same as that in the thin ship theory.

$$y = y_o + \int_{F.P.}^{x_o} \frac{m(x_o,z_o)}{2U} dx_o \quad (y_o = 0) \qquad (4.2)$$

Since the fluid particle should be on the isobar, the transformation in the z_o-direction (refer to (1.11)) is

$$z = z_o - \frac{U}{g} u_o(x_o,0,z_o) \qquad (4.3)$$

From the geometrical point of view, the mapping function (4.1) maps a horizontal straight line on the centerplane in LS onto a space curve, the isobar along the ship hull, in RS. The same mapping relation will transform the prescribed values of velocity components, u_o, v_o, and w_o, (not velocity potential as in conformal mapping), along the straight line to corresponding values at points along the isobar.

So

$$u(x,y,z) = u_o(x_o,0,z_o) \qquad (4.4)$$

$$v(x,y,z) = v_o(x_o,0,z_o) \qquad (4.5)$$

$$w(x,y,z) = w_o(x_o,0,z_o) \qquad (4.6)$$

The wave elevations, $\zeta(x,y)$ and $\zeta^o(x_o,y_o)$, are defined on the undisturbed free surfaces, $z=0$ and $z_o=0$, in RS and LS respectively. So a mapping function for these two undisturbed free surfaces has only two components, $x = x(x_o,y_o)$ and $y = y(x_o,y_o)$. The wave elevation in RS can be written as

$$\zeta(x,y) = \zeta[x(x_o,y_o),y(x_o,y_o)] = \zeta^o(x_o,y_o) \qquad (4.7)$$

That means the wave elevation at the point (x,y) in RS is the same as that at the corresponding point (x_o,y_o) in LS.

Similarly, the hull functions, $f(x,z)$ and $f^o(x_o,z_o)$, are defined on the centerplanes, y=0 and y_o=0, in RS and LS respectively. The hull function in RS can be written as

$$f(x,z) = f[x(x_o,z_o),\ z(x_o,z_o)] = f^o(x_o,z_o) \qquad (4.8)$$

That means the breadth of the ship hull at point (x,z) in RS is the same as that at the corresponding point (x_o,z_o) in LS.

In fact, we can prove that this kind of transformation may make the solution in LS satisfy the exact kinematic boundary conditions, (1.1) and (1.2), in RS.

The exact and linearized kinematic boundary conditions on the free surface, (1.2) and (1.10), can be written as

$$[1 + \frac{u(x,y,\zeta(x,y))}{U}]\ \frac{\partial\zeta}{\partial x} + \frac{v(x,y,\zeta(x,y))}{U}\ \frac{\partial\zeta}{\partial y} = \frac{w(x,y,\zeta(x,y))}{U} \qquad (4.9)$$

and

$$\frac{\partial\zeta^o}{\partial x_o} = \frac{w_o(x_o,y_o,0)}{U} \qquad (4.10)$$

Since the mapping function between the two undisturbed free surfaces is $x = x(x_o,y_o)$ and $y = y(x_o,y_o)$ and the wave elevations at the corresponding points, (x,y) and (x_o,y_o), are the same, equation (4.10) can be written as

$$\frac{\partial x}{\partial x_o}\frac{\partial \zeta}{\partial x} + \frac{\partial y}{\partial x_o}\frac{\partial \zeta}{\partial y} = \frac{w_o(x_o,y_o,0)}{U} \qquad (4.11)$$

Comparing (4.11) and (4.9), we may take

$$u(x,y,\zeta(x,y)) = u_o(x_o,y_o,0) \qquad (4.12)$$

$$v(x,y,\zeta(x,y)) = v_o(x_o,y_o,0) \qquad (4.13)$$

$$w(x,y,\zeta(x,y)) = w_o(x_o,y_o,0) \qquad (4.14)$$

and equation (4.9) will become

$$[1 + \frac{u_o(x_o,y_o,0)}{U}]\frac{\partial \zeta}{\partial x} + \frac{v_o(x_o,y_o,0)}{U}\frac{\partial \zeta}{\partial y} = \frac{w_o(x_o,y_o,0)}{U} \qquad (4.15)$$

Then the mapping function between the two undisturbed free surfaces may be written as

$$x = x_o + \int_{-\infty}^{x_o} \frac{u_o(x_o,y_o,0)}{U}\, dx_o \qquad (4.16)$$

$$y = y_o + \int_{-\infty}^{x_o} \frac{v_o(x_o,y_o,0)}{U}\, dx_o \qquad (4.17)$$

If the transformation of the velocity components, (4.12) to (4.14), is considered the transformation in the z_o-direction can be found from (4.10) as

$$z = z_o + \int_{-\infty}^{x_o} \frac{w_o(x_o,y_o,0)}{U}\, dx_o, \quad (z_o = 0) \tag{4.18}$$

From (1.11), (4.12) and (4.18), the dynamic boundary condition on the free surface in RS satisfied by this transformation is

$$\zeta(x,y) = -\frac{U}{g}\, u(x,y,\zeta(x,y)) \tag{4.19}$$

Although equation (4.19) is not the exact dynamic boundary condition (see (1.7)), it is a great improvement.

Similarly, in order to make the linearized solution satisfy the exact kinematic boundary condition on the hull surface in RS, the other set of transformations can be found as

$$x = x_o + \int_{-\infty}^{x_o} \frac{u_o(x_o,0,z_o)}{U}\, dx_o \tag{4.20}$$

$$y = y_o + \int_{-\infty}^{x_o} \frac{v_o(x_o,0,z_o)}{U}\, dx_o \tag{4.21}$$

$$z = z_o + \int_{-\infty}^{x_o} \frac{w_o(x_o,0,z_o)}{U}\, dx_o \tag{4.22}$$

and

$$u(x,f(x,z),z) = u_o(x_o,0,z_o) \tag{4.23}$$

$$v(x,f(x,z),z) = v_o(x_o,0,z_o) \tag{4.24}$$

$$w(x,f(x,z),z) = w_o(x_o,0,z_o) \tag{4.25}$$

Comparing (4.20) to (4.25) with (4.12) to (4.18), it can be shown that these two sets of transformation are consistent. The transformation for the whole flow field can be written as

$$x = x_o + \int_{-\infty}^{x_o} \frac{u_o(x_o,y_o,z_o)}{U}\, dx_o \quad (4.26)$$

$$y = y_o + \int_{-\infty}^{x_o} \frac{v_o(x_o,y_o,z_o)}{U}\, dx_o \quad (4.27)$$

$$z = z_o + \int_{-\infty}^{x_o} \frac{w_o(x_o,y_o,z_o)}{U}\, dx_o \quad (4.28)$$

$$u(x,y,z) = u_o(x_o,y_o,z_o) \quad (4.29)$$

$$v(x,y,z) = v_o(x_o,y_o,z_o) \quad (4.30)$$

$$w(x,y,z) = w_o(x_o,y_o,z_o) \quad (4.31)$$

and

$$\zeta(x,y) = \zeta^o(x_o,y_o) \quad (4.32)$$

$$f(x,z) = f^o(x_o,z_o) \quad (4.33)$$

Since only the ship hull in RS is known, the transformation (4.26) to (4.33), can be interpreted as a kind of inverse streamline tracing method which forces the ship hull in RS to be a stream surface. But obviously, the velocity components cannot satisfy the continuity equation in RS. (refer to conclusion)

Now, we assume that only the points on the centerplane in LS are considered; that the streamlines can be replaced by the isobars along the ship hull (recall the draft-length ratio of the ship is small); that the longitudinal shift can be calculated from the F.P.; and that the longitudinal slope of the ship hull is small. Then the transformation is found to be identical to that of Guilloton's method, (4.1) to (4.8).

In our problem, since the ship hull in RS and a specified Froude number are given, the "linearized hull" cannot be found directly. However, from the formulas of Guilloton's method, the source strength $m(x_o, z_o)$ in LS can be expressed in terms of the slopes, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial z}$ of the given "real hull" as

$$\frac{m(x_o, z_o)}{2U} = \frac{1+u_o(x_o, 0, z_o)/U}{1+\alpha^2/2} \frac{\partial f}{\partial x} - \frac{U^2}{g} \frac{\partial}{\partial x_o} \left(\frac{u_o(x_o, 0, z_o)}{U}\right) \frac{\partial f}{\partial z} \qquad (4.34)$$

Therefore, the "linearized hull" can be found by the following iterative process:

(1) Assume a trial value of the source strength for each source panel. The longitudinal local slope of the "real hull" corresponding to the centroid of each panel may be taken as the initial value of the quantity, m/2U, i.e. initially, the "linearized hull" is assumed to be the same as the given "real hull".

(2) Calculate the u_o-component velocity (3.26) along each longitudinal straight line on the centerplane in LS.

(3) Substitute the calculated u_o-component velocity into (4.1) and (4.3) to find the profile $\zeta(x,y)$ of each isobar in RS corresponding to the straight line in LS.

(4) Calculate the slopes, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial z}$, of the "real hull" corresponding to the points on each isobar and substitute them into (4.34) to find the new strength of each panel.

(5) If the new strength of each source panel is the same as the trial value, the calculation stops. Otherwise, the new strength of each source panel will be taken as the new trial value and the calculation repeated.

After the "linearized hull" is found, the flow quantities can be calculated by the thin ship theory and transformed to that in RS. The wave-making resistance in RS actually should be calculated by integrating the pressure around the real hull as (1.6), but if we neglect the higher order term (as below), the wave-making resistance of the "linearized hull" can be taken as that of the "real hull".

The transformation of the points on the centerplane in LS can be written as

$$x = x_o + A(x_o,z_o) \quad , \quad A(x_o,z_o) = \int_o^{x_o} \frac{u_o(x_o,0,z_o)/U-\alpha^2/2}{1 + \alpha^2/2} dx_o \quad (4.35)$$

$$z = z_o + C(x_o,z_o) \quad , \quad C(x_o,z_o) = -\frac{U}{g} u_o(x_o,0,z_o) \quad (4.36)$$

and equation (4.23) to (4.25)

According to Bernoulli's equation, the pressure at the point on the real hull is

$$P(x,f(x,z),z) = -\rho[Uu(x,f(x,z),z) + \frac{1}{2}(u^2+v^2+w^2)+gz] \quad (4.37)$$

Substituting the corresponding linearized quantities into RHS of (4.37).

$$P(x,f(x,z),z) = -\rho\{U\, u_o(x_o,0,z_o) + \frac{1}{2}(u_o^2+v_o^2+w_o^2)+g[z_o+C(x_o,z_o)]\} \quad (4.38)$$

Expanding the longitudinal slope of the "real hull", $f_x(x,z)$, with respect to the linearized quantities, we can obtain:

$$f_x(x,z) = f^o_{x_o}(x_o,z_o)+(-f^o_{x_o}A_{x_o} - f^o_{z_o}C_{x_o}) + \ldots \quad (4.39)$$

and

$$dxdz = |x_{x_o}z_{z_o}-x_{z_o}z_{x_o}|dx_o dz_o$$

$$= |1 + C_{z_o}+A_{x_o}+A_{x_o}C_{z_o}-C_{x_o}A_{z_o}|dx_o dz_o \quad (4.40)$$

Substituting (4.38), (4.39) and (4.40) into (1.6), the wave-making resistance of the real hull can be written as

$$R = -2\rho U \iint_{S_o} u_o(x_o,0,z_o)f^o_{x_o}(x_o,z_o)dx_o dz_o$$

$$- 2\rho \iint_{S_o} [U\, u_o(x_o,0,z_o)+gC(x_o,z_o)](C_{z_o}f^o_{x_o}-C_{x_o}f^o_{z_o})dx_o dz_o \quad (4.41)$$

$$+ \ldots\ldots\ldots\ldots\ldots$$

The first term of (4.41) is the wave-making resistance, R_o, of the linearized hull (see (1.13)), and since $C(x_o,z_o) = -\frac{U}{g}u_o(x_o,z_o)$, we find the second term is zero. Then the difference in the wave-making resistance between the "linearized hull" and the "real hull" is of the fourth order, i.e.

$$R = -2\rho U \iint_{S_o} u_o(x_o,0,z_o) f^o_{x_o}(x_o,z_o) dx_o dz_o + O(\varepsilon^4) \qquad (4.42)$$

For certain nonlinear problems, we may find a transformation and then transform the linearized solution from the "linearized space" to the "real space". In this way, we may obtain a "better" solution. At least the solution satisfies the field equation and boundary conditions to the same order. But this is not the case for the three-dimensional steady ship-flow problem. Therefore the solution obtained from Guilloton's method can only be justified by comparing the computed results with the experimental results.

2.5 Cubic spline curve fitting[11]

Ordinarily, the shape of the ship hull is defined by the hull offsets. In order to find the slopes, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial z}$, of the ship hull, we should find the equations of the smooth ship lines i.e. waterlines and framelines, passing through the given ship-offsets.

Let us consider a single-valued curve with continuous first and second derivatives passing through the given points, say (x_i, y_i), i=1,2,...N. This curve can be divided into (N-1) segments corresponding to the given (N-1) intervals. It is assumed that the equation of each segment is a three-degree polynomial, for example: the equation of ith segment is:

$$f_i(x) = A_i(x-x_i)^3 + B_i(x-x_i)^2 + C_i(x-x_i) + D_i \quad (5.1)$$

Hence, there are 4(N-1) unknowns, A_i, B_i, C_i and D_i, i=1,2,... N-1. We can find 4(N-1) equations from the following conditions to solve for these 4(N-1) unknowns:

(a) The curve should pass through N given points (N equations)

$$f_i(x_i) = y_i, \quad i=1,2,...N-1 \quad (5.2)$$

$$f_{N-1}(x_N) = y_N \quad (5.3)$$

(b) The curve is continuous (N-2 equations)

$$f_i(x_{i+1}) = f_{i+1}(x_{i+1}) \quad i=1,2,...N-2 \quad (5.4)$$

(c) The first derivative of the curve is continuous (N-2 equations)

$$f'_i(x_{i+1}) = f'_{i+1}(x_{i+1}) \quad i=1,2,...N-2 \quad (5.5)$$

(d) The second derivative of the curve is continuous (N-2 equations)

$$f''_i(x_{i+1}) = f''_{i+1}(x_{i+1}) \quad i=1,2,...N-2 \quad (5.6)$$

(e) Assume second derivatives of the ends are zero (2 equations)

$$f''_1(x_1) = 0 \quad (5.7)$$

$$f''_{N-1}(x_N) = 0 \tag{5.8}$$

Solving this system of simultaneous equations is very tedious, after some manipulations, we can obtain the following formulas:

$$\begin{aligned} A_i &= \frac{1}{6\Delta x_i}\left(y''_{i+1} - y''_i\right) \\ B_i &= \frac{1}{2}\, y''_i \\ C_i &= \frac{\Delta y_i}{\Delta x_i} - \frac{1}{6}\,\Delta x_i\left(y''_{i+1} + 2y''_i\right) \\ D_i &= y_i \end{aligned} \tag{5.9}$$

and

$$\begin{bmatrix} 2(\Delta x_1 + \Delta x_2) & \Delta x_2 & & & \bigcirc \\ \Delta x_2 & 2(\Delta x_2 + \Delta x_3) & \Delta x_3 & & \\ & \Delta x_3 & 2(\Delta x_3 + \Delta x_4) & \Delta x_4 & \\ & \bigcirc & \ddots & \ddots & \ddots \\ & & & \Delta x_{N-2} & 2(\Delta x_{N-2}+\Delta x_{N-1}) \end{bmatrix} \cdot \begin{bmatrix} y''_2 \\ y''_3 \\ y''_4 \\ \vdots \\ y''_{N-1} \end{bmatrix}$$

$$
= \begin{bmatrix} 6(\Delta y_2/\Delta x_2 - \Delta y_1/\Delta x_1) - \Delta x_1 y_1'' \\ 6(\Delta y_3/\Delta x_3 - \Delta y_2/\Delta x_2) \\ 6(\Delta y_4/\Delta x_4 - \Delta y_3/\Delta x_3) \\ \cdot \\ \cdot \\ \cdot \\ 6(\Delta y_{N-1}/\Delta x_{N-1} - \Delta y_{N-2}/\Delta x_{N-2}) - \Delta x_{N-1} y_N'' \end{bmatrix} \quad (5.10)
$$

where

$$\Delta x_i = x_{i+1} - x_i \quad , \quad \Delta y_i = y_{i+1} - y_i$$

$$y_i'' = f_i''(x_i) \quad , \quad y_{i+1}'' = f_i''(x_{i+1})$$

and $y_i = f_i(x_i)$

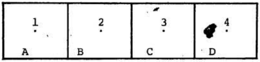
A special feature of equation (5.9) is that four coefficients of arbitrary segment i only depend on the two given points, (x_i, y_i) and (x_{i+1}, y_{i+1}), and the two second derivatives, y_i'' and y_{i+1}'', at given points. Hence, once equation (5.10) has been solved, the whole curve can be specified and the slope at any point of the curve can easily be calculated.

CHAPTER 3

COMPUTATIONAL METHOD AND RESULTS

We would like efficient computation and accurate results not only for wave-making resistance but also for flow quantities around the "real hull". The major differences in the computational scheme used from that of others were firstly using the source strength as the convergence criterion for finding the "linearized hull" and secondly using the cubic spline function for fitting the given ship-offsets. The computer programs were set up, taking into consideration the symmetric and antisymmetric properties of the element disturbance velocity components induced by the source panels. The whole computational scheme can be separated into two parts. One is to compute the disturbance velocity components, $u_o(x_o,0,z_o)$ and $w_o(x_o,0,z_o)$, induced by the source panels with "unit" strength, and the other is to find the source strength in LS as mentioned in section 2.4.

For the techniques used in computing the first part, we assume that the area of the centerplane of the "linearized hull" is the same as that of the "real hull". The centerplane is divided into 200 rectangular source panels (twenty-one stations including F.P. and A.P., and eleven waterlines including the base line and design waterline). The centroid of each panel is chosen as the control point (field point), so that there are 40,000 elemental disturbance velocities to be calculated, i.e. the elemental disturbance velocity of each source panel with respect to each control point. It is a very tedious work. But as mentioned before, we can take advantage of symmetric and antisymmetric properties and the Heaviside function in the formulas of the elemental disturbance velocity components to reduce the laborious work considerably. For example: there are four source panels, A,B,C, and D in longitudinal direction such as



The corresponding control points are 1,2,3 and 4 respectively. There are 16 elemental disturbance velocities to be calculated. From (3.5), (3.7), (3.14) and (3.21), we know the elemental disturbance velocity of u-component includes two parts, one is the symmetric part, $u_s = u_1 + u_2$, and the other is the free wave part, $u_f = u_3$. So in this example, the 16 elemental disturbance velocities of u-components can be written as

$$u(I,J) = u_s(I,J) + u_f(I,J) \qquad I=A,B,C, \text{ and } D,\ J=1,2,3 \text{ and } 4$$

Since u_s is symmetric and the area of each source panel is the same, we have:

$$u_s(A,1) = u_s(B,2) = u_s(C,3) = u_s(D,4)$$

$$u_s(A,2) = u_s(B,3) = u_s(C,4) = u_s(B,1) = u_s(C,2) = u_s(D,3)$$

$u_s(A,3) = u_s(B,4) = u_s(C,1) = u_s(D,2)$

and $u_s(A,4) = u_s(D,1)$

Hence, only $u_s(A,4)$, $u_s(B,4)$, $u_s(C,4)$ and $u_s(D,4)$ will be calculated. If the control point is in front of the source panel, the free wave part, u_f, has no contribution. We also have

$u_f(B,1) = u_f(C,2) = u_f(D,3) = u_f(C,1) = u_f(D,2) = u_f(D,1) = 0$

and similarly

$u_f(A,1) = u_f(B,2) = u_f(C,3) = u_f(D,4)$

$u_f(A,2) = u_f(B,3) = u_f(C,4)$

$u_f(A,3) = u_f(B,4)$

Only $u_f(A,4)$, $u_f(B,4)$, $u_f(C,4)$ and $u_f(D,4)$ will be calculated. From (3.6), (3.8) and (3.20), $w_1 + w_2$ has the antisymmetric property, and the same technique can be used to calculate the elemental disturbance velocity of w-component. Thus only the elemental disturbance velocities of each panel with respect to the last control point have to be calculated. In the case of 200 panels with 200 control points, the number of calculations will be reduced from 40,000 to 2,000.

In the second part of the computational scheme, we first find the second derivatives, $\frac{\partial^2 y}{\partial x^2}$ and $\frac{\partial^2 y}{\partial z^2}$, at each intersecting point of waterlines and stations, thus the coefficients of each segment function can be calculated. Then the y-coordinate and the first derivatives, $\frac{\partial y}{\partial x}$ and $\frac{\partial y}{\partial z}$, at any point on the ship hull can be obtained by interpolation. If the ship hull is smooth, we can generate the ship lines very accurately with this method. The number of iterations for finding the "linearized hull" depends on the value of the Froude number and the shape of the ship hull. To save computer time, we can use the "linearized hull" of a lower Froude number as the trial value for a higher Froude number. Generally, the number of iterations is between 40 and 60 for the convergence criterion 1×10^{-4}.

Two models have been selected for computation: Wigley model 3012, a mathematical hull form, and Series 60 block 60, a conventional merchant ship hull. The geometries of the models are given as follows:

(1) Wigley model 3012

$B/L = 0.1$, $H/L = 0.0625$, $C_B = 0.444$, $C_{PR} = 0.667$

$C_x = 0.667$, $C_s = 0.661$ and $L/L_{PP} = 1.000$ (where L=LWL)

The hull surface is defined by

$$y = \frac{B}{2} \left[1 - \left(\frac{2x}{L}\right)^2\right] \left[1 - \left(\frac{z}{H}\right)^2\right]$$

(2) Series 60 block 60

$B/L_{PP} = 0.1333$, $H/L_{PP} = 0.0533$, $C_B = 0.600$, $C_{PR} = 0.614$

$C_x = 0.977$, $C_s = 0.710$ and $L/L_{PP} = 1.0167$ (where L = LWL)

The ship-offsets are in table 1, the bow and stern contours and lines are shown in Fig. 3 and Fig. 4.

The computed results of the wave-making resistance, wave profile, isobars and flow directions along the ship hull are shown in Fig. 5 to Fig. 21. The wave resistance and wave profile curves are plotted over the experimental curves from reference [12] for comparison.

Generally speaking, Guilloton's method gives very good results which are close to the experimental results for Froude number from 0.25 to 0.35 approximately. Especially, in the wave resistance curves, there are no large humps and hollows which usually exist in Michell's resistance curve (see Fig. 5 and Fig. 12). Unfortunately, there are no experimental results to compare with the calculated isobars and flow directions, but from the tendency of the isobars and flow directions, it appears in the correct sense because at the free surface the streamline and isobar coincide, below the free surface, the streamlines have a stronger tendency to go down near the bow and come up near the stern as in [13]. Comparing with series 60 block 60, Wigley model 3012 has smaller "breadth" and simpler geometrical shape, the variation of its isobars and flow directions are smaller, especially around the stern.

In the last two figures, Fig. 22 and Fig. 23, the calculated wave-making resistance of Wigley hull and Series 60 by Guilloton's method are compared with the results obtained by other researchers. The computational method developed in this thesis has demonstrated its effectiveness and the computed results are quite consistent with that of others, even showing some improvement! The deviations may be due to different numerical techniques for computation, such as the number of source panels, the method of curve fitting, the convergence criterion, etc.

CHAPTER 4

DISCUSSION AND CONCLUDING REMARKS

The main purpose of this thesis is to account for some nonlinear effects of steady ship-flow problem by Guilloton's transformation. Although there is no rigorously theoretical background for Guilloton's method, the computed results have shown that the prediction of the ship flow for a certain Froude number range is very good. In the range of Froude numbers, roughly between 0.25 and 0.35, we may infer that just considering the flow in the vicinity of the ship hull, it could be more important to satisfy the hull and free surface boundary conditions than the field equation. The nonlinear effects included in Guilloton's method are to reduce the oscillatory behaviour of the wave resistance curve based on the thin ship theory and to shift the phase of the wave elevation along the ship hull as that of other higher order theories. [14]. From the definition of the fluid domain in the steady ship-flow problem, Guilloton's method seems more reasonable than the thin ship theory, since the domain is defined below the "disturbed" free surface and out of the interior region of the ship. Another feature of Guilloton's method is to correct the paradox of the thin ship theory which implies that the wave resistance is the same no matter which direction the ship moves, bow or stern ahead. But due to the results of the transformation, the wave-making resistances for these two cases will be different.

and reliable methods to obtain reasonable results for the stage of preliminary design of conventional ships.

The reasons why the calculated results do not match the experimental results at high Froude number may be due to:

(a) The sinkage and trim are not considered by Guilloton's method.

(b) The isobars calculated by equation (4.3) are no longer correct.

(c) The thin ship theory may not apply to the linearized hull since the distortion of hull geometry is too excessive.

(d) It can be erroneous for the linearized solution to satisfy the field equation in RS.

For a ship with a flat bottom, such as Series 60 block 60, the bottom part of the hull can not be described by the hull equation, $y = \pm f(x,z)$, so that the calculated flow near the bottom can not exactly correspond to the real flow, even though good results for wave-making resistance are obtained. The calculated flow based on the thin ship theory or Guilloton's method may not be good enough to form the starting point of the boundary layer calculation for ship hulls.

It appears that Guilloton's method for solving the steady ship-flow problem has many disadvantages. The difficulties of the flow problem still can not totally be resolved. However, comparing with other sophisticated methods, for example: higher order theory, finite element method and finite difference method, we may say that Guilloton's method is one of the simplest

REFERENCES

1. Michell, J.H., "The wave resistance of a ship", Philos. Mag. (5) 45, 1898, pp. 106-123.

2. Guilloton, R., "L'étude théorique du bateau en fluide parfait", Bull. Ass. Tech. Mar. Aeronaut. 64, 1964, pp. 538-561.

3. Van Dyke, M., "Perturbation methods in fluid mechanics" Academic Press, New York, 1964.

4. Noblesse, F., "A perturbation analysis of the wave-making resistance of a ship, with an interpretation of Guilloton's method", J. Ship Res. Vol. 19, 1975, pp. 140-148.

5. Dagan, G., "A method of computing nonlinear wave resistance of thin ships by coordinate straining", J. Ship Res. Vol. 19, 1975, pp. 149-154.

6. Wehausen, J.V., "The wave resistance of ships", Advances in Applied Mechanics, Academic Press, New York, Vol. 13, 1973.

7. Kellog, O.D., "Foundation of potential theory" Berlin; Springer, 1929.

8. Havelock, T.H., "The theory of wave resistance", Proc. Roy. Soc. Ser., A.138, 1932, pp. 339-348, Coll. papers, pp. 377-389.

9. Eggers, K., Shama, S., Ward, L., "An assessment of some experimental methods for determining the wave-making characteristics of a ship form", Trans. Soc. Nav. Arch., Mar. Eng. 75, 1967, pp. 112-144.

10. Gadd, G.E., "Wave resistance calculations by Guilloton's method", Tran. Royal Inst. Naval Arch. Vol. 115, 1973, pp. 377-384.

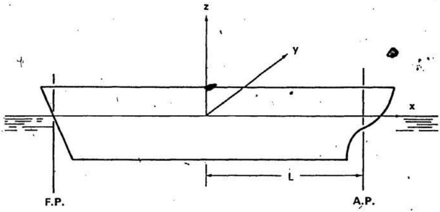
11. Spath, H., "Spline algorithms for curves and surfaces", Translated by Haskins, W.D., and Sager, H.W., Utilitas Mathematica Publishing Inc., 1974.

12. "Proceedings of the workshop on ship wave-resistance computation", Washington, D.C., 1979.

13. Misra, S.C., "The calculation of potential flow on ship hulls" Royal Inst. Naval Arch., 1983 (issued for written discussion).

14. Hong, S.Y., "Numerical calculation of second-order wave resistance" J. Ship Res. Vol. 21, 1977, pp. 94-106.

COORDINATE SYSTEM



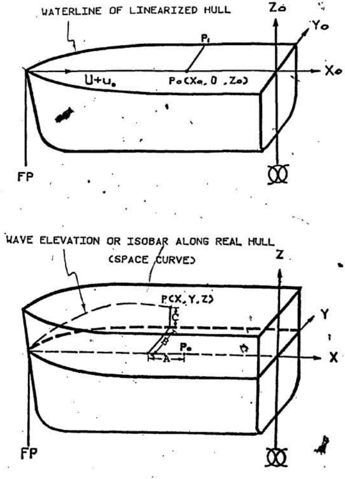
x, y, z — Translating coordinate system with x in the opposite direction to the ship's forward motion, z vertically upward, and the origin at the intersection of the planes of the undisturbed free-surface and the midship section.*

x', y', z' — Coordinate system fixed in ship and coinciding with the x-y-z system.

*Midship section is, by definition, at the midpoint between perpendiculars.

FIGURE 1

GEOMETRY OF GUILLOTON'S TRANSFORMATION



$A(x_o, z_o)$
$B(x_o, z_o)$: Displacements of Guilloton's transformation in x_o-, y_o- and z_o- direction, respectively.
$C(x_o, z_o)$

FIGURE 2

TABLE 1 — TABLE OF OFFSETS

SERIES 60, $C_B = 0.60$
(FROM TODD, 1953)

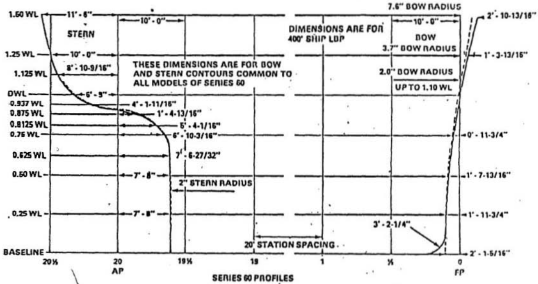
Half breadths of waterline given as fraction of maximum beam on each waterline

Model = 4210W
W.L. 1.00 is the designed load waterline

Forebody prismatic coefficient = 0.581
Afterbody prismatic coefficient = 0.646
Total prismatic coefficient = 0.614

		Waterlines							Area as fraction of max. area to 1.00 W.L.
Sta.	Tan.	0.075	0.25	0.50	0.75	1.00	1.25	1.50	
FP	0.000	0.000	0.000	0.000	0.000	0.000	0.020	0.042	0.000
½	0.009	0.032	0.042	0.041	0.043	0.051	0.076	0.120	0.042
1	0.013	0.064	0.082	0.087	0.090	0.102	0.133	0.198	0.085
1½	0.019	0.095	0.126	0.141	0.148	0.160	0.195	0.278	0.135
2	0.024	0.127	0.178	0.204	0.213	0.228	0.270	0.360	0.192
3	0.055	0.196	0.294	0.346	0.368	0.391	0.440	0.531	0.323
4	0.134	0.314	0.436	0.502	0.535	0.562	0.607	0.683	0.475
5	0.275	0.466	0.589	0.660	0.691	0.718	0.754	0.804	0.630
6	0.469	0.630	0.733	0.802	0.824	0.841	0.862	0.889	0.771
7	0.666	0.779	0.854	0.906	0.917	0.926	0.936	0.946	0.880
8	0.831	0.898	0.935	0.971	0.977	0.979	0.981	0.982	0.955
9	0.945	0.964	0.979	0.996	1.000	1.000	1.000	1.000	0.990
10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
11	0.965	0.982	0.990	1.000	1.000	1.000	1.000	1.000	0.996
12	0.882	0.922	0.958	0.994	1.000	1.000	1.000	1.000	0.977
13	0.767	0.826	0.892	0.962	0.987	0.994	0.997	1.000	0.938
14	0.622	0.701	0.781	0.884	0.943	0.975	0.990	0.999	0.863
15	0.463	0.560	0.639	0.754	0.857	0.937	0.977	0.994	0.750
16	0.309	0.413	0.483	0.592	0.728	0.857	0.933	0.975	0.609
17	0.168	0.267	0.330	0.413	0.541	0.725	0.844	0.924	0.445
18	0.065	0.152	0.193	0.236	0.321	0.536	0.709	0.834	0.268
18½	0.032	0.102	0.130	0.156	0.216	0.425	0.626	0.769	0.187
19	0.014	0.058	0.076	0.085	0.116	0.308	0.530	0.686	0.109
19½	0.010	0.020	0.020	0.022	0.033	0.193	0.418	0.579	0.040
AP	0.000	0.000	0.000	0.000	0.000	0.082	0.270	0.420	0.004
Max half beam*	0.710	0.866	0.985	1.000	1.000	1.000	1.000	1.000	

*As fraction of maximum load waterline beam.



NOTE: 15 INCH BOW RADIUS AT 1.75 WL
24 INCH BOW RADIUS AT 1.95 WL

FIGURE 3 — Bow and Stern Contours
(from Todd, 1963)

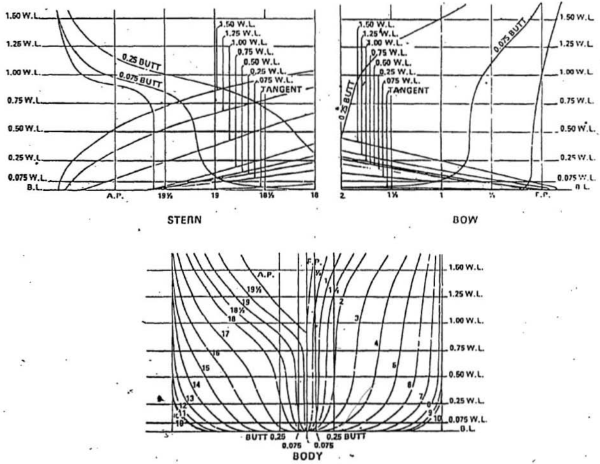


FIGURE 4 — Lines of Series 60, C_B = 0.60 Model 4210W (from Todd, 1953)

WIGLEY HULL - RESISTANCE COEFFICIENT

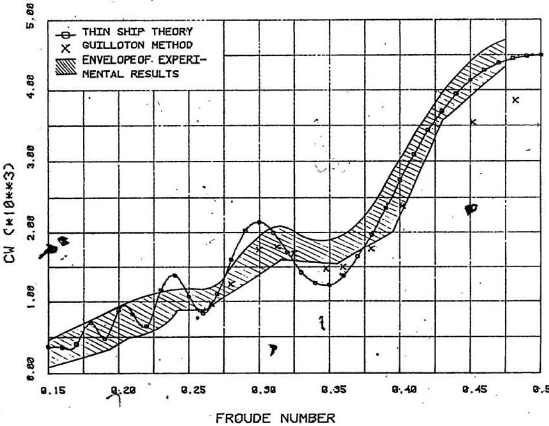
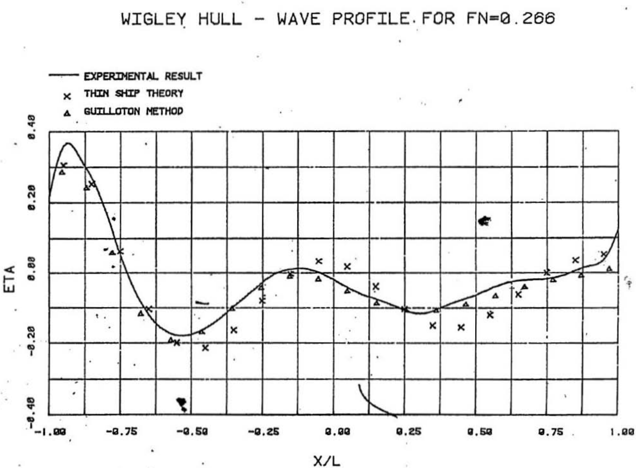


FIGURE 5



WIGLEY HULL - WAVE PROFILE FOR FN=0.266
EXPERIMENTAL RESULT
THIN SHIP THEORY
GUILLOTON METHOD
ETA
0.40
0.20
0.00
-0.20
-0.40
-1.00
-0.75
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
X/L

FIGURE 6

WIGLEY HULL - ISOBARS FOR FN=0.266

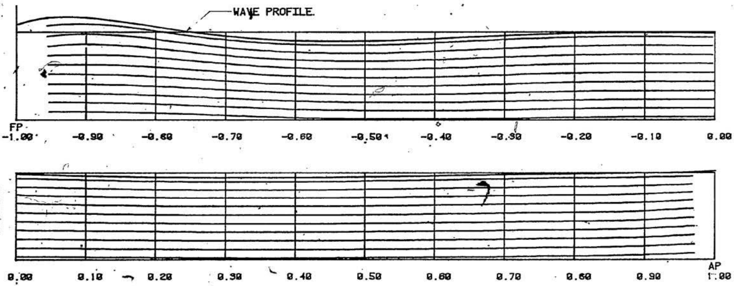


FIGURE 7

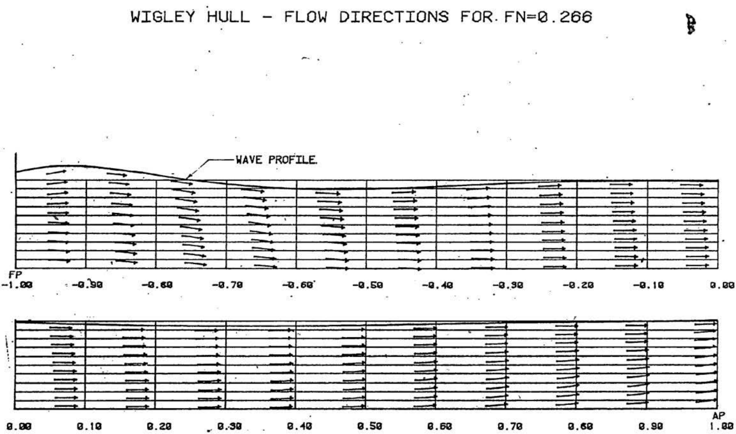

WIGLEY HULL - FLOW DIRECTIONS FOR FN=0.266
WAVE PROFILE
FP
-1.00
-0.90
-0.80
-0.70
-0.60
-0.50
-0.40
-0.30
-0.20
-0.10
0.00
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
AP
1.00

FIGURE 8

WIGLEY HULL - WAVE PROFILE FOR FN=0.348

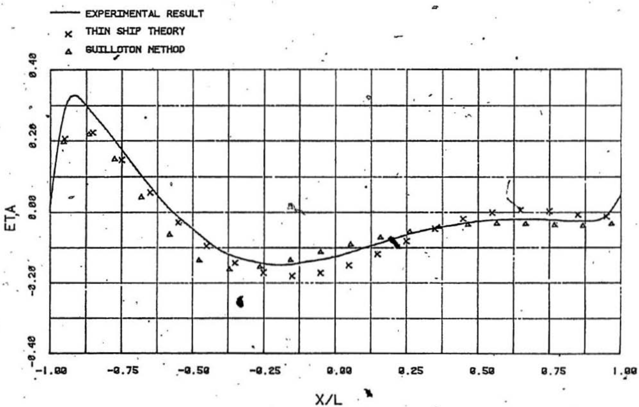


FIGURE 9

WIGLEY HULL - ISOBARS FOR FN=0.348

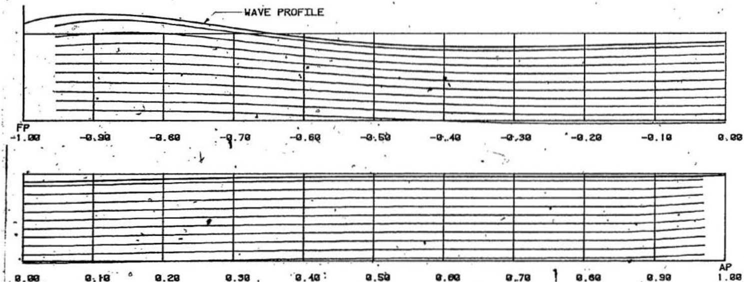


FIGURE 10

WIGLEY HULL - FLOW DIRECTIONS FOR FN=0.348

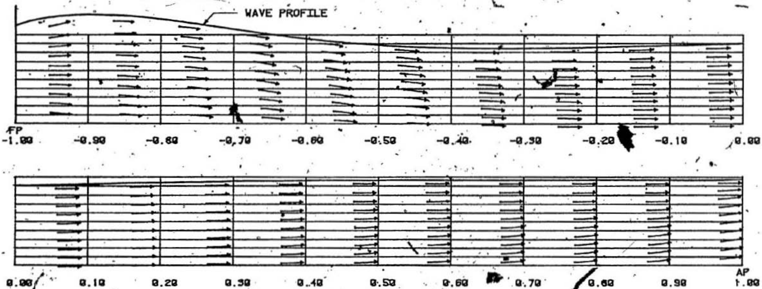


FIGURE 11

SERIES 60 BLOCK 60

- RESISTANCE COEFFICIENT

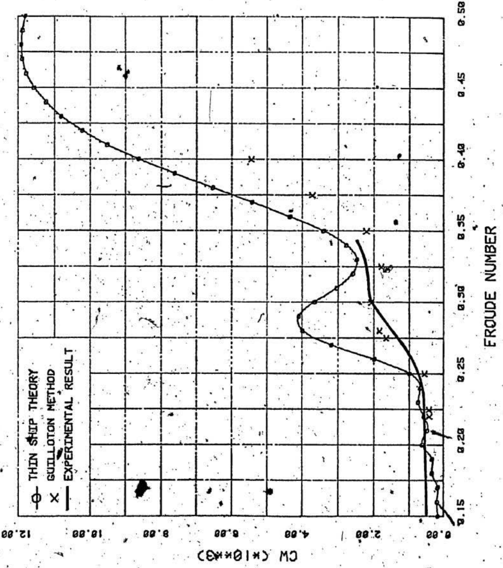


FIGURE 12

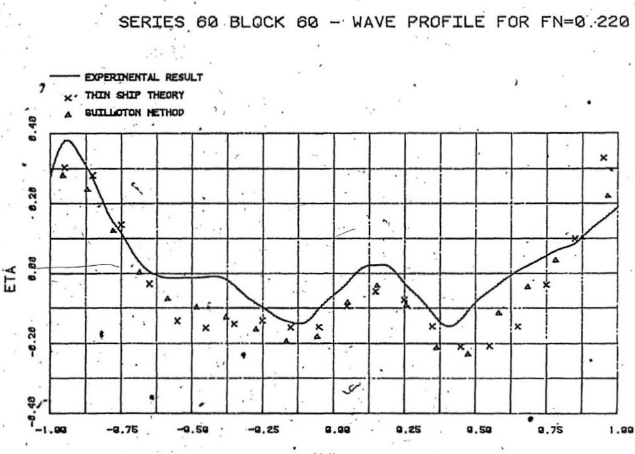

SERIES 60 BLOCK 60 - WAVE PROFILE FOR FN=0.220
EXPERIMENTAL RESULT
THIN SHIP THEORY
GUILLOTON METHOD
ETA
0.40
0.20
0.00
-0.20
-0.40
-1.00
-0.75
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
X/L

FIGURE 13

SERIES 60 BLOCK 60 - ISOBARS FOR FN=0.220

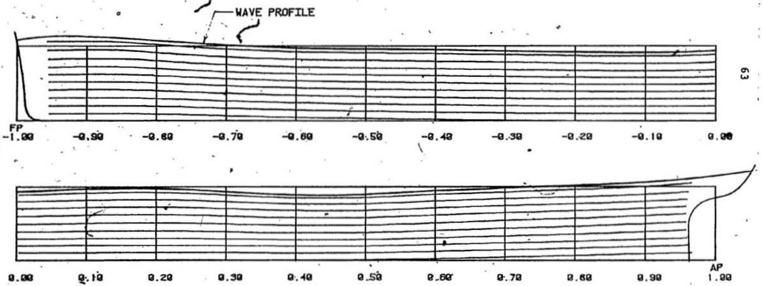


FIGURE 14

SERIES 60 BLOCK 60 - FLOW DIRECTIONS FOR FN=0.220

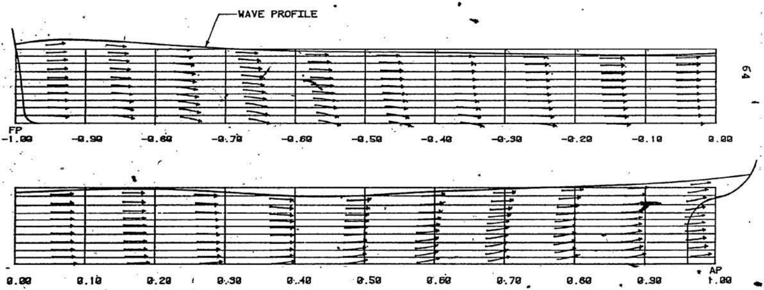


FIGURE 16

SERIES 60 BLOCK 60 - WAVE PROFILE FOR FN=0.280

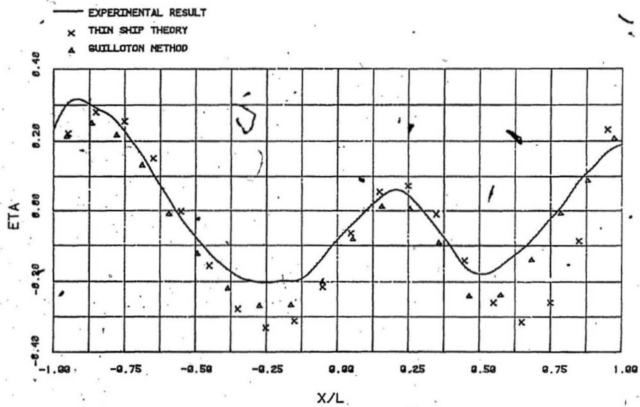


FIGURE 16

SERIES 60 BLOCK 60 - ISOBARS FOR FN=0.280

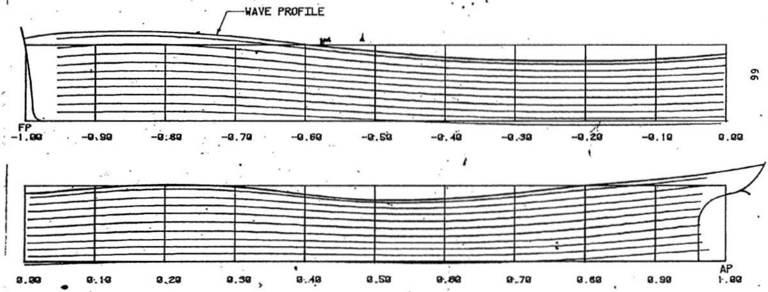


FIGURE 17

SERIES 60 BLOCK 60 - FLOW DIRECTIONS FOR FN=0.280

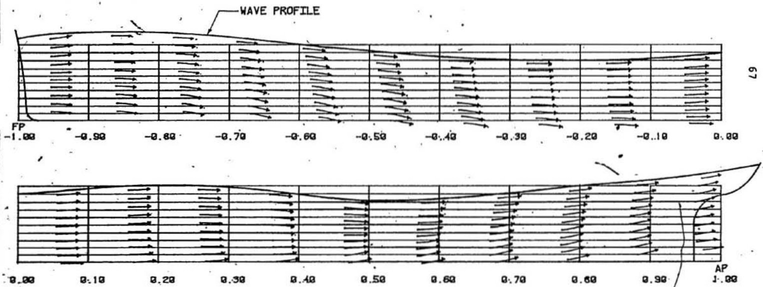


FIGURE 16.

SERIES 60 BLOCK 60 - WAVE PROFILE FOR FN=0.350

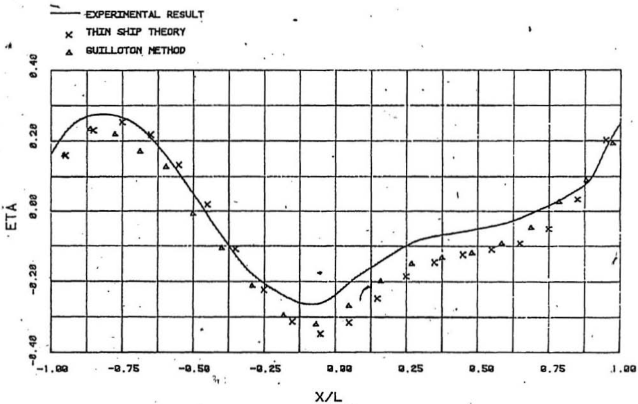


FIGURE 19

SERIES 60 BLOCK 60 - ISOBARS FOR FN=0.350

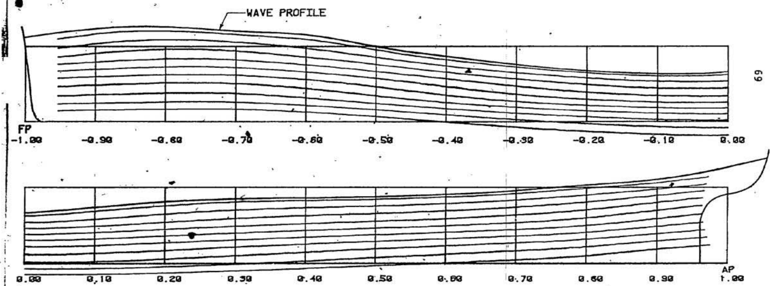


FIGURE 20

SERIES 60 BLOCK 60 - FLOW DIRECTIONS FOR FN=0.350

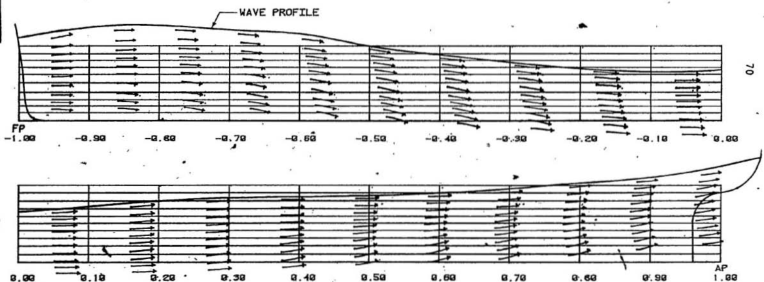
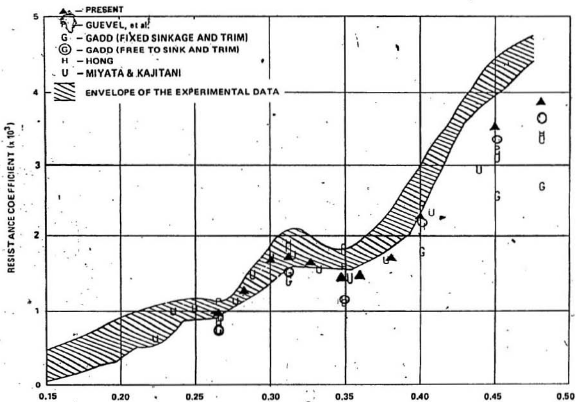


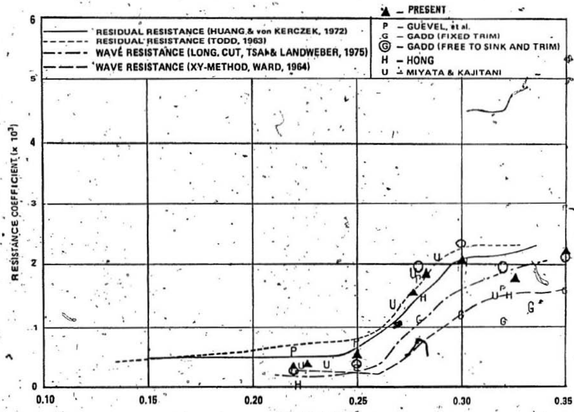
FIGURE 21



FROUDE NUMBER

Wigley Hull - Guilloton's Method

FIGURE 22



Series 60 - Guilloton's Method

FIGURE 23

APPENDIX : COMPUTER PROGRAM

```
C*********************************************************************
C       TITLE OF THE THESIS :
C
C       ANALYSIS OF SHIP FLOW IN AN IDEAL FLUID USING GUILLOTON'S
C       METHOD AND SPLINE FUNCTIONS
C
C       FUNCTION: THIS PROGRAM COMPUTES THE WAVE RESISTANCE COEFF. AND
C                 WAVE PROFILES FROM THE GIVEN SHIP-OFFSETS AND THE FROUDE
C                 NUMBERS BY THIN SHIP THEORY AND GUILLOTON'S METHOD.
C
C       WAVE RESISTANCE COEFFICIENT IS      CW=RW/(0.5*RHO*U*U*WS)
C       NONDIMENSIONAL WAVE ELEVATION IS    ETA=Zw/(U/(2*G))
C
C*********************************************************************
        DIMENSION ETA(20),ETA1(20),FIL1(4),SM(20,10),SM1(20,10)
        COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
       1,XK0,WS,XS,XE
        COMMON /SA/ NFN,FN(20)
        COMMON /SC/ OFY(22,20),XA(22),ZA(20),YX2(22,20),YZ2(22,20)
        COMMON /SD/ SHIP(20),XLL,B,TD,WS1
        COMMON /S/ S(20,10),S1(20,10),S2(20,10),UI(20,10)
        NX=21
        NX1=NX-1
        NZ=11
        NZ1=NZ-1
        XL=2.0
        TYPE *,'ENTER OUTPUT DATAFILE NAME'
        ACCEPT 1,FIL1
1       FORMAT(4A4)
        CALL ASSIGN(2,FIL1)
        CALL INPUT
        CALL POINT
C       TYPE *,'POINT, FINISHED'
        CALL CURFIT(SM)
C---------------------------------------------------------------------
C       OUTPUT: SHIP,SM(I,J)
        WRITE(2,100) SHIP
100     FORMAT(1X,20A4)
        WRITE(2,*) ' '
        WRITE(2,*) 'L=',XLL,'   T=',TD,'   B=',B,'   WS=',WS1
        WRITE(2,*) ' '
        WRITE(2,*) ' SOURCE STRENGTH BY THIN SHIP THEORY'
        WRITE(2,*) ' '
        DO 5 I=1,NX1
          WRITE(2,*) (SM(I,J),J=1,10)
5       CONTINUE
C---------------------------------------------------------------------
C       NFN=1
C       FN(1)=0.220
        DO 10 I=1,NFN
          FR=FN(I)
          XK0=1./(2.*FR*FR)
          DO 20 II=1,NX1
            DO 20 JJ=1,NZ1
              S(II,JJ)=SM(II,JJ)
              S1(II,JJ)=SM(II,JJ)
              S2(II,JJ)=SM(II,JJ)
20        CONTINUE
          CALL GULTON(SM1)
          CALL MICH(SM,CW)
          CALL MICH(SM1,CW1)
          CALL WAVE(SM,SM1,ETA,ETA1)
```

```
C-----------------------------------------------------------------------
C         OUTPUT FR,SH1,CW,CW1,ETA,ETA1
          WRITE(2,40)
          WRITE(2,*) 'SOURCE STRENGTH BY GUILLOTON METHOD FOR FN=',FR
          WRITE(2,*) ' '
          DO 30 II=1,NX1
             WRITE(2,*) (SH1(II,JJ),JJ=1,NZ1)
30        CONTINUE
          WRITE(2,*) ' '
          WRITE(2,*) 'FR=',FR,'   CW=',CW,'   CW1=',CW1
          WRITE(2,*) ' '
          WRITE(2,*) 'CW  : WAVE RESISTANCE COEFF. BY THIN SHIP THEORY'
          WRITE(2,*) 'CW1 : WAVE RESISTANCE COEFF. BY GUILLOTON METHOD'
          WRITE(2,40)
40        FORMAT(1H1)
          WRITE(2,*) 'WAVE PROFILE FOR FN=',FR
          WRITE(2,*) ' '
          DO 50 J=1,NX1
             WRITE(2,*) 'X/L=',XC(J),'   ETA=',ETA(J),'   ETA1=',ETA1(J)
50        CONTINUE
          WRITE(2,*) ' '
          WRITE(2,*) 'ETA  : WAVE PROFILE BY THIN SHIP THOERY'
          WRITE(2,*) 'ETA1 : WAVE PROFILE BY GUILLOTON METHOD'
C-----------------------------------------------------------------------
10    CONTINUE
1000  STOP
      END
      SUBROUTINE POINT
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XX0,WS,XS,XE
      SX=XL/(NX-1)
      SZ=T/(NZ-1)
      DO 10 I=1,NX
         X(I)=-1.+SX*(I-1)
         IF(I.EQ.NX) GO TO 10
         XC(I)=-1+SX*(I-0.5)
10    CONTINUE
      DO 20 I=1,NZ
         Z(I)=-T+SZ*(I-1)
         IF(I.EQ.NZ) GO TO 20
         ZC(I)=-T+SZ*(I-0.5)
20    CONTINUE
      RETURN
      END
      SUBROUTINE INPUT
C***********************************************************************
C     FUNCTION: THIS SUBROUTINE READS INPUT DATA FROM INPUT DATAFILE
C               AND NORMALIZES THE SHIP-OFFSETS.
C     THE NONDIMENSIONL LBP IS 2.    (FROM -1. TO 1.)
C***********************************************************************
      DIMENSION FILE(4)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XX0,WS,XS,XE
      COMMON /SA/ NFN,FN(20)
      COMMON /SB/ NST,OFX(25),OFZ(25,20),OFY(25,20),NOF(25)
      COMMON /SD/ SHIP(20),XLL,B,TD,WS1
      TYPE *,'ENTER OFFSET DATAFILE'
      ACCEPT 1,FILE
1     FORMAT(4A4)
      CALL ASSIGN(1,FILE)
      READ(1,200) SHIP
200   FORMAT(20A4)
```

```
      READ(1,*) NFN
      READ(1,*) (FN(I),I=1,NFN)
      READ(1,*) XLL,B,TD,WS1
      SCALE=XLL/2.
      WS=WS1/(SCALE*SCALE)
      T=TD*2./XLL
      READ(1,*) NST
      DO 10 I=1,NST
         READ(1,*) OFX(I),NOF(I)
         N1=NOF(I)
         DO 20 J=1,N1
            READ(1,*) OFZ(I,J),OFY(I,J)
            OFZ(I,J)=(OFZ(I,J)-TD)*2./XLL
            OFY(I,J)=OFY(I,J)*2./XLL
20       CONTINUE
10    CONTINUE
      CALL CLOSE(1)
      RETURN
      END
      SUBROUTINE CURFIT(SM)
      DIMENSION SM(20,10)
      DIMENSION FILE(4),XX(25),YY(25),Y2(25)
      DIMENSION YXX(25),YX(22,20),OFY(25,20)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XX0,[illegible]
      COMMON [illegible]/ NST,OX(25),OZ(25,20),OY(25,20),N(25),C(25)
      COMMON [illegible]/ OFYY(22,20),XA(22),ZA(20),YX2(22,20),YZ2(22,20)
      NX2=NX+1
      NZ2=NZ+0
      NXX=NX2
      NZZ=NZ2
      XL1=XL
      T1=T
      CALL PNT(XL1,T1,NXX,NZZ)
      DO 30 I=1,NST
         N1=N(I)
         DO 40 J=1,N1
            XX(J)=OZ(I,J)
            YY(J)=OY(I,J)
40       CONTINUE
         CALL CUBIC1(N1,XX,YY,Y2)
         ID=1
         CALL INTRP(ID,NZ2,ZA,N1,XX,YY,Y2,YXX)
         DO 50 J=1,NZ2
            OFY(I,J)=YXX(J)
50       CONTINUE
30    CONTINUE
      DO 60 J=1,NZ2
         DO 70 I=1,NST
            YY(I)=OFY(I,J)
70       CONTINUE
         CALL CUBIC1(NST,OX,YY,Y2)
         ID=1
         CALL INTRP(ID,NX2,XA,NST,OX,YY,Y2,YXX)
         DO 80 I=1,NX2
            OFYY(I,J)=YXX(I)
80       CONTINUE
60    CONTINUE
      DO 90 I=1,NZ2
         DO 100 J=1,NX2
            XX(J)=XA(J)
            YY(J)=OFYY(J,I)
```

```
180       CONTINUE
          CALL CUBIC1(NX2,XX,YY,Y2)
          ID=2
          CALL INTRP(ID,NX2,XX,NX2,XX,YY,Y2,YXX)
          DO 110 J=1,NX2
             YX(J,I)=YXX(J)
110       CONTINUE
90     CONTINUE
       II=0
       DO 120 I=1,NX2
       IF(I.EQ.1) GO TO 120
       IF(I.EQ.NX2) GO TO 120
       II=II+1
       JJ=0
          DO 130 J=1,NZ2
             IF(J.GE.1.AND.J.LE.5) GO TO 130
             IF(J.EQ.7) GO TO 130
             IF(J.GE.NZ2-3) GO TO 130
             JJ=JJ+1
             SH(II,JJ)=2.*YX(I,J)
130       CONTINUE
120    CONTINUE
       DO 140 J=1,NZ2
          DO 150 I=1,NX2
             XX(I)=XA(I)
             YY(I)=OFYY(I,J)
150       CONTINUE
          CALL CUBIC1(NX2,XX,YY,Y2)
          DO 160 I=1,NX2
             YX2(I,J)=Y2(I)
160       CONTINUE
140    CONTINUE
       DO 170 I=1,NX2
          DO 180 J=1,NZ2
             XX(J)=ZA(J)
             YY(J)=OFYY(I,J)
180       CONTINUE
          CALL CUBIC1(NZ2,XX,YY,Y2)
          DO 190 J=1,NZ2
             YZ2(I,J)=Y2(J)
190       CONTINUE
170    CONTINUE
       RETURN
       END
       SUBROUTINE INTRP(ID,N2,ZA,N1,XX,YY,Y2,OUT)
       DIMENSION ZA(N2),XX(N1),YY(N1),Y2(N1),OUT(N2)
       DIMENSION H(20),HY(20),A(20),B(20),C(20),D(20)
       DO 60 II=1,N1-1
          H(II)=XX(II+1)-XX(II)
          HY(II)=YY(II+1)-YY(II)
60     CONTINUE
       DO 70 II=1,N1-1
          A(II)=(Y2(II+1)-Y2(II))/(6.*H(II))
          B(II)=0.5*Y2(II)
          C(II)=HY(II)/H(II)-H(II)*(Y2(II+1)+2.*Y2(II))/6.
          D(II)=YY(II)
70     CONTINUE
       DO 10 I=1,N2
          ZZ=ZA(I)
          DO 20 J=1,N1-1
             IF(ZZ.GE.XX(J).AND.ZZ.LE.XX(J+1)) GO TO 30
20        CONTINUE
```

```
30        DEL=ZZ-XX(J)
          IF(ID.EQ.1) OUT(I)=(((A(J)*DEL)+B(J))*DEL+C(J))*DEL+D(J)
          IF(ID.EQ.2) OUT(I)=((3.*A(J))*DEL+2.*B(J))*DEL+C(J)
10      CONTINUE
        RETURN
        END
        SUBROUTINE PNT(XL,T,NX2,NZ2)
        COMMON /SC/ OFYY(22,20),X(22),Z(20),YX2(22,20),YZ2(22,20)
        SX=XL/(NX2-2)
        SZ=T/(NZ2-10)
        DO 10 I=1,NX2
          J=I-1
          IF(I.EQ.1) X(I)=-1.0
          IF(I.EQ.NX2) X(I)=1.0
          IF(I.NE.1.AND.I.NE.NX2) X(I)=-1.+SX*(J-0.5)
10      CONTINUE
        Z(1)=-T
        Z(2)=Z(1)+SZ/16.
        Z(3)=Z(2)+SZ/16.
        Z(4)=Z(3)+SZ/8.
        Z(5)=Z(4)+SZ/8.
        Z(6)=Z(5)+SZ/8.
        Z(7)=Z(6)+SZ/2.
        Z(8)=Z(7)+SZ/2.
        DO 20 I=9,NZ2
          Z(I)=Z(I-1)+SZ
20      CONTINUE
        RETURN
        END
        SUBROUTINE SULTON(SH1)
C************************************************************************
C       FUNCTION: THIS SUBROUTINE COMPUTES NEW SOURCE STRENGTH BY
C                 SULLOTON'S METHOD.
C************************************************************************
        DIMENSION FILE(4),SH1(20,10)
        COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
       1,XX0,WS,XS,XE
        COMMON /C/ UT(20,10,10),UL(20,10,10)
        COMMON /S/ S(20,10),S1(20,10),S2(20,10),UI(20,10)
        COMMON /SC/ OFY(22,20),XA(22),ZA(20),YX2(22,20),YZ2(22,20)
        XS=-1.0
        XE=1.0
        CALL INDU2
        ID=1
20      TYPE *,'ITERATION',ID
        CALL CHANG
        IR=0
        CALL TRAN(IR,ID)
        IF(IR.NE.1) GO TO 30
        IF(ID.EQ.100) GO TO 30
        ID=ID+1
        GO TO 20
30      DO 40 I=1,NX1
          DO 40 J=1,NZ1
            SH1(I,J)=S1(I,J)
40      CONTINUE
        RETURN
        END
        SUBROUTINE CHANG
        DIMENSION U(20,10)
        COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
       1,XX0,WS,XS,XE
```

```
      COMMON /C/ UT(20,10,10),UL(20,10,10)
      COMMON /S/ S(20,10),S1(20,10),S2(20,10),UI(20,10)
      DO 20 I=1,NZ1
      DO 20 J=1,NX1
         UI(J,I)=0.0
         NO=(I-1)*NX1+J
         DO 30 II=1,NX1
         IS1=NX1-J+II
         IS2=NX1+J-II
         DO 30 JJ=1,NZ1
            NO1=(JJ-1)*NX1+II
            IF(II.LE.J) U(II,JJ)=UT(IS1,JJ,I)
            IF(II.GT.J) U(II,JJ)=-UL(IS2,JJ,I)
30       CONTINUE
         DO 40 II=1,NX1
         DO 40 JJ=1,NZ1
            UI(J,I)=UI(J,I)+U(II,JJ)*S1(II,JJ)
40       CONTINUE
20    CONTINUE
200   RETURN
      END
      SUBROUTINE TRAN(IR,ID)
      DIMENSION Y(20),XM(21),YM(21),ZM(21),UP(20),TN(20),YX1(20)
      DIMENSION YZ1(20),XMM(20,10),ZMM(20,10)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XK0,W0,X0,XE
      COMMON /S/ S(20,10),S1(20,10),S2(20,10),UI(20,10)
      F1(A1,C1)=(1.+A1)/SQRT(1+C1*C1)
      EMAX=0.0
      DO 10 I=1,NZ1
         DO 20 J=1,NX1
            A1=UI(J,I)
            C1=S1(J,I)/2.
            TN(J)=F1(A1,C1)
            Y(J)=TN(J)
20       CONTINUE
         IDX=1
         CALL XMCAL(IDX,Y,XM)
         DO 30 J=1,NX1
            XM(J)=XM(J)-1.
            XMM(J,I)=XM(J)
30       CONTINUE
         DO 40 J=1,NX1
            Y(J)=S1(J,I)/2.
            ZM(J)=ZC(I)-UI(J,I)/XK0
            ZMM(J,I)=ZM(J)
40       CONTINUE
C        CALL XMCAL(IDX,Y,YM)
         CALL PARTL(YX1,YZ1,XM,ZM)
         DO 50 J=1,NX1
            Y(J)=UI(J,I)
50       CONTINUE
         IDX=0
         CALL XMCAL(IDX,Y,UP)
         DO 60 J=1,NX1
            Y(J)=2.*(TN(J)*YX1(J)-UP(J)*YZ1(J)/XK0)
            S1(J,I)=Y(J)
60       CONTINUE
         DO 70 J=1,NX1
            Y(J)=S2(J,I)
70       CONTINUE
         DO 80 J=1,NX1
```

```
            CX=ABS(S2(J,I)/S1(J,I)-1.)
            IF(CX.GT.EMAX) EMAX=CX
            IF(CX.GT.1.E-4) IR=1
            S2(J,I)=S1(J,I)
20       CONTINUE
10    CONTINUE
      TYPE *,'ERRMAX=',EMAX
      IF(IR.EQ.1) GO TO 100
      WRITE(2,*) ' '
      WRITE(2,*) 'ISOBARS'
      WRITE(2,*) ' '
      DO 110 I=1,NZ1
         WRITE(2,*) ' '
         WRITE(2,*) 'No. ',I
         WRITE(2,*) ' '
         WRITE(2,*) 'X-COOR.'
         WRITE(2,*) (XMM(J,I),J=1,NX1)
         WRITE(2,*) ' '
         WRITE(2,*) 'Z-COOR.'
         WRITE(2,*) (ZMM(J,I),J=1,NX1)
110   CONTINUE
100   RETURN
      END
      SUBROUTINE PARTL(YX1,YZ1,XM,ZM)
      DIMENSION YX1(20),YZ1(20),YT(22),YT2(22)
      DIMENSION XM(21),ZM(21)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,Y,T,X(21),Z(11),XC(20),ZC(10)
     1,XX0,WS,XS,XE
      COMMON /SC/ OFY(22,20),XX(22),ZZ(20),YX2(22,20),YZ2(22,20)
      DO 30 I=1,NX1
         XI=XM(I)
         ZI=ZM(I)
         DO 40 J=1,NX2-1
            IF(XI.GE.XX(J).AND.XI.LE.XX(J+1)) GO TO 50
40       CONTINUE
         TYPE *,'OUT OF RANGE XX=',XI
         IF(XI.GT.XX(NX2)) J=NX2-1
         IF(XI.GT.XX(NX2)) DELX=XX(NX2)-XX(NX2-1)
         IF(XI.LT.XX(1)) J=1
         IF(XI.LT.XX(1)) DELX=0.0
         YZ1(I)=1.E-10
         GO TO 125
50       DELX=XI-XX(J)
42       IS1=J
         IE1=J+1
         H1=XX(IE1)-XX(IS1)
         DO 90 J=1,NZ2
70          Y2=OFY(IE1,J)
            Y1=OFY(IS1,J)
            Y22=YX2(IE1,J)
            Y12=YX2(IS1,J)
60             HY=Y2-Y1
               A=(Y22-Y12)/(6.*H1)
               B=0.5*Y12
               C=HY/H1-H1*(Y22+2.*Y12)/6.
               D=Y1
               YT(J)=(((A*DELX)+B)*DELX+C)*DELX+D
90          CONTINUE
          CALL CUBIC1(NZ2,ZZ,YT,YT2)
          DO 100 J=1,NZ2-1
            IF(ZI.GE.ZZ(J).AND.ZI.LE.ZZ(J+1)) GO TO 110
100      CONTINUE
```

```
      IF(ZI.GT.ZZ(NZ2)) J=NZ2-1
      IF(ZI.GT.ZZ(NZ2)) DELZ=ZZ(NZ2)-ZZ(NZ2-1)
      IF(ZI.LT.ZZ(1)) YZ1(I)=1.E-10
      IF(ZI.LT.ZZ(1)) GO TO 125
110   DELZ=ZI-ZZ(J)
120   IS=J
      IE=J+1
      H=ZZ(IE)-ZZ(IS)
      HY=YT(IE)-YT(IS)
      A=(YT2(IE)-YT2(IS))/(6.*H)
      B=0.5*YT2(IS)
      C=HY/H-H*(YT2(IE)+2.*YT2(IS))/6.
      D=YT(IS)
      YZ1(I)=((3.*A)*DELZ+2.*B)*DELZ+C
125   DO 140 J=1,NZ2-1
         IF(ZI.GE.ZZ(J).AND.ZI.LE.ZZ(J+1)) GO TO 150
140   CONTINUE
      IF(ZI.GT.ZZ(NZ2)) J=NZ2-1
      IF(ZI.GT.ZZ(NZ2)) DELZ=ZZ(NZ2)-ZZ(NZ2-1)
      IF(ZI.LT.ZZ(1)) YX1(I)=1.E-10
      IF(ZI.LT.ZZ(1)) GO TO 30
150   DELZ=ZI-ZZ(J)
220   IS1=J
      IE1=J+1
      H1=ZZ(IE1)-ZZ(IS1)
      DO 190 J=1,NX2
         Y2=OFY(J,IE1)
         Y1=OFY(J,IS1)
         Y22=YZ2(J,IE1)
         Y21=YZ2(J,IS1)
         HY=Y2-Y1
         A=(Y22-Y12)/(6.*H1)
         B=0.5*Y12
         C=HY/H1-H1*(Y22+2.*Y12)/6.
         D=Y1
         YT(J)=(((A*DELZ)+B)*DELZ+C)*DELZ+D
190   CONTINUE
      CALL CUBIC1(NX2,XX,YT,YT2)
      DO 200 J=1,NX2-1
         IF(XI.GE.XX(J).AND.XI.LE.XX(J+1)) GO TO 210
200   CONTINUE
      TYPE *,'OUT OF RANGE XX=',XI
      IF(XI.GT.XX(NX2)) J=NX2-1
      IF(XI.GT.XX(NX2)) DELX=XX(NX2)-XX(NX2-1)
      IF(XI.LT.XX(1)) J=1
      IF(XI.LT.XX(1)) DELX=0.0
      YX1(I)=1.E-10
      GO TO 30
C     GO TO 211
210   DELX=XI-XX(J)
211   IS=J
      IE=J+1
      H=XX(IE)-XX(IS)
      HY=YT(IE)-YT(IS)
      A=(YT2(IE)-YT2(IS))/(6.*H)
      B=0.5*YT2(IS)
      C=HY/H-H*(YT2(IE)+2.*YT2(IS))/6.
      D=YT(IS)
      YX1(I)=((3.*A)*DELX+2.*B)*DELX+C
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XMCAL(IDX,Y,XM)
      DIMENSION H(20),HY(20),Y(20),Y2(20),XM(21)
      COMMON /A/ NX,N1,N2,NZ,NZ1,NZ2,XL,T,X1(21),Z1(11),X(20),Z(10)
     1,XX0,WS,XS,XE
C     TYPE *,'X'
C     TYPE *,X
      NN=N1-1
      DO 10 I=1,NN
         I1=I+1
         H(I)=X(I1)-X(I)
         HY(I)=Y(I1)-Y(I)
10    CONTINUE
      CALL CUBIC1(N1,X,Y,Y2)
      S1=HY(1)/H(1)-H(1)*(Y2(2)+2.*Y2(1))/6.
      DH=X(1)-XS
      DH1=X(1)*X(1)-XS*XS
      IF(IDX.NE.0) XM(1)=S1*DH1/2.+(Y(1)-S1*X(1))*DH
      DO 20 I=2,N1
         I1=I-1
         A=(Y2(I1+1)-Y2(I1))/(6.*H(I1))
         B=0.5*Y2(I1)
         C=HY(I1)/H(I1)-H(I1)*(Y2(I1+1)+2.*Y2(I1))/6.
         D=Y(I1)
         H1=H(I1)
         IF(IDX.EQ.0) XM(I1)=C
         IF(IDX.NE.0) XM(I)=XM(I1)+((((A/4.)*H1+B/3.)*H1+C/2.)*H1
     1+D)*H1
20    CONTINUE
      DH=XE-X(N1)
      S2=((3.*A)*DH+2.*B)*DH+C
      IF(IDX.EQ.0) XM(N1)=S2
      DH1=XE*XE-X(N1)*X(N1)
      IF(IDX.NE.0) XM(N1+1)=XM(N1)+S2*DH1/2.+(Y(N1)-S2*X(N1))*DH
      RETURN
      END
      SUBROUTINE INDU2
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XX0,WS,XS,XE
      COMMON /C/ UT(20,10,10),UL(20,10,10)
      COMMON /D/ V1(20,10),V2(20,10),V3(20,10)
      Y=0.0
      XX=XC(NX1)
      DO 20 L=1,NZ1
      ZZ=ZC(L)
      CALL SU1(XX,Y,ZZ)
      CALL SU2(XX,Y,ZZ)
      CALL SU3(XX,Y,ZZ)
      DO 30 JJ=1,NZ1
         DO 30 II=1,NX1
            UL(II,JJ,L)=V1(II,JJ)+V2(II,JJ)
            UT(II,JJ,L)=UL(II,JJ,L)+V3(II,JJ)
30       CONTINUE
20    CONTINUE
      RETURN
      END
      SUBROUTINE MICH(SS,CW)
C***********************************************************************
C     FUNCTION: THIS SUBROUTINE CALCULATES THE WAVE RESISTANCE COEFF.
C               BY MICHELL INTEGRAL.
C***********************************************************************
      DIMENSION SS(20,10)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
```

```
      1,XK0,WS,XS,XE
      COMMON /SM/ SM(20,10)
      EXTERNAL RT
      DO 1 I=1,NX1
         DO 1 J=1,NZ1
            SM(I,J)=SS(I,J)
1     CONTINUE
      ST=0.0
      EN=89.995*3.1416/180.
      CALL SIMPSN(ST,EN,RT,VAL)
      CC=2./(WS*3.1416*XK0*XK0)
      CW=VAL*CC
      RETURN
      END
      FUNCTION RT(THETA)
      DIMENSION XM(20),YM(20),ZM(10),TEMP(20)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
      1,XK0,WS,XS,XE
      COMMON /SM/ SM(20,10)
      F1=COS(THETA)
      B=XK0/F1
      A=B/F1
      ZN=Z(NZ)*A
      D=EXP(ZN)
      DO 10 I=1,NZ1
         K=NZ1+1-I
         T=A*Z(K)
         D1=EXP(T)
         ZM(K)=D-D1
         D=D1
10    CONTINUE
      XN=X(NX)*B
      D=SIN(XN)
      E=COS(XN)
      DO 20 I=1,NX1
         K=NX1+1-I
         T=B*X(K)
         D1=SIN(T)
         E1=COS(T)
         XM(K)=D-D1
         YM(K)=E-E1
         D=D1
         E=E1
20    CONTINUE
      DO 30 I=1,NX1
         TEMP(I)=0.0
         DO 40 J=1,NZ1
            TEMP(I)=TEMP(I)+SM(I,J)*ZM(J)
40       CONTINUE
30    CONTINUE
      PS=0.0
      QS=0.0
      DO 50 I=1,NX1
         PS=PS+TEMP(I)*XM(I)
         QS=QS+TEMP(I)*YM(I)
50    CONTINUE
      RT=(PS*PS+QS*QS)*F1*F1*F1
      RETURN
      END
      SUBROUTINE WAVE(SM,SM1,ETA,ETA1)
      DIMENSION U(20,10),ETA(20),SM(20,10),ETA1(20),SM1(20,10)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
```

```
      1,XK0,WS,XS,XE
      COMMON /D/ V1(20,10),V2(20,10),V3(20,10)
      XX=XC(NX1)
      YY=0.0
      ZZ=0.0
      CALL SU1(XX,YY,ZZ)
      CALL SU2(XX,YY,ZZ)
      CALL SU3(XX,YY,ZZ)
      DO 10 I=1,NX1
      DO 10 J=1,NZ1
         V1(I,J)=V1(I,J)+V2(I,J)
         V3(I,J)=V1(I,J)+V3(I,J)
10    CONTINUE
      DO 20 J=1,NX1
         ETA(J)=0.0
         DO 30 II=1,NX1
            IS1=NX1-J+II
            IS2=NX1+J-II
            DO 30 JJ=1,NZ1
               IF(II.LE.J) U(II,JJ)=V3(IS1,JJ)
               IF(II.GT.J) U(II,JJ)=-V1(IS2,JJ)
30       CONTINUE
         DO 40 II=1,NX1
            DO 40 JJ=1,NZ1
               ETA(J)=ETA(J)+U(II,JJ)*SM(II,JJ)
40       CONTINUE
20    CONTINUE
      DO 50 I=1,NX1
         ETA(I)=-2.*ETA(I)
50    CONTINUE
      DO 120 J=1,NX1
         ETA1(J)=0.0
         DO 130 II=1,NX1
            IS1=NX1-J+II
            IS2=NX1+J-II
            DO 130 JJ=1,NZ1
               IF(II.LE.J) U(II,JJ)=V3(IS1,JJ)
               IF(II.GT.J) U(II,JJ)=-V1(IS2,JJ)
130      CONTINUE
         DO 140 II=1,NX1
            DO 140 JJ=1,NZ1
               ETA1(J)=ETA1(J)+U(II,JJ)*SM1(II,JJ)
140      CONTINUE
120   CONTINUE
      DO 150 I=1,NX1
         ETA1(I)=-2.*ETA1(I)
150   CONTINUE
      RETURN
      END
      SUBROUTINE SU1(XX,Y,ZZ)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
      1,XK0,WS,XS,XE
      COMMON /D/ V1(20,10),V2(20,10),V3(20,10)
      COMMON /F/ F(21,11)
      DOUBLE PRECISION F1,F
      DO 10 J=1,NZ
         Z1=Z(J)
         DO 10 I=1,NX
            X1=X(I)
            CALL COEF1(XX,Y,ZZ,X1,Z1,XK0,F1)
            F(I,J)=F1
10    CONTINUE
```

```
      CC=1./(4.*3.1416)
      DO 30 J=1,NZ1
         J1=J+1
         DO 30 I=1,NX1
            I1=I+1
            P=F(I1,J1)+F(I,J)-F(I,J1)-F(I1,J)
            V1(I,J)=P*CC
30    CONTINUE
      RETURN
      END
      SUBROUTINE COEF1(XX,Y,ZZ,X1,Z1,XK0,F1)
      DOUBLE PRECISION R1,R2,F1,D1,D2,XY,YZ
      DOUBLE PRECISION DX1,C1,C2
      DX1=X1-XX
      C1=Z1-ZZ
      C2=Z1+ZZ
      IF(C1.EQ.0.0) C1=-1.E-4
      IF(C2.EQ.0.0) C2=-1.E-4
      XY=DX1*DX1+Y*Y
      R1=SQRT(XY+C1*C1)
      R2=SQRT(XY+C2*C2)
      IF(XY.LE.1.E-10) F1=DLOG(ABS(C2)/ABS(C1))
      D1=C1+R1
      D2=C2+R2
      IF(XY.GT.1.E-10) F1=DLOG(D1/D2)
      RETURN
      END
      SUBROUTINE SU2(XX,Y,ZZ)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XK0,WS,XS,XE
      COMMON /D/ V1(20,10),V2(20,10),V3(20,10)
      COMMON /F/ F(21,11)
      DO 10 J=1,NZ
         Z1=Z(J)
         DO 10 I=1,NX
C           TYPE *,'SU2    J=',J,'    I=',I
            X1=X(I)
            CALL COEF(XX,Y,ZZ,X1,Z1,XK0,F1)
            F(I,J)=F1
10    CONTINUE
      CC=-1./(4.*3.1416*3.1416)
      DO 30 J=1,NZ1
         J1=J+1
         DO 30 I=1,NX1
            I1=I+1
            Q=F(I1,J1)+F(I,J)-F(I,J1)-F(I1,J)
            V2(I,J)=Q*CC
30    CONTINUE
      RETURN
      END
      SUBROUTINE COEF(XX,Y,ZZ,X1,Z1,XK0,F1)
      DIMENSION SF1(3),SF2(3)
      COMMON /A1/ Y1,XK,DX1,C1
      Y1=Y
      XK=XK0
      DX1=XX-X1
      C1=ZZ+Z1
      Y1=Y
      XK=XK0
      ST=0.0
      EN=89.995*3.1416/180.
      N1=50
```

```
      D1=(EN-ST)/N1
      DO 10 J=1,3
         SF1(J)=0.0
10    CONTINUE
      DO 20 I=1,N1+1
         THETA=ST+D1*(I-1)
         CALL COM(THETA,F)
         II=I/2*2
         IF(I.EQ.1.OR.I.EQ.N1+1) IC=1
         IF(II.EQ.I) IC=2
         IF(II.NE.I.AND.I.NE.1.AND.I.NE.N1+1) IC=3
         SF1(IC)=SF1(IC)+F
20    CONTINUE
      FSUM1=(SF1(1)+4.*SF1(2)+2.*SF1(3))*D1/3.
      FSUM2=FSUM1
      ICC=1
      IF(ICC.EQ.1) GO TO 110
      K=1
45    K=K+1
      SF2(1)=SF1(1)
      SF2(3)=SF1(2)+SF1(3)
      N2=N1
      D2=D1/2.
      SF2(2)=0.0
      DO 70 I=1,N2
         THETA=ST+D2*(2*I-1)
         CALL COM(THETA,F)
         SF2(2)=SF2(2)+F
70    CONTINUE
      FSUM2=(SF2(1)+4.*SF2(2)+2.*SF2(3))*D2/3.
      IF(FSUM1.EQ.0.0) CXF=0.0
      IF(FSUM1.NE.0.0) CXF=ABS(FSUM2/FSUM1-1.)
      IF(CXF.LE.1.E-3) GO TO 110
      FSUM1=FSUM2
      DO 100 J=1,3
         SF1(J)=SF2(J)
100   CONTINUE
      N1=2.*N2
      D1=D2
      IF(K.GT.7) GO TO 110
      GO TO 45
C
110   F1=FSUM2
      RETURN
      END
      SUBROUTINE COM(THETA,YF)
      DIMENSION B(2)
      COMMON /A1/ Y,XK0,DX1,C1
      T=TAN(THETA)
      FF1=COS(THETA)
      CO2=FF1*FF1
      F2=SIN(THETA)
      F3=Y*F2
      A=XK0/CO2
      B(1)=DX1*FF1+F3
      B(2)=DX1*FF1-F3
      XT=C1*A
      YF=0.0
      IF(ABS(Y).LE.1.E-10) IS=1
      IF(ABS(Y).GT.1.E-10) IS=2
      DO 20 I=1,IS
         YT=B(I)*A
```

```
      FS=XT*XT+YT*YT
      GS=XT+0.02*XT*XT+0.02*YT*YT
      IF(FS.GT.1.AND.GS.GT.0.) CALL SIMP(XT,YT,VAL,VAL1)
      IF(FS.LE.1.OR.GS.LE.0.) CALL SCON(XT,YT,VAL,VAL1)
      YF1=2.*VAL+ALOG(C1*C1+B(I)*B(I))
      YF=YF+YF1
20    CONTINUE
      IF(IS.EQ.1) YF=2.*YF
      RETURN
      END
      SUBROUTINE SIMP(TC1,TB1,VALR,VALI)
      DIMENSION S1(3),S2(3),SA1(3),SA2(3),CX(2)
      N1=20
      D1=3.1416/(2.*N1)
      K=1
      DO 10 I=1,3
        S1(I)=0.0
        SA1(I)=0.0
10    CONTINUE
      DO 20 I=1,N1+1
        ALPHA=D1*(I-1)
        IF(I.EQ.N1+1) ALPHA=89.995*3.1416/180.
        T=TAN(ALPHA)
        CS=COS(ALPHA)
        A1=T+TC1
        XI1=EXP(-T)/(CS*CS)
        XIR=A1*XI1
        XII=TB1*XI1
        XI2=(A1*A1)+(TB1*TB1)
        VS=XIR/XI2
        VS1=XII/XI2
        II=I/2*2
        IF(I.EQ.1.OR.I.EQ.N1+1) IC=1
        IF(II.EQ.I) IC=2
        IF(II.NE.I.AND.I.NE.1.AND.I.NE.N1+1) IC=3
        S1(IC)=S1(IC)+VS
        SA1(IC)=SA1(IC)+VS1
20    CONTINUE
      VALT=(S1(1)+4.*S1(2)+2.*S1(3))*D1/3.
      VA1=(SA1(1)+4.*SA1(2)+2.*SA1(3))*D1/3.
45    K=K+1
      S2(1)=S1(1)
      S2(3)=S1(2)+S1(3)
      SA2(1)=SA1(1)
      SA2(3)=SA1(2)+SA1(3)
      N2=N1
      D2=D1/2.
      S2(2)=0.0
      SA2(2)=0.0
      DO 70 I=1,N2
        ALPHA=D2*(2*I-1)
        T=TAN(ALPHA)
        CS=COS(ALPHA)
        A1=T+TC1
        XI1=EXP(-T)/(CS*CS)
        XIR=A1*XI1
        XII=TB1*XI1
        XI2=(A1*A1)+(TB1*TB1)
        VS=XIR/XI2
        VS1=XII/XI2
        S2(2)=S2(2)+VS
        SA2(2)=SA2(2)+VS1
```

```
70    CONTINUE
      VAL2=(S2(1)+4.*S2(2)+2.*S2(3))*D2/3.
      VA2=(SA2(1)+4.*SA2(2)+2.*SA2(3))*D2/3.
      IF(VAL1.EQ.0.0) CX(1)=0.0
      IF(VAL1.NE.0.0) CX(1)=ABS(VAL2/VAL1-1.)
      IF(VA1.EQ.0.0) CX(2)=0.0
      IF(VA1.NE.0.0) CX(2)=ABS(VA2/VA1-1.)
      IF(CX(1).LE.1.E-3.AND.CX(2).LE.1.E-3) GO TO 110
      VAL1=VAL2
      VA1=VA2
      DO 100 J=1,3
         S1(J)=S2(J)
         SA1(J)=SA2(J)
100   CONTINUE
      N1=2*N2
      D1=D2
      IF(K.GT.10) GO TO 110
      GO TO 45
C
110   VALR=VAL2
      VALI=-VA2
      RETURN
      END
      SUBROUTINE SCOM(XT,YT,VALR,VALI)
      COMPLEX*8 Z,VN,Z1
      VNR=0.0
      VNI=0.0
      Z=CMPLX(XT,YT)
      R=ABS(Z)
      IF(XT.NE.0.0) TH=ATAN(ABS(YT/XT))
      IF(XT.EQ.0.0) TH=3.1416/2.
      IF(XT.GE.0.0.AND.YT.GT.0.0) TH=TH+0.0
      IF(XT.GE.0.0.AND.YT.LT.0.0) TH=-TH
      IF(XT.LT.0.0.AND.YT.GT.0.0) TH=3.1416-TH
      IF(XT.LT.0.0.AND.YT.LT.0.0) TH=-3.1416+TH
      IF(YT.EQ.0.0) GO TO 10
      Z1=-Z
      VN=-0.5772157-CMPLX(ALOG(R),TH)-Z1
      DO 20 I=2,1000
         XN=FLOAT(I)
         Z1=(XN-1)*Z1*(-Z)/(XN*XN)
         VN=VN-Z1
         ARL=REAL(VN)
         AIM=AIMAG(VN)
         IF(ARL.EQ.0.0.AND.AIM.EQ.0.0) GO TO 25
         BRL=ABS(1.-VNR/ARL)
         BIM=ABS(1.-VNI/AIM)
         IF(ABS(BRL).LE.1.E-5.AND.ABS(BIM).LE.1.E-5) VN=VN*EXP(Z)
         IF(ABS(BRL).LE.1.E-5.AND.ABS(BIM).LE.1.E-5) GO TO 100
25       VNR=ARL
         VNI=AIM
20    CONTINUE
      VN=VN*EXP(Z)
      WRITE(5,*)'UNCONVERGENCE Z=',Z
      GO TO 100
10    X1=ABS(XT)
      VN=-0.5772157-ALOG(X1)-X1
      DO 30 I=2,1000
         XN=FLOAT(I)
         X1=(XN-1)*X1*ABS(XT)/(XN*XN)
         VN=VN-X1
         ARL=REAL(VN)
```

```
      AIM=AIMAG(VN)
      BRL=ABS(1.-ARL/VNR)
      BIM=ABS(1.-AIM/VNI)
      IF(ABS(BRL).LE.1.E-3.AND.ABS(BIM).LE.1.E-3) VN=VN*EXP(Z)
      IF(ABS(BRL).LE.1.E-3.AND.ABS(BIM).LE.1.E-3) GO TO 100
      VNR=ARL
      VNI=AIM
30    CONTINUE
      TYPE *,'UNCONVERGENCE X=',X
      VN=VN*EXP(Z)
100   VALR=REAL(VN)
      VALI=AIMAG(VN)
      RETURN
      END
      SUBROUTINE SU3(XX,Y,ZZ)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,XK0,WS,XS,XE
      COMMON /D/ V1(20,10),V2(20,10),V3(20,10)
      COMMON /F/ F(21,11)
      COMMON /A1/ Y1,XK,DX1,C1,II
      Y1=Y
      XK=XK0
      DO 10 J=1,NZ
       Z1=Z(J)
       DO 10 I=1,NX
C       TYPE *,'SU3   J=',J,'   I=',I
        X1=X(I)
        DX1=XX-X1
        C1=ZZ+Z1
        IF(ABS(C1).LE.1.E-6) C1=-1.E-4
        IF(ABS(Y).GT.1.E-10) CALL INT1(F1)
        IF(ABS(Y).LE.1.E-10) CALL INT2(F1)
        F(I,J)=F1
10    CONTINUE
      CC=1./3.1416
      DO 30 J=1,NZ1
       J1=J+1
       DO 30 I=1,NX1
        I1=I+1
        P=F(I1,J1)+F(I,J)-F(I,J1)-F(I1,J)
        V3(I,J)=P*CC
40      CONTINUE
30    CONTINUE
      RETURN
      END
      SUBROUTINE INT2(F1)
      COMMON /A1/ Y,XK0,DX1,C1,II
      EXTERNAL FU
      IF(DX1.LE.0.0) DX1=0.0
      SUM=0.0
      SUM1=0.0
      TH1=0.0
      DO 10 I=1,1000
       SC=FLOAT(I)
       S=XK0*DX1/(SC*3.1416)
       IF(S.GT.1) GO TO 10
       TH2=ACOS(S)
       IF(TH2.GT.3.1416/2.) TH2=3.1416/2.
       CALL SIMPSN(TH1,TH2,FU,VAL)
       SUM=SUM+VAL
       IF(TH2.GE.3.1416/2.) GO TO 20
       IF(SUM.EQ.0.0) CX=0.0
```

```
      IF(SUM.NE.0.0) CX=ABS(SUM1/SUM-1.)
      IF(CX.LE.1.E-3) GO TO 20
      SUM1=SUM
      TH1=TH2
10    CONTINUE
20    ANG=TH2/3.1416*180.
      F1=SUM*2.
      RETURN
      END
      SUBROUTINE INT1(F1)
      DIMENSION T(10)
      COMMON /A1/ Y,XK0,DX1,C1,II
      EXTERNAL FU1
      F(X)=X*180./3.1416
      GG(A,B,THETA)=A*COS(THETA)-B-TAN(THETA)
      PI=3.1416
      B=DX1/Y
      R=SQRT(DX1*DX1+Y*Y)
      ALPHA=ACOS(DX1/R)
      AA=F(TH1)
      THF=PI/2.
      F1=0.0
      TH1=-PI/2.+ALPHA
      IF(TH1.GT.0.0) GO TO 20
      TT=F(TH1)
      SUM1=0.0
      N=0
      THETA=-89.*3.1416/180.
10    N=N+1
      BB=FLOAT(N)
      A=BB*PI/(XK0*Y)
5     TH2=ATAN(A*COS(THETA)-B)
      IF(ABS(THETA/TH2-1.).LE.1.E-2) TS=(THETA+TH2)/2.
      IF(ABS(THETA/TH2-1.).LE.1.E-2) CALL STACK(N,TS,T)
      IF(ABS(THETA/TH2-1.).LE.1.E-2) GO TO 10
      THETA=(TH2+THETA)/2.
      IF(THETA.LT.0.0) GO TO 5
      N=N-1
      IF(N.EQ.0) N=1
      IF(N.GT.10) NN=10
      IF(N.LE.10) NN=N
      AU=0.0
      DO 15 I=1,NN
        N1=NN-I+1
        AL=T(N1)
        IF(AL.LT.TH1) AL=TH1
        IF(N.EQ.1) AL=TH1
        CALL SIMPSN(AL,AU,FU1,VAL)
        A1=F(AU)
        A2=F(AL)
        F1=F1+VAL
        IF(ABS(AL-TH1).LE.1.E-6) GO TO 20
        IF(F1.EQ.0.0) CX=0.0
        IF(F1.NE.0.0) CX=ABS(SUM1/F1-1.)
        IF(CX.LE.1.E-4) GO TO 20
        SUM1=F1
        AU=AL
15    CONTINUE
20    IF(TH1.LT.0.0) TH1=0.0
      XL=TH1
23    XU=89.99*3.1416/180.
      BB=FLOAT(N)
```

```
      A=BB*PI/(XX0*Y)
25    XM=(XL+XU)/2.
      YL=GG(A,B,XL)
      YU=GG(A,B,XU)
      YM=GG(A,B,XM)
      IF(YM*YL.GE.0.0) XL=XM
      IF(YM*YL.LT.0.0) XU=XM
      T1=F(XL)
      T2=F(XU)
      IF(ABS(XU-XL).LE.1.E-6) TH2=(XL+XU)/2.
      IF(ABS(XU-XL).LE.1.E-6) GO TO 30
      GO TO 25
30    IF(TH2.LT.TH1) GO TO 50
      CALL SIMPSN(TH1,TH2,FU1,VAL)
      A1=F(TH1)
      A2=F(TH2)
      F1=F1+VAL
      IF(TH1.GT.F(80).AND.ABS(VAL).LE.1.E-20) GO TO 70
      IF(F1.EQ.0.0) GO TO 55
      IF(F1.NE.0.0) CX=ABS(SUM1/F1-1.)
      IF(CX.LE.1.E-4) GO TO 70
      SUM1=F1
55    TH1=TH2
50    N=N+1
      XL=TH1
      IF(N.LE.1000) GO TO 25
70    RETURN
      END
      SUBROUTINE STACK(N,TS,T)
      DIMENSION T(10)
      IF(N.LE.10) GO TO 10
      DO 20 I=1,9
        I1=I+1
        T(I)=T(I1)
20    CONTINUE
      T(10)=TS
      GO TO 30
10    T(N)=TS
30    RETURN
      END
      SUBROUTINE SIMPSN(A,B,F,VAL)
      DIMENSION S1(3),S2(3)
      N1=10
      D1=(B-A)/N1
      K=1
      DO 10 I=1,3
        S1(I)=0.0
10    CONTINUE
      DO 20 I=1,N1+1
        ALPHA=A+D1*(I-1)
        VS=F(ALPHA)
        II=I/2*2
        IF(I.EQ.1.OR.I.EQ.N1+1) IC=1
        IF(II.EQ.I) IC=2
        IF(II.NE.I.AND.I.NE.1.AND.I.NE.N1+1) IC=3
        S1(IC)=S1(IC)+VS
20    CONTINUE
      VAL1=(S1(1)+4.*S1(2)+2.*S1(3))*D1/3.
45    K=K+1
      S2(1)=S1(1)
      S2(3)=S1(2)+S1(3)
      N2=N1
```

```
      D2=D1/2.
      S2(2)=0.0
      DO 70 I=1,N2
         ALPHA=A+D2*(2*I-1)
         VS=F(ALPHA)
         S2(2)=S2(2)+VS
70    CONTINUE
      VAL2=(S2(1)+4.*S2(2)+2.*S2(3))*D2/3.
      IF(VAL1.EQ.0.0) CX=0.0
      IF(VAL1.NE.0.0) CX=ABS(VAL2/VAL1-1.)
      IF(CX.LE.1.E-4) GO TO 110
      VAL1=VAL2
      DO 100 J=1,3
         S1(J)=S2(J)
100   CONTINUE
      N1=2*N2
      D1=D2
      IF(K.GT.10) GO TO 110
      GO TO 45
C
110   VAL=VAL2
200   RETURN
      END
      FUNCTION FU(THETA)
      DOUBLE PRECISION XI1,F1,F2,B1,B2,A
      COMMON /A1/ Y,XK0,DX1,C1,I
      F1=COS(THETA)
      F2=SIN(THETA)
      A=XK0/(F1*F1)
      XI1=EXP(A*C1)
      B1=XK0*DX1/F1
      FU=XI1*SIN(B1)
      RETURN
      END
      FUNCTION FU1(THETA)
      COMMON /A1/ Y,XK0,DX1,C1,I
      F1=COS(THETA)
      F2=SIN(THETA)
      A=XK0/(F1*F1)
      XI1=EXP(A*C1)
      B1=A*(DX1*F1+Y*F2)
      FU1=XI1*SIN(B1)
      RETURN
      END
      SUBROUTINE CUBIC1(N,X,Y,Y2)
      DIMENSION X(N),Y(N),Y2(N),F(30),G(30)
      Y2(1)=0.0
      Y2(N)=0.0
      N1=N-1
      G(1)=0.
      F(1)=0.
      DO 2 K=1,N1
         J2=K+1
         H2=X(J2)-X(K)
         R2=(Y(J2)-Y(K))/H2
         IF(K.EQ.1) GO TO 1
         Z=1./(2.*(H1+H2)-H1*G(J1))
         G(K)=Z*H2
         H=6.*(R2-R1)
         IF(K.EQ.2) H=H-H1*Y2(1)
         IF(K.EQ.N1) H=H-H2*Y2(N)
         F(K)=Z*(H-H1*F(J1))
```

```
1          J1=K
           H1=H2
           R1=R2
2        CONTINUE
         Y2(N1)=F(N1)
         IF(N1.LE.2) RETURN
         N2=N1-1
         DO 3 J1=2,N2
           K=N-J1
           Y2(K)=F(K)-G(K)*Y2(K+1)
3        CONTINUE
         RETURN
         END
```

```
C***********************************************************************
C
C       PROGRAM VLCTY
C
C       THIS PROGRAM CALCULATES THE NONDIMENSIONAL VELOCITY COMPONENTS
C     ALONG THE ISOBARS WHICH ARE IN THE OUTPUT DATA OF PROGRAM "THNGUL".
C
C     INPUT DATA:
C     (1) FROUDE NUMBER, DRAFT AND LBP OF THE SHIP
C     (2) SOURCE STRENGTH OF EACH SOURCE PANEL WHICH IS ALSO IN THE OUTPUT
C         DATA OF PROGRAM "THNGUL".
C
C***********************************************************************
C
      DIMENSION FILE(4),UU(20),U(20,10),WW(20),SM(20,10),FIL1(4)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,FR,VEL,XS,XE,XK0
      COMMON /D/ V1(20,10,3),V2(20,10,3),V3(20,10,3)
      TYPE *,'ENTER OUTPUTFILE NAME'
      ACCEPT 1,FILE
1     FORMAT(4A4)
      CALL ASSIGN(1,FILE)
      TYPE *,'ENTER INPUTFILE NAME'
      ACCEPT 1,FIL1
      CALL ASSIGN(2,FIL1)
      XL=2.0
      NX=21
      NZ=11
      NX1=NX-1
      NZ1=NZ-1
      READ(2,*) FR,TD,XLL
      DO 3 I=1,NX1
         READ(2,*) (SM(I,J),J=1,NZ1)
3     CONTINUE
      CALL CLOSE(2)
      T=2.*TD/XLL
      XK0=1./(2.*FR*FR)
      WRITE(1,*)'FN = ',FR
      CALL POINT
      DO 100 K=1,NZ1
      ZZ=ZC(K)
      XX=XC(NX1)
      ICP=3
      YY=0.0
      CALL SUP1(XX,YY,ZZ,ICP)
      CALL SUP2(XX,YY,ZZ,ICP)
      CALL SUP3(XX,YY,ZZ,ICP)
      DO 10 I=1,NX1
      DO 10 J=1,NZ1
         V1(I,J,1)=V1(I,J,1)+V2(I,J,1)
         V3(I,J,1)=V1(I,J,1)+V3(I,J,1)
         V1(I,J,3)=V1(I,J,3)+V2(I,J,3)
         V3(I,J,3)=V1(I,J,3)+V3(I,J,3)
10    CONTINUE
      DO 20 J=1,NX1
         UU(J)=0.0
         DO 30 II=1,NX1
            IS1=NX1-J+II
            IS2=NX1+J-II
            DO 30 JJ=1,NZ1
               IF(II.LE.J) U(II,JJ)=V3(IS1,JJ,1)
               IF(II.GT.J) U(II,JJ)=-V1(IS2,JJ,1)
```

```
30          CONTINUE
            DO 40 II=1,NX1
               DO 40 JJ=1,NZ1
                  UU(J)=UU(J)+U(II,JJ)*SM(II,JJ)
40          CONTINUE
20       CONTINUE
         DO 50 J=1,NX1
            WW(J)=0.0
            DO 60 II=1,NX1
               IS1=NX1-J+II
               IS2=NX1+J-II
               DO 60 JJ=1,NZ1
                  IF(II.LE.J) U(II,JJ)=V3(IS1,JJ,3)
                  IF(II.GT.J) U(II,JJ)=V1(IS2,JJ,3)
60          CONTINUE
            DO 70 II=1,NX1
               DO 70 JJ=1,NZ1
                  WW(J)=WW(J)+U(II,JJ)*SM(II,JJ)
70          CONTINUE
50       CONTINUE
         WRITE(1,*) ' '
         WRITE(1,*) ' '
         WRITE(1,*) 'ISOBAR No. ',K
         WRITE(1,*) ' '
         WRITE(1,*) 'NONDIMENSIONAL DISTURBANCE VELOCITY u/U'
         WRITE(1,*) (UU(I),I=1,NX1)
         WRITE(1,*) ' '
         WRITE(1,*) 'NONDIMENSIONAL DISTURBANCE VELOCITY w/U'
         WRITE(1,*) (WW(I),I=1,NX1)
         WRITE(1,*) ' '
100      CONTINUE
         STOP
         END
         SUBROUTINE POINT
         COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
         1,FR,VEL,XS,XE,XK0
         SX=XL/(NX-1)
         SZ=T/(NZ-1)
         DO 10 I=1,NX
            X(I)=-1.+SX*(I-1)
            IF(I.EQ.NX) GO TO 10
            XC(I)=-1+SX*(I-0.5)
10       CONTINUE
         DO 20 I=1,NZ
            Z(I)=-T+SZ*(I-1)
            IF(I.EQ.NZ) GO TO 20
            ZC(I)=-T+SZ*(I-0.5)
20       CONTINUE
         RETURN
         END
         SUBROUTINE SUP1(XX,Y,ZZ,ICP)
         COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
         1,FR,VEL,XS,XE,XK0
         COMMON /D/ V1(20,10,3),V2(20,10,3),V3(20,10,3)
         COMMON /F/ F(21,11,3),F1(3)
         DOUBLE PRECISION F1,F
         DO 10 J=1,NZ
            Z1=Z(J)
            DO 10 I=1,NX
               X1=X(I)
               CALL COEF1(XX,Y,ZZ,X1,Z1,XK0,F1,ICP)
C              TYPE *,F1
```

```
            DO 20 II=1,ICP
               F(I,J,II)=F1(II)
20          CONTINUE
10    CONTINUE
      CC=1./(4.*3.1416)
      DO 30 J=1,NZ1
         J1=J+1
         DO 30 I=1,NX1
            I1=I+1
            DO 40 II=1,ICP
               P=F(I1,J1,II)+F(I,J,II)-F(I,J1,II)-F(I1,J,II)
               V1(I,J,II)=P*CC
40          CONTINUE
30    CONTINUE
      RETURN
      END
      SUBROUTINE COEF1(XX,Y,ZZ,X1,Z1,XK0,F1,ICP)
      DIMENSION F1(3)
      DOUBLE PRECISION R1,R2,F1,D1,D2,XY,YZ
      DOUBLE PRECISION DX1,C1,C2
      DX1=X1-XX
      C1=Z1-ZZ
      C2=Z1+ZZ
      IF(C1.EQ.0.0) C1=-1.E-4
      IF(C2.EQ.0.0) C2=-1.E-4
      XY=DX1*DX1+Y*Y
      R1=SQRT(XY+C1*C1)
      R2=SQRT(XY+C2*C2)
      IF(ABS(Y).LE.1.E-10) F1(2)=0.0
      IF(ABS(Y).GT.1.E-10) F1(2)=ATAN(DX1*C1)/(Y*R1))
     1-ATAN((DX1*C2)/(Y*R2))
      IF(XY.LE.1.E-10) F1(1)=DLOG(ABS(C2)/ABS(C1))
      D1=C1+R1
      D2=C2+R2
      IF(XY.GT.1.E-10) F1(1)=DLOG(D1/D2)
      YZ=Y*Y+C1*C1
      IF(YZ.LE.1.E-10) F1(3)=-2.*DLOG(ABS(DX1))
      D1=DX1+R1
      D2=DX1+R2
      IF(YZ.GT.1.E-10) F1(3)=DLOG(D1*D2)
      IF(ICP.EQ.1) F1(2)=0.0
      IF(ICP.EQ.1) F1(3)=0.0
      RETURN
      END
      SUBROUTINE SUP2(XX,Y,ZZ,ICP)
      COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
     1,FR,VEL,XS,XE,XK0
      COMMON /D/ V1(20,10,3),V2(20,10,3),V3(20,10,3)
      COMMON /F/ F(21,11,3),F1(3)
      DO 10 J=1,NZ
         Z1=Z(J)
         DO 10 I=1,NX
C        TYPE *,'SU2   J=',J,'   I=',I
            X1=X(I)
            CALL COEF(XX,Y,ZZ,X1,Z1,XK0,F1,ICP)
            DO 20 II=1,ICP
               F(I,J,II)=F1(II)
20          CONTINUE
10    CONTINUE
      CC=-1./(4.*3.1416*3.1416)
      DO 30 J=1,NZ1
         J1=J+1
```

```
      DO 38 I=1,NX1
         I1=I+1
         DO 48 II=1,ICP
            Q=F(I1,J1,II)+F(I,J,II)-F(I,J1,II)-F(I1,J,II)
            V2(I,J,II)=Q*CC
            IF(II.EQ.3) V2(I,J,II)=2.*V2(I,J,II)
48       CONTINUE
38    CONTINUE
      RETURN
      END
      SUBROUTINE COEF(X,Y,Z,X1,Z1,XX8,F1,ICP)
      DIMENSION F1(3),YF(3),TH(4)
      COMMON /A1/ Y1,XX,DX1,C1
      Y1=Y
      XX=XX8
      DX1=X-X1
      C1=Z+Z1
      IF(C1.EQ.8.8) C1=-1.E-4
      TH(1)=8.8
      TH(2)=88.8*3.1416/188.
      TH(3)=89.995*3.1416/188.
      DO 18 I=1,ICP
      F1(I)=8.8
      YF(I)=8.8
18    CONTINUE
      CALL ITS(THETA)
      TH(4)=THETA
      NN=4
      CALL COMP(NN,TH)
      DO 28 I=1,3
         ST=TH(I)
         EN=TH(I+1)
         IF(ST.EQ.EN) GO TO 28
         N=58
         ICC=1
         CALL INT(N,ST,EN,YF,ICP,ICC)
         DO 38 J=1,ICP
            F1(J)=F1(J)+YF(J)
38       CONTINUE
28    CONTINUE
58    RETURN
      END
      SUBROUTINE COMP(N,X)
      DIMENSION X(N)
C     TYPE *,X
      DO 5 J=1,N-1
         K=N-J
         DO 18 I=1,K
            IF(X(I).LE.X(I+1)) GO TO 18
            TEMP=X(I)
            X(I)=X(I+1)
            X(I+1)=TEMP
18       CONTINUE
5     CONTINUE
C     TYPE *,X
      RETURN
      END
      SUBROUTINE ITS(THETA)
      DIMENSION B(2)
      COMMON /A1/ Y,XX8,DX1,C1
      THEND=89.995*3.1416/188.
      THET=18.*3.1416/188.
```

```
      ICNT=1
20    F1=COS(THET)
      F2=SIN(THET)
      A=XK0/(F1*F1)
      B(1)=DX1+Y*F2
      B(2)=DX1-Y*F2
      IF(B(1)*B(2).LT.0.) THETA=THEND
      IF(B(1)*B(2).LT.0.) GO TO 100
      IF(B(1).GE.0.0) T1=-3.1416*2.
      IF(B(1).LT.0.0) T1=3.1416*2.
30    XT=C1*A
      TEMP=0.
      DO 40 I=1,2
      YT=B(I)*A
      FS=XT*XT+YT*YT
      GS=XT+0.02*XT*XT+0.02*YT*YT
      IF(FS.GT.1.AND.GS.GT.0.) CALL SIMP(XT,YT,VAL,VAL1)
      IF(FS.LE.1.OR.GS.LE.0.) CALL SCOM(XT,YT,VAL,VAL1)
      TEMP=TEMP+(ATAN(B(I)/C1)+VAL1)/T1
40    CONTINUE
      A1=ALOG(TEMP)/C1
      CX1=ABS(A1/A-1.)
      A=A1
      IF(CX1.GT.1.E-4) GO TO 30
      A2=XK0/A1
      IF(ABS(A2).GT.1.) THETA=THEND
      IF(ABS(A2).LE.1.) THETA=ACOS(SQRT(A2))
      CX2=ABS(THETA/THET-1.)
      THET=THETA
      ICNT=ICNT+1
      IF(ICNT.GT.50) GO TO 100
      IF(THETA.NE.THEND.AND.CX2.GT.1.E-4) GO TO 20
100   RETURN
      END
      SUBROUTINE INT(N1,ST,EN,YF,ICP,ICC)
      DIMENSION YF(3),F(3),SF1(3,3),SF2(3,3),FSUM1(3),FSUM2(3),CXF(3)
      COMMON /A1/ Y,XK0,DX1,C1
C     WRITE(4,*) ' '
      D1=(EN-ST)/N1
      DO 10 I=1,ICP
      DO 10 J=1,3
        SF1(I,J)=0.0
10    CONTINUE
      DO 20 I=1,N1+1
        THETA=ST+D1*(I-1)
        ANG=THETA*180./3.1416
        CALL COM(THETA,F,ICP)
C       WRITE(4,*) ANG,F
        II=I/2*2
        IF(I.EQ.1.OR.I.EQ.N1+1) IC=1
        IF(II.EQ.I) IC=2
        IF(II.NE.I.AND.I.NE.1.AND.I.NE.N1+1) IC=3
        DO 30 J=1,ICP
          SF1(J,IC)=SF1(J,IC)+F(J)
30      CONTINUE
20    CONTINUE
      DO 40 I=1,ICP
        FSUM1(I)=(SF1(I,1)+4.*SF1(I,2)+2.*SF1(I,3))*D1/3.
        FSUM2(I)=FSUM1(I)
40    CONTINUE
C     TYPE *,FSUM1
      IF(ICC.EQ.1) GO TO 110
```

```
      K=1
45    K=K+1
      DO 50 I=1,ICP
         SF2(I,1)=SF1(I,1)
         SF2(I,3)=SF1(I,2)+SF1(I,3)
50    CONTINUE
      N2=N1
      D2=D1/2.
      DO 60 I=1,ICP
         SF2(I,2)=0.0
60    CONTINUE
      DO 70 I=1,N2
         THETA=ST+D2*(2*I-1)
         CALL COM(THETA,F,ICP)
         DO 80 J=1,ICP
            SF2(J,2)=SF2(J,2)+F(J)
80       CONTINUE
70    CONTINUE
      IR=0
      DO 90 I=1,ICP
         FSUM2(I)=(SF2(I,1)+4.*SF2(I,2)+2.*SF2(I,3))*D2/3.
         IF(FSUM1(I).EQ.0.0) CXF(I)=0.0
         IF(FSUM1(I).NE.0.0) CXF(I)=ABS(FSUM2(I)/FSUM1(I)-1.)
         IF(CXF(I).GT.1.E-3) IR=1
90    CONTINUE
C     TYPE *,FSUM2
      IF(IR.EQ.0) GO TO 110
      DO 100 I=1,ICP
         FSUM1(I)=FSUM2(I)
         DO 100 J=1,3
            SF1(I,J)=SF2(I,J)
100   CONTINUE
      N1=2.*N2
      D1=D2
      IF(K.GT.7) GO TO 110
      GO TO 45
C
110   DO 120 I=1,ICP
         YF(I)=FSUM2(I)
120   CONTINUE
      RETURN
      END
      SUBROUTINE COM(THETA,YF,ICP)
      DIMENSION YF(3),B(2)
      COMMON /A1/ Y,XK0,DX1,C1
      T=TAN(THETA)
      FF1=COS(THETA)
      CO2=FF1*FF1
      F2=SIN(THETA)
      F3=Y*F2
      A=XK0/CO2
      B(1)=DX1*FF1+F3
      B(2)=DX1*FF1-F3
      XT=C1*A
      DO 10 I=1,ICP
         YF(I)=0.0
10    CONTINUE
      IF(ABS(Y).LE.1.E-8) IS=1
      IF(ABS(Y).GT.1.E-8) IS=2
      DO 20 I=1,IS
      YT=B(I)*A
      FS=XT*XT+YT*YT
```

```
      GS=XT+0.02*XT*XT+0.02*YT*YT
      IF(FS.GT.1.AND.GS.GT.0.) CALL SIMP(XT,YT,VAL,VALI)
      IF(FS.LE.1.OR.GS.LE.0.) CALL SCOM(XT,YT,VAL,VALI)
      YF1=2.*VAL+ALOG(C1*C1+B(I)*B(I))
      YF(1)=YF(1)+YF1
      IF(ICP.EQ.1) GO TO 20
      IF(IS.EQ.2.AND.I.EQ.1) YF(2)=YF1
      IF(IS.EQ.2.AND.I.EQ.2) YF(2)=YF(2)-YF1
      IF(IS.EQ.1) YF(2)=0.0
      IF(YT.LT.0.0) T1=-3.1416
      IF(YT.GT.0.0) T1=3.1416
      YGS=ATAN(B(I)/C1)+T1*EXP(XT)
      YF(3)=YF(3)+VALI+YGS
20    CONTINUE
      YF(2)=YF(2)*T
      YF(3)=YF(3)/FF1
      IF(IS.EQ.2) GO TO 300
      DO 30 I=1,ICP
        YF(I)=2.*YF(I)
30    CONTINUE
300   RETURN
      END
      SUBROUTINE SIMP(TC1,TB1,VALR,VALI)
      DIMENSION S1(3),S2(3),SA1(3),SA2(3),CX(2)
      N1=20
      D1=3.1416/(2.*N1)
      K=1
      DO 10 I=1,3
        S1(I)=0.0
        SA1(I)=0.0
10    CONTINUE
      DO 20 I=1,N1+1
        ALPHA=D1*(I-1)
        IF(I.EQ.N1+1) ALPHA=89.995*3.1416/180.
        T=TAN(ALPHA)
        CS=COS(ALPHA)
        A1=T+TC1
        XI1=EXP(-T)/(CS*CS)
        XIR=A1*XI1
        XII=TB1*XI1
        XI2=(A1*A1)+(TB1*TB1)
        VS=XIR/XI2
        VS1=XII/XI2
        II=I/2*2
        IF(I.EQ.1.OR.I.EQ.N1+1) IC=1
        IF(II.EQ.I) IC=2
        IF(II.NE.I.AND.I.NE.1.AND.I.NE.N1+1) IC=3
        S1(IC)=S1(IC)+VS
        SA1(IC)=SA1(IC)+VS1
20    CONTINUE
      VAL1=(S1(1)+4.*S1(2)+2.*S1(3))*D1/3.
      VA1=(SA1(1)+4.*SA1(2)+2.*SA1(3))*D1/3.
45    K=K+1
      S2(1)=S1(1)
      S2(3)=S1(2)+S1(3)
      SA2(1)=SA1(1)
      SA2(3)=SA1(2)+SA1(3)
      N2=N1
      D2=D1/2.
      S2(2)=0.0
      SA2(2)=0.0
      DO 70 I=1,N2
```

```
      ALPHA=D2*(2*I-1)
      T=TAN(ALPHA)
      CS=COS(ALPHA)
      A1=T+TC1
      XI1=EXP(-T)/(CS*CS)
      XIR=A1*XI1
      XII=TB1*XI1
      XI2=(A1*A1)+(TB1*TB1)
      VS=XIR/XI2
      VS1=XII/XI2
      S2(2)=S2(2)+VS
      SA2(2)=SA2(2)+VS1
70    CONTINUE
      VAL2=(S2(1)+4.*S2(2)+2.*S2(3))*D2/3.
      VA2=(SA2(1)+4.*SA2(2)+2.*SA2(3))*D2/3.
      IF(VAL1.EQ.0.0) CX(1)=0.0
      IF(VAL1.NE.0.0) CX(1)=ABS(VAL2/VAL1-1.)
      IF(VA1.EQ.0.0) CX(2)=0.0
      IF(VA1.NE.0.0) CX(2)=ABS(VA2/VA1-1.)
      IF(CX(1).LE.1.E-3.AND.CX(2).LE.1.E-3) GO TO 110
      VAL1=VAL2
      VA1=VA2
      DO 100 J=1,3
        S1(J)=S2(J)
        SA1(J)=SA2(J)
100   CONTINUE
      N1=2*N2
      D1=D2
      IF(K.GT.10) GO TO 110
      GO TO 45
C
110   VALR=VAL2
      VALI=-VA2
      RETURN
      END
      SUBROUTINE SCOM(XT,YT,VALR,VALI)
      COMPLEX*8 Z,VN,Z1
      VNR=0.0
      VNI=0.0
      Z=CMPLX(XT,YT)
      R=ABS(Z)
      IF(XT.NE.0.0) TH=ATAN(ABS(YT/XT))
      IF(XT.EQ.0.0) TH=3.1416/2.
      IF(XT.GE.0.0.AND.YT.GT.0.0) TH=TH+0.0
      IF(XT.GE.0.0.AND.YT.LT.0.0) TH=-TH
      IF(XT.LT.0.0.AND.YT.GT.0.0) TH=3.1416-TH
      IF(XT.LT.0.0.AND.YT.LT.0.0) TH=-3.1416+TH
      IF(XT.EQ.0) GO TO 10
      Z1=Z
      VN=-0.5772157-CMPLX(ALOG(R),TH)-Z1
      DO 20 I=2,1000
        XN=FLOAT(I)
        Z1=(XN-1)*Z1*(-Z)/(XN*XN)
        VN=VN-Z1
        ARL=REAL(VN)
        AIM=AIMAG(VN)
        IF(ARL.EQ.0.0.AND.AIM.EQ.0.0) GO TO 25
        BRL=ABS(1.-VNR/ARL)
        BIM=ABS(1.-VNI/AIM)
        IF(ABS(BRL).LE.1.E-5.AND.ABS(BIM).LE.1.E-5) VN=VN*EXP(Z)
        IF(ABS(BRL).LE.1.E-5.AND.ABS(BIM).LE.1.E-5) GO TO 100
25      VNR=ARL
```

```
         VNI=AIM
   20  CONTINUE
       VN=VN*EXP(Z)
       WRITE(6,*)'UNCONVERGENCE Z=',Z
       GO TO 100
   10  X1=ABS(XT)
       VN=-0.5772157-ALOG(X1)-X1
       DO 30 I=2,1000
         XN=FLOAT(I)
         X1=(XN-1)*X1*ABS(XT)/(XN*XN)
         VN=VN-X1
         ARL=REAL(VN)
         AIM=AIMAG(VN)
         BRL=ABS(1.-ARL/VNR)
         BIM=ABS(1.-AIM/VNI)
         IF(ABS(BRL).LE.1.E-3.AND.ABS(BIM).LE.1.E-3) VN=VN*EXP(Z)
         IF(ABS(BRL).LE.1.E-3.AND.ABS(BIM).LE.1.E-3) GO TO 100
         VNR=ARL
         VNI=AIM
   30  CONTINUE
       TYPE *,'UNCONVERGENCE X=',X
       VN=VN*EXP(Z)
  100  VALR=REAL(VN)
       VALI=AIMAG(VN)
       RETURN
       END
       SUBROUTINE SUP3(XX,Y,ZZ,ICP)
       COMMON /A/ NX,NX1,NX2,NZ,NZ1,NZ2,XL,T,X(21),Z(11),XC(20),ZC(10)
      1,FR,VEL,XS,XE,XK0
       COMMON /D/ V1(20,10,3),V2(20,10,3),V3(20,10,3)
       COMMON /F/ F(21,11,3),F1(3)
       COMMON /A1/ Y1,XK,DX1,C1,II
       Y1=Y
       XK=XK0
       DO 10 J=1,NZ
         Z1=Z(J)
         DO 10 I=1,NX
C          TYPE *,'SU3   J=',J,'   I=',I
           X1=X(I)
           DX1=XX-X1
           C1=ZZ+Z1
           IF(ABS(C1).LE.1.E-8) C1=-1.E-4
           IF(ABS(Y).GT.1.E-8) CALL INT1(F1,ICP)
           IF(ABS(Y).LE.1.E-8) CALL INT2(F1,ICP)
           DO 20 II=1,ICP
             F(I,J,II)=F1(II)
   20      CONTINUE
   10  CONTINUE
       CC=1./3.1416
       DO 30 J=1,NZ1
         J1=J+1
         DO 30 I=1,NX1
           I1=I+1
           DO 40 II=1,ICP
             P=F(I1,J1,II)+F(I,J,II)-F(I,J1,II)-F(I1,J,II)
             V3(I,J,II)=P*CC
   40      CONTINUE
   30  CONTINUE
       RETURN
       END
       SUBROUTINE INT2(F1,ICP)
       DIMENSION F1(3)
```

```
      COMMON /A1/ Y,XK0,DX1,C1,II
      EXTERNAL FU
      DO 5 II=1,ICP
C        IF(DX1.LE.0.0) F1(II)=0.0
C        IF(DX1.LE.0.0) GO TO 5
         IF(II.EQ.2) F1(II)=0.0
         IF(II.EQ.2) GO TO 5
         IF(DX1.LE.0.0) DX1=0.0
         SUM=0.0
         SUM1=0.0
         TH1=0.0
         DO 10 I=1,1000
            IF(II.EQ.1) SC=FLOAT(I)
            IF(II.NE.1) SC=FLOAT(I)*2.
            S=XK0*DX1/(SC*3.1416)
            IF(S.GT.1) GO TO 10
            TH2=ACOS(S)
            IF(TH2.GT.3.1416/2.) TH2=3.1416/2.
            G1=TH1*180./3.1416
            G2=TH2*180./3.1416
            CALL SIMPSN(TH1,TH2,FU,VAL)
            SUM=SUM+VAL
            IF(TH2.GE.3.1416/2.) GO TO 20
            IF(SUM.EQ.0.0) CX=0.0
            IF(SUM.NE.0.0) CX=ABS(SUM1/SUM-1.)
            IF(CX.LE.1.E-4) GO TO 20
            SUM1=SUM
            TH1=TH2
10       CONTINUE
20       ANG=TH2/3.1416*180.
         F1(II)=SUM*2.
5     CONTINUE
      RETURN
      END
      SUBROUTINE INT1(F1,ICP)
      DIMENSION F1(3),T(10)
      COMMON /A1/ Y,XK0,DX1,C1,II
      EXTERNAL FU1
      F(X)=X*180./3.1416
      GG(A,B,THETA)=A*COS(THETA)-B-TAN(THETA)
      PI=3.1416
      B=DX1/Y
      R=SQRT(DX1*DX1+Y*Y)
      ALPHA=ACOS(DX1/R)
      AA=F(TH1)
      THF=PI/2.
      DO 100 II=1,ICP
C     TYPE *,'*****II=',II
         F1(II)=0.0
         TH1=-PI/2.+ALPHA
C        TYPE *,'TH1=',TH1
         IF(TH1.GT.0.0) GO TO 20
         TT=F(TH1)
C        TYPE *,'II TH1=',TT
         SUM1=0.0
         N=0
         THETA=-80.*3.1416/180.
10       N=N+1
         BB=FLOAT(N)
         A=BB*PI/(XK0*Y)
5        TH2=ATAN(A*COS(THETA)-B)
         IF(ABS(THETA/TH2-1.).LE.1.E-2) TS=(THETA+TH2)/2.
```

```
      IF(ABS(THETA/TH2-1.).LE.1.E-2) CALL STACK(N,TS,T)
      IF(ABS(THETA/TH2-1.).LE.1.E-2) GO TO 10
      THETA=(TH2+THETA)/2.
      IF(THETA.LT.0.0) GO TO 5
      N=N-1
C     TYPE *,'THE NUMBER OF ZERO POINT N=',N
      IF(N.EQ.0) N=1
      IF(N.GT.10) NN=10
      IF(N.LE.10) NN=N
      AU=0.0
      DO 15 I=1,NN
        N1=NN-I+1
        AL=T(N1)
        IF(AL.LT.TH1) AL=TH1
        IF(N.EQ.1) AL=TH1
        CALL SIMPSN(AL,AU,FU1,VAL)
        A1=F(AU)
        A2=F(AL)
C       TYPE *,'AU=',A1,'AL=',A2,'VAL=',VAL
        F1(II)=F1(II)+VAL
C       TYPE *,'F1(II)=',F1(II)
        IF(ABS(AL-TH1).LE.1.E-6) GO TO 20
        IF(F1(II).EQ.0.0) CX=0.0
        IF(F1(II).NE.0.0) CX=ABS(SUM1/F1(II)-1.)
        IF(CX.LE.1.E-4) GO TO 20
        SUM1=F1(II)
        AU=AL
15    CONTINUE
20    IF(TH1.LT.0.0) TH1=0.0
      XL=TH1
23    XU=89.99*3.1416/180.
      BB=FLOAT(N)
      A=BB*PI/(XK0*Y)
25    XM=(XL+XU)/2.
      YL=GG(A,B,XL)
      YU=GG(A,B,XU)
      YM=GG(A,B,XM)
C     TYPE *,'XL=',XL,'YL=',YL
C     TYPE *,'XM=',XM,'YM=',YM
C     TYPE *,'XU=',XU,'YU=',YU
      IF(YM*YL.GE.0.0) XL=XM
      IF(YM*YL.LT.0.0) XU=XM
      T1=F(XL)
      T2=F(XU)
      IF(ABS(XU-XL).LE.1.E-6) TH2=(XL+XU)/2.
      IF(ABS(XU-XL).LE.1.E-6) GO TO 30
      GO TO 25
30    IF(TH2.LT.TH1) GO TO 50
C     IF(ABS(TH1-TH2).LE.1.E-6) TH2=89.99*3.1416/180.
      CALL SIMPSN(TH1,TH2,FU1,VAL)
      A1=F(TH1)
      A2=F(TH2)
C     TYPE *,'TH1=',A1,'TH2=',A2,'VAL=',VAL
      F1(II)=F1(II)+VAL
      IF(TH1.GT.F(80).AND.ABS(VAL).LE.1.E-20) GO TO 100
C     TYPE *,'F1(II)=',F1(II)
      IF(F1(II).EQ.0.0) GO TO 55
      IF(F1(II).NE.0.0) CX=ABS(SUM1/F1(II)-1.)
      IF(CX.LE.1.E-4) GO TO 100
      SUM1=F1(II)
55    TH1=TH2
50    N=N+1
```

```
      XL=TH1
      IF(N.LE.1000) GO TO 23
100   CONTINUE
70    RETURN
      END
      SUBROUTINE STACK(N,TS,T)
      DIMENSION T(10)
      IF(N.LE.10) GO TO 10
      DO 20 I=1,9
        I1=I+1
        T(I)=T(I1)
20    CONTINUE
      T(10)=TS
      GO TO 30
10    T(N)=TS
30    RETURN
      END
      SUBROUTINE SIMPSN(A,B,F,VAL)
      DIMENSION S1(3),S2(3)
      N1=10
      D1=(B-A)/N1
      K=1
      DO 10 I=1,3
        S1(I)=0.0
10    CONTINUE
      DO 20 I=1,N1+1
        ALPHA=A+D1*(I-1)
        VS=F(ALPHA)
        II=I/2*2
        IF(I.EQ.1.OR.I.EQ.N1+1) IC=1
        IF(II.EQ.I) IC=2
        IF(II.NE.I.AND.I.NE.1.AND.I.NE.N1+1) IC=3
        S1(IC)=S1(IC)+VS
20    CONTINUE
      VAL1=(S1(1)+4.*S1(2)+2.*S1(3))*D1/3.
45    K=K+1
      S2(1)=S1(1)
      S2(3)=S1(2)+S1(3)
      N2=N1
      D2=D1/2.
      S2(2)=0.0
      DO 70 I=1,N2
        ALPHA=A+D2*(2*I-1)
        VS=F(ALPHA)
        S2(2)=S2(2)+VS
70    CONTINUE
      VAL2=(S2(1)+4.*S2(2)+2.*S2(3))*D2/3.
      IF(VAL1.EQ.0.0) CX=0.0
      IF(VAL1.NE.0.0) CX=ABS(VAL2/VAL1-1.)
      IF(CX.LE.1.E-4) GO TO 110
      VAL1=VAL2
      DO 100 J=1,3
        S1(J)=S2(J)
100   CONTINUE
      N1=2*N2
      D1=D2
      IF(K.GT.10) GO TO 110
      GO TO 45
C
110   VAL=VAL2
200   RETURN
      END
```

```
      FUNCTION FU(THETA)
      DOUBLE PRECISION XI1,F1,F2,B1,B2,A
      COMMON /A1/ Y,XK0,DX1,C1,I
      F1=COS(THETA)
      F2=SIN(THETA)
      A=XK0/(F1*F1)
      XI1=EXP(A*C1)
      B1=XK0*DX1/F1
      B2=A*F2*Y
      IF(I.EQ.1) FU=XI1*SIN(B1)
      IF(I.EQ.2) FU=XI1*COS(B1)*F2/F1
C     IF(I.EQ.3) FU=-XI1*(COS(B1)-1.)/F1
      IF(I.EQ.3) FU=-XI1*COS(B1)/F1
      RETURN
      END
      FUNCTION FU1(THETA)
      COMMON /A1/ Y,XK0,DX1,C1,I
      F1=COS(THETA)
      F2=SIN(THETA)
      A=XK0/(F1*F1)
      XI1=EXP(A*C1)
      B1=A*(DX1*F1+Y*F2)
      IF(I.EQ.1) FU1=XI1*SIN(B1)
      IF(I.EQ.2) FU1=XI1*SIN(B1)*F2/F1
      IF(I.EQ.3) FU1=-XI1*(COS(B1)-1.)/F1
      RETURN
      END
```