AD-A015 699

MATHEMATICAL MODEL OF AN AIR CUSHION VEHICLE

Damon Cummings, et al

Charles Stark Draper Laboratory, Incorporated

Prepared for:

Naval Training Equipment Center

May 1975

AD-A015699

Technical Report: NAVTRAEQUIPCEN 73-C-0138-1

MATHEMATICAL MODEL OF AN AIR CUSHION VEHICLE

Final Report

The Charles Stark Draper Laboratory, Inc.
68 Albany Street
Cambridge, Massachusetts 02139
NAVTRAEQUIPCEN Task No. 3741

May 1975

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NAVTRAEQUIPCEN 73-C-0138-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Mathematical Model of an Air Cushion Vehicle Final Report		5. TYPE OF REPORT & PERIOD COVERED Final Report 8 June 1973 thru 8 Nov 1974
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Damon Cummings, Stanley Shursky Edward Kern, Ronald Yeung		8. CONTRACT OR GRANT NUMBER(s) N61339-73-C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS The C.S. Draper Laboratory, Inc. & The Mass. Institute of Technology Cambridge, Massachusetts		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS TASK 3741-01P01
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Orlando, Florida 32813		12. REPORT DATE May 1975
		13. NUMBER OF PAGES ~~126~~ 135
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ACV, Mathematical Modeling, Simulation, Training Device, Landing Craft, Cushion dynamics

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

This report describes the development of a mathematical model of the Bell Aerospace Air Cushion Vehicle Landing Craft (JEFF-B). The model is intended for use in a real time hands on pilot training simulator. Equations of motion, cushion dynamics, control and machinery dynamics and water wave effects are modeled.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

In recent years, the development of various air cushion vehicles (ACV) is drawing worldwide interest. In conjunction with the Navy's Sea Control Ship Program, the military application of various air cushion vehicles is steadily increasing. This study was initiated to provide the necessary mathematical model of a selected vehicle for use in a training device for the ACV.

Under the study contract, a mathematical model of an Amphibious Assault Landing Craft, (AALC), JEFF-B, developed by the Bell Aerospace Company was provided. The six-degree-of-freedom equations of motion of the craft were derived with respect to the pilot's seat. The techniques used in simulating the craft's engines and system components were conventional. The technique used in simulating the cushion dynamics was very sophisticated. The model is suitable for real time digital computer activated training simulation.

This report contains three sections and two appendices. The first section is intended as a general introduction to the mathematical modeling problem and unique vehicle characteristics. Second section describes the subsystems of the vehicle and their integration into the whole simulation. The third section develops the algebraic model used for each component of the simulation and the equations of motion. The appendices describe mathematical details of the derivation of the vehicle-generated water wave elevations.

Wei-Hung Yih

WEI-HUNG YIH
Project Engineer
Naval Training Equipment Center

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

PROGRAM ILLUSTRATIONS

SECTION I

INTRODUCTION

GENERAL DESCRIPTION

The mathematical model described herein is intended for use in a pilot training simulator. It attempts to model the vehicle responses to pilot controls and external environmental inputs such as wind and waves as faithfully as can be done within the real time capability of digital computation. The model is based on the description of the JEFF B craft given in Bell Aerospace's Preliminary Design Summary Report, (PDSR) updated by verbal and written information obtained as the vehicle construction progresses. The capability exists of updating the model further as model and full scale test information is received and as the vehicle hardware is altered. However, this model is not intended for design or engineering purposes. In many cases, such as the cushion pressure model, the calculations are based on scanty experimental and analytical evidence that should not be taken for more than what it is; namely, the best estimate we can make of the dynamic charactersistics of the particular system. Extrapolation of such derivations and curve fits to other designs is not recommended. As further experimental and theoretical work is done at Naval Ships Research and Development Center and elsewhere, more refined procedures will undoubtedly become available for various parts of the system. The model is broken down into subsystems in such a way that updates are readily incorporated.

Many of the forces acting on the vehicle are curve fits to experimental data obtained by Bell Aerospace and used in their calculations and preliminary fan characteristics, and control forces. However, the basic dynamics of the vehicle, in particular the cushion pressure system and its interaction with the water or ground surface has been developed by the Draper Laboratory specifically for the purposes of this six degree of freedom simulation.

There are two ways a simulation problem of this type may be approached. In most marine vehicle problems the user is interested in motions and maneuvers made around some "normal" operating mode. This leads to an assumption of a basic operating mode and expressions for perturbations in series form and, if the system is reasonably linear, it may be linearized about the operating point and higher terms in the series expansion dropped. If on the contrary large excursions from a basic operating point and a highly nonlinear system are encountered, the perturbation approach leads to difficulties both in obtaining the higher order terms and in including them in a real time computational model. In such a case a second approach may be made incorporating the basic physics of the vehicle and environmental forces. For this pilot training simulation large variations in speed, large accelerations, large sideslip, yaw and apparent wind angles, and large variations in control and machinery forces must be anticipated. Moreover small changes in vehicle state, particularly in changing ground clearance or pitch angle, result in highly nonlinear force characteristics with large coupling effects between degrees of freedom. As a result, the more basic approach of modeling the forces on the vehicle as a function of its present state and recent history has been chosen rather than a perturbation approach.

The greatest difference between the Air Cushion Vehicle and conventional marine vehicle modeling is in the description of the air support system itself. The cushion airflow and pressure characteristics greatly influence the motion of the craft in all degrees of freedom both by exerting direct forces on the vehicle and by altering the machinery speed and water surface wave shape. Much effort has therefore been devoted to a model for the cushion pressures that is valid over the full range of vehicle states. The mathematical model described herein for cushion flow is fundamentally a low Mach number or incompressible model. It assumes that variations in air density are small in the system so that pressures are determined by fan and escape area characteristics. The volume flow rate into any cushion must therefore be equal to the escape rate plus the rate of change of cushion volume with time. Initial calculations were made including the effects of air compressibility leading to extremely high frequency pressure oscillations in response to step heave force inputs. The frequency of these pressure variations is of the order of ten times the natural frequencies of craft motion. Their incorporation in a real time digital model is not feasible due to their high speed and their effect on craft motion is small. Therefore, the compressible flow model was dropped and further effort was devoted to refinement of the incompressible flow model. It should be stressed, however, that both these numerical results and experimental data ‹ ւ SES craft indicate that high frequency pressure oscillations exist and can have major impact on structural and machinery design. They are filtered out in the present mathematical model only because of the real time computational restraint and their small effect on overall craft motion.

The cushion pressure system sed on the Bell Aerospace PDSR. A centrifugal fan on each side of the ve le feeds the forward and aft cushion compartments through piping, bags, . nd holes. Total flow through the fans of 21×10^3 cubic feet per second corresponds to a fan pressure of 133 psf and a cushion pressure of 109 psf with a clearance of .25 feet between water surface and skirt hemline. At this condition, approximately 40% of the fan outlet air is exited through the thrust nozzles. As the vehicle approaches the ground, clearance decreases, cushion flow decreases, pressure increases, and nozzle flow increases. To model these effects computationally system characteristics have been pieced together from the Bell PDSR and analytical assumptions.

The fan characteristics are derived from the Bell carpet plot shown in figure 1. Since rpm is a variable in the simulation, but a 3 dimensional surface fit is considered wasteful of computer time, the fans are basically fit with a parabola at 2000 rpm and assumed to follow the pump affinity laws. Presssure is assumed proportional to rpm squared and flow proportional to rpm. The fan curves are therefore effectively non dimensionalized based on their 2000 rpm performance.

Pressure loss between fan and cushion is assumed to be proportional to the square of the flow rate to the particular cushion. Flow through the thrust nozzle is assumed proportional to the square root of the fan pressure.

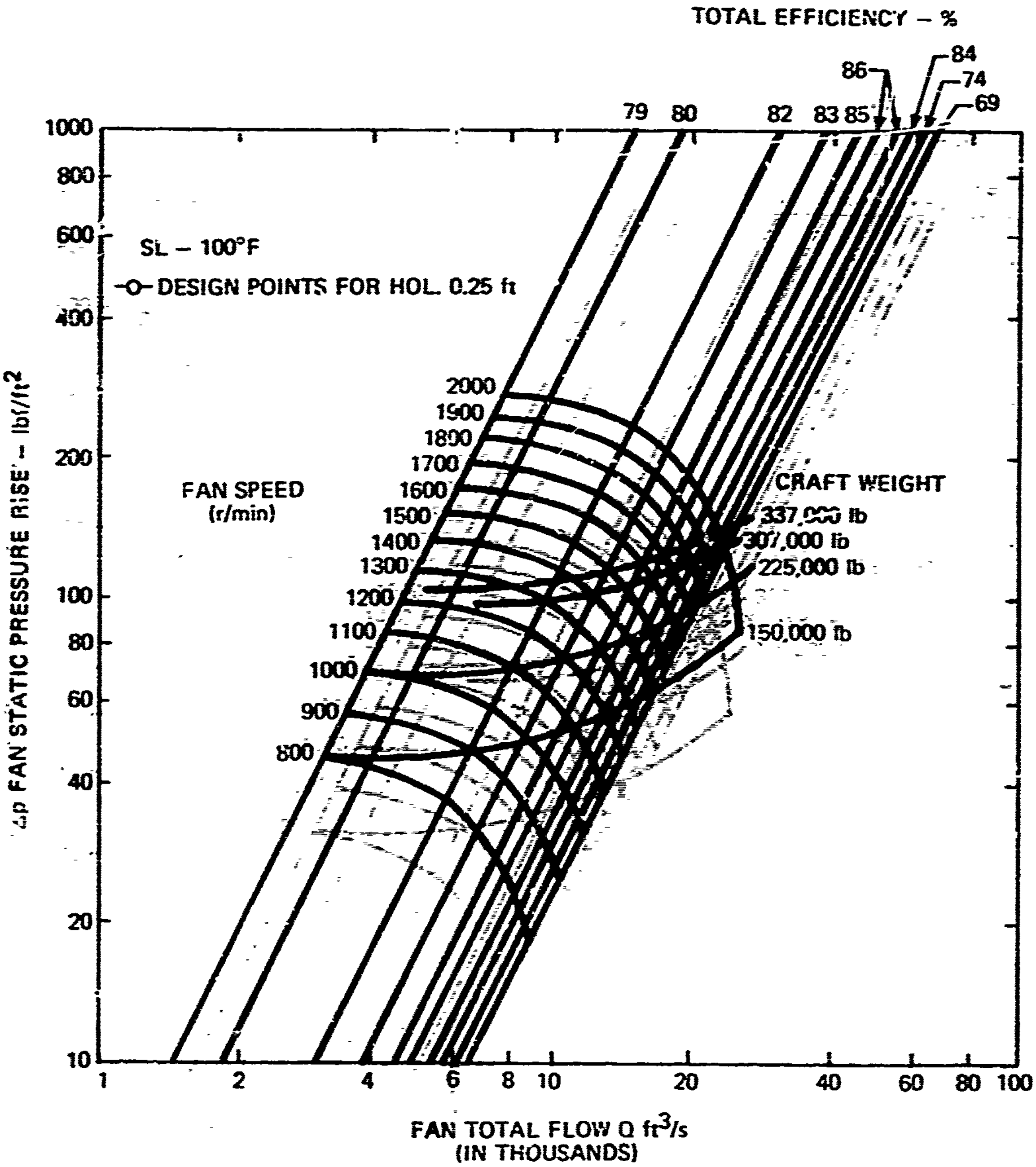


Figure 1. Fan Performance

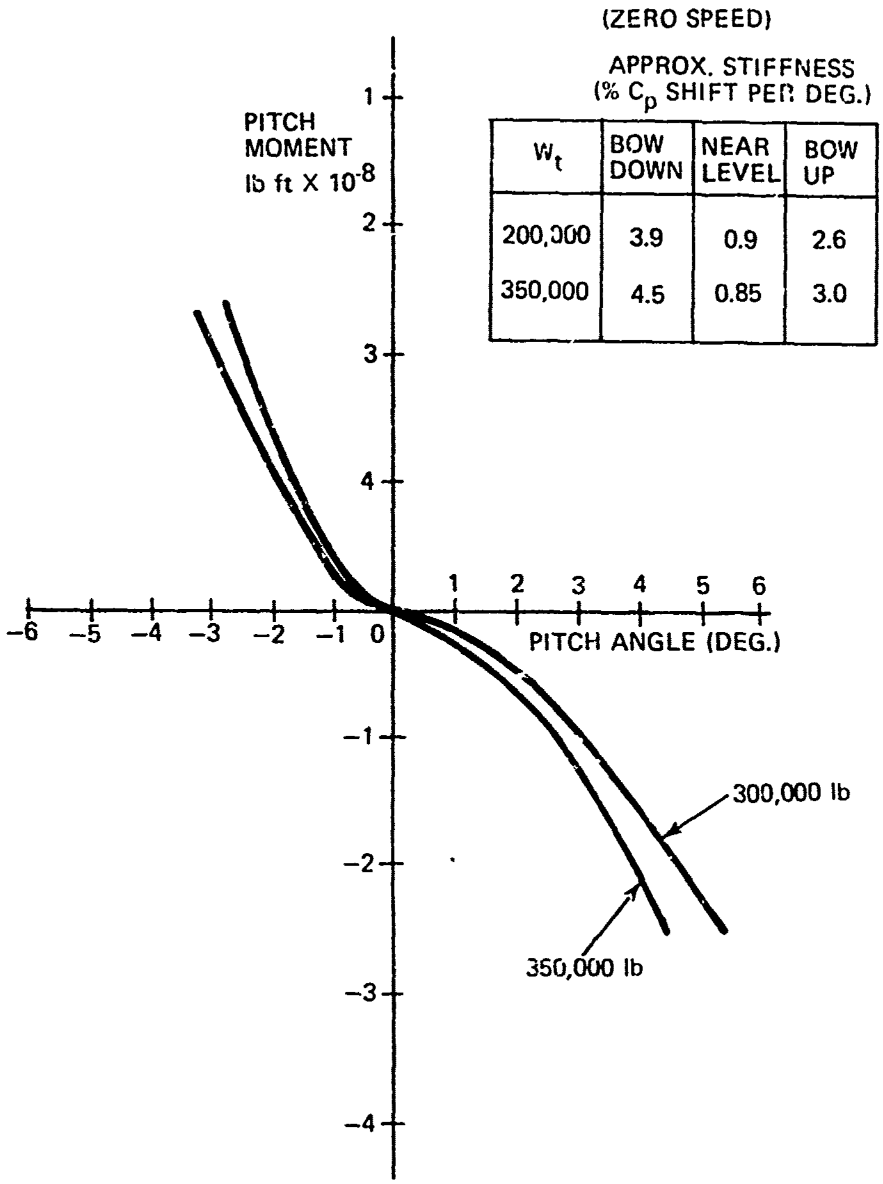
The most critical part of the cushion pressure system is the determination of clearance between the hemline of the skirts and the water surface. These clearances determine the escape area for cushion air and therefore the stiffness of the vehicle in heave, pitch, and roll. As pointed out in the Bell PDSR and confirmed by our numerical modeling, assumption of rigid skirts yields heave stiffnesses far greater than experiment. The mechanism appears to be that as the vehicle approaches the surface, the increased pressure raises the skirt hemline. Therefore, the clearances do not close as much as the vehicle moves down. However as the clearance departs from the equilibrium value of .25 feet, the vehicle stiffness increases dramatically, as shown by the Bell experimental results in figure 2. This highly nonlinear behavior is evidently caused by limits to the amount of vertical skirt motion. This behavior has been modeled by allowing the skirts to move upwards with increasing cushion pressure, but limiting the motion to six inches.

Clearance between the hull bottom plating and the water surface is determined at 35 points as shown in figure 3. The points interior to compartments are used for volume and rate of change of volume computations. The points around the compartment periphery are used for determination of gap size and skirt drag. At each time step the height of the water surface due to seaway and vehicle generated waves is computed at the thirty five points. Over land the ground contour is used. The heights of vehicle above water, skirt clearances, and skirt motion are then computed. The relationship between compartment pressure and escape flow rate is then set up using a discharge coefficient of 0.6. The flow input to the cushion from the fan must correspond to the escape flow plus the change of compartment volume with time. If it does not, iteration is performed until the correct pressures and flow rates are achieved.

The fan flow rate and pressure lead to a power absorbed by the fans. This power plus that absorbed by the propeller on each side determines the power and torque loading on the engine at its present rpm. If the engines on a side are producing more torque than the fan and propeller absorb, the shafting accelerates during the next time step. Engine torque, power, and fuel rate characteristics are derived from figure 4 obtained from Bell Aerospace. The inertia of the engines, shafting, fan and propeller system on each side is approximately 3.53 ft. lbs. sec^2 as seen at turbine speed.

The interaction between water surface and vehicle is rather simple although the derivation of the water surface shape is extremely sophisticated. Pitch, roll, and heave forces are derived from the cushion pressures and differences in pressures between cushion compartments. Drag, sideforce and yaw moments are derived from the slope of the wave surface under each compartment, and the pressure in the compartment. Included in the computation of skirt position is the amount of skirt draging in the water. This is used with an empirical drag coefficient to derive skirt viscous drag.

The computation of water levels beneath the craft is based on a superposition of waves existing before the craft arrived (the seaway and surf) and the vehicle generated wave system. The seaway is made up of deep water



W_t	BOW DOWN	NEAR LEVEL	BOW UP
200,000	3.9	0.9	2.6
350,000	4.5	0.85	3.0

Figure 2. Pitch Moment versus Pitch Angle Over Water

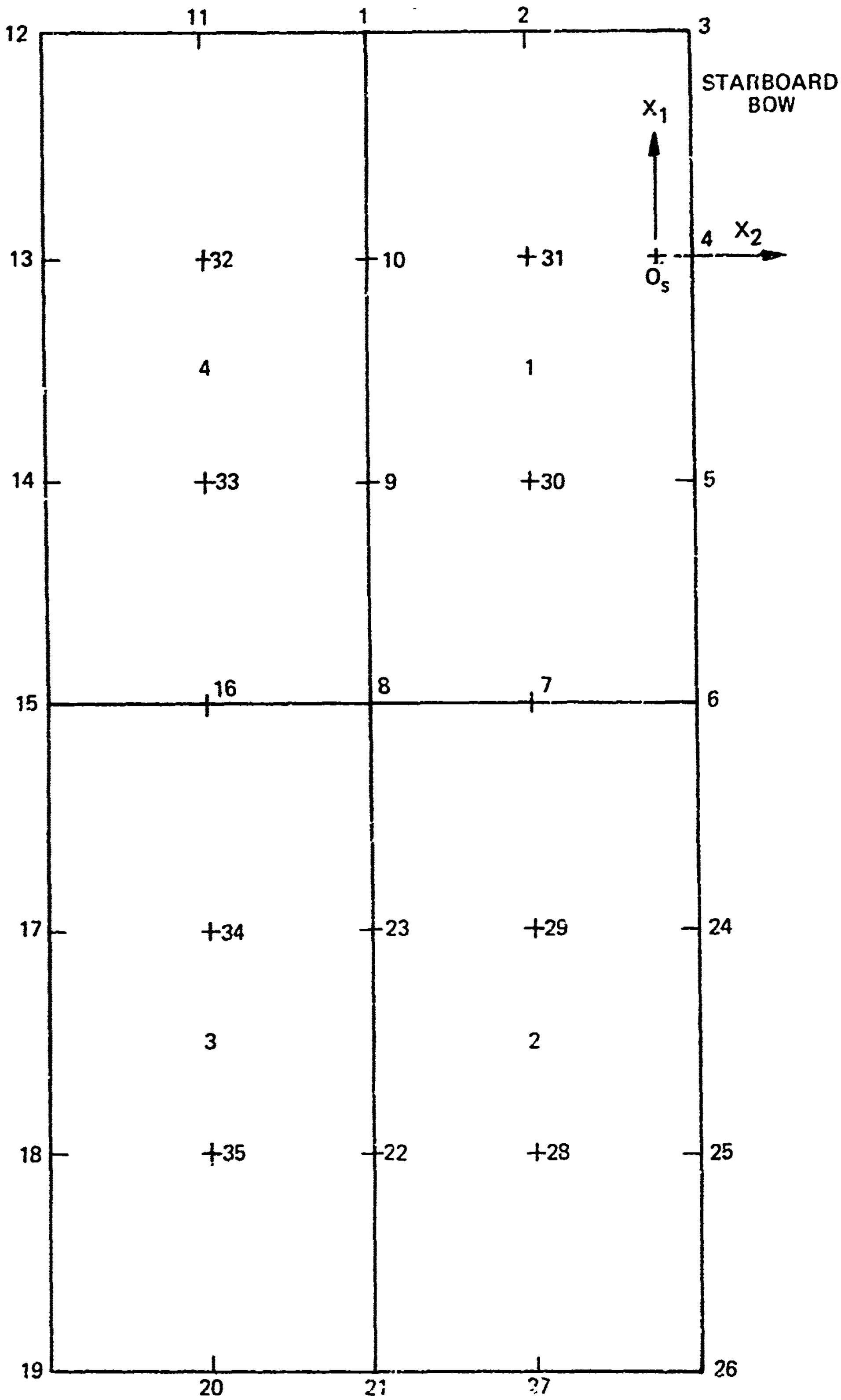


Figure 3. Location of Measurement Points

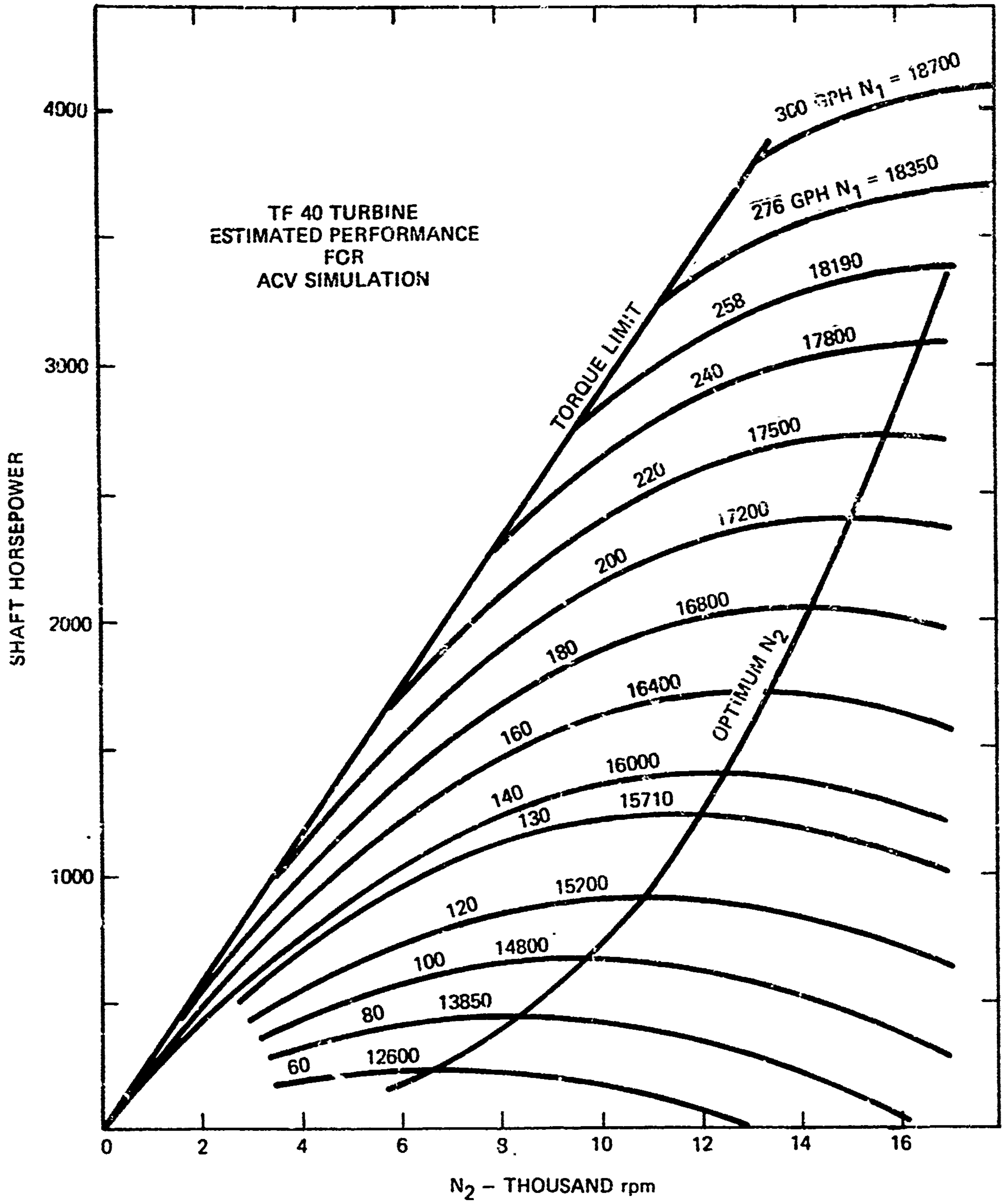


Figure 4. TF 40 Turbine Estimated Performance
For ACV Simulation

waves traveling North offshore. As they approach the beach which runs East-West, the waves gradually change length, height, and speed as a function of local water depth. When they reach very shallow water the waves gradually "break". Their amplitude is reduced and reaches zero at the shore line.

The vehicle generated wave system is made up of waves caused by the pressure distribution of the vehicle predominantly during its recent past. The wave height at each point under the craft can be computed as the sum of effects of pressure, position, and orientation over the last fifty time steps. This procedure is used for two reasons in this simulation model. First, the important output for craft motion is the total height of the wave system under the craft. The air escape and rate of change of cushion volume is dependent on these heights and the forces on the vehicle result from the interaction between surface profile and cushion pressure. The usual formulation using wave drag and "added mass" loses this information entirely. The second reason is that radical maneuvers with large accelerations and yaw rates are anticipated, causing the usual perturbation expansions for wave forces to be inadequate.

Other forces on the vehicle are derived directly from Bell PDSR plots and from their initial horizontal plane simulation. This includes curve fits for aerodynamic forces and moments, propeller forces and torque, and rudder forces.

SECTION II

COMPONENTS OF THE SIMULATION

INTRODUCTION

This section maps out the subsystems comprising the AALC simulation. A functional breakdown is given and the relationship of the components to the whole is described.

FUNCTIONAL BREAKDOWN. The functional organization of the AALC simulation follows figure 5. This simulation consists of essentially five parts:

1. Effector and Control Simulation
2. Ship Motion Simulation
3. Sensors Simulation
4. Mission Environment Simulation
5. Outputs to displays and motion actuators

Effector and Control Simulation. The controls simulated are those available to the pilot for maneuvering and propelling the ACV while on cushion. They are:

1. The throttle levers determine the fuel rate to each turbine ordered by the pilot. There are six of these levers, one for each turbine gas generator. They are referred to in the Bell literature as N1 control, for gas generator rpm control. In this model these settings are referred to as N1DES (1---6), the 6 values of desired gas generator rpm. These values are input to the engine simulation of turbine speed and power. (Item 6 in figure 5.)

2. The power turbine output shafts are linked for the starboard three turbines. Therefore, only 2 controls are required for turbine shaft output speed regulation. These two levers, referred to in the Bell literature as N2 controls, are effectively variable governors. They represent the maximum speed the pilot wishes the output shafts to turn. Of course if the torque loading is too high for the turbines to reach that speed at the present N1 throttle settings, the N2 control is ineffective. If the N2 control setting is exceeded, the throttles are shut down until the desired N2 is reached. In this report the N2 control is referred to an N2GOV (1, 2), the desired upper limits to starboard and port turbine power output shaft speeds. (Item 6 in figure 5.)

3. The rudder angles are controlled by foot pedals. There are rudders behind each of the propellers, however, they turn simultaneously. The ordered rudder angle is referred to an RCONT in the simulation. (Item 23 in figure 5.)

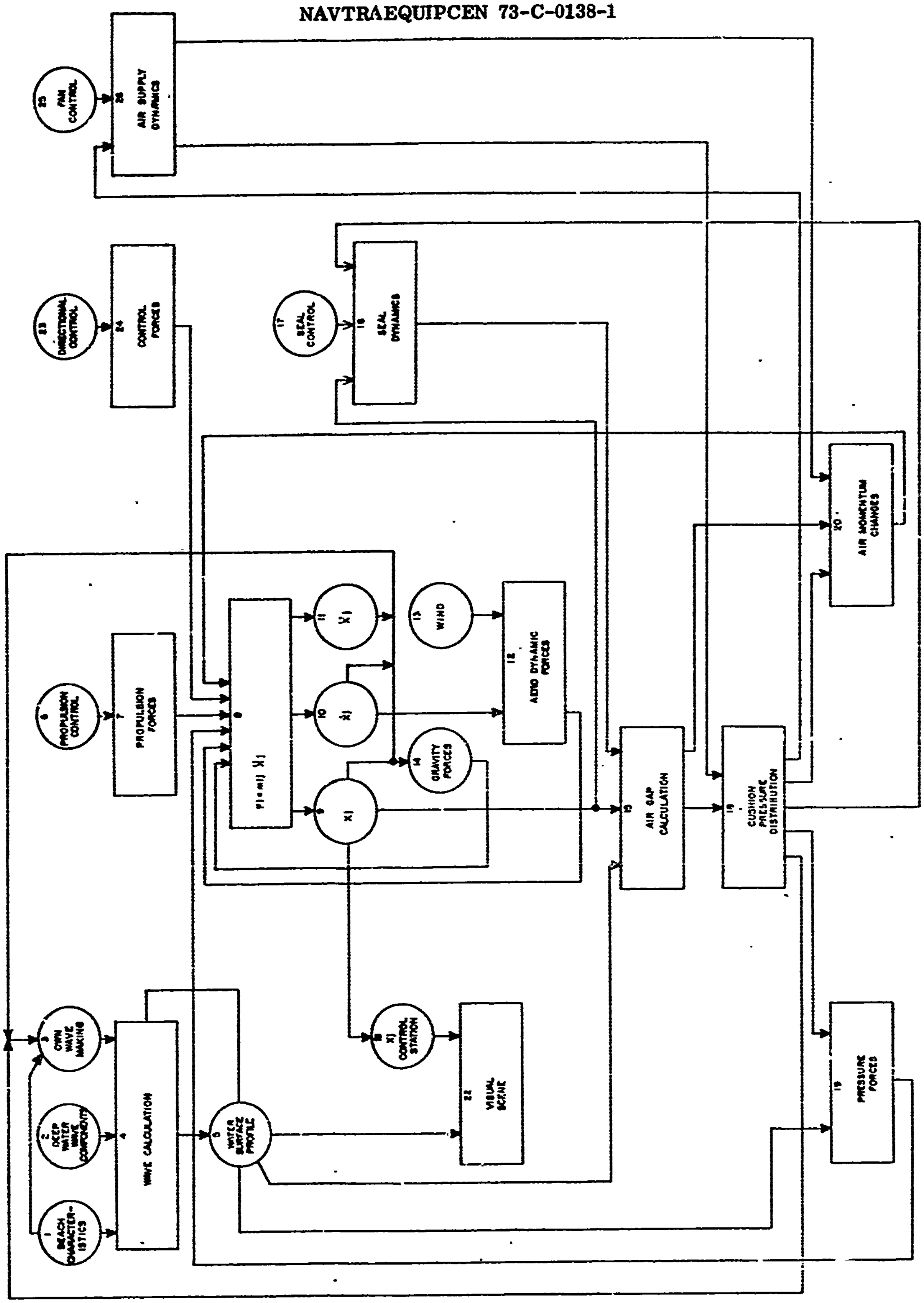


Figure 5. ACV Simulation - Broad Brush Interactions

4. The thrust nozzles are controlled by two cabin controls. The pilots steering wheel rotates from -30° to $+30^{\circ}$. This angle, called ASTEER in the simulation controls the thrust nozzle angle through a 180° range. The other control, called SWITCH, is a Boolean switch. If its value is negative, the nozzles push within 90° of ahead. If positive the nozzles push within 90° of aft. (Items 6 and 23 in figure 5.)

5. The propeller blade pitch control is by levers for the starboard and port propellers. The controls are referred to as β control in the Bell literature. In this report the propeller pitches commanded by the pilot are called APROP (1,2), ordered blade pitch on the starboard and port propellers. (Item 6 in figure 5.)

The effectors cause forces and moments on the vehicle which depend on their state, their control, and their environment. The effectors simulated are as follows:

1. Propellers. The propeller thrust and torque as a function of rpm, pitch angle, and inflow velocity are curve fits based on the Bell preliminary horizontal plane simulation. (Item 7 in figure 5.)

2. Rudders. The rudder forces and moments as a function of rudder angle, propeller slipstream velocity, and vehicle velocity are curve fit based on the Bell preliminary horizontal plane simulation. (Item 24 in figure 5.)

3. Thrust Nozzles. The forces and moments exerted on the vehicle by the thrust nozzles are derived from the efflux of momentum in the nozzle air discharge.

4. Engines and fans. The engine and fan characteristics are derived from data in the Bell PDSR. (Items 7 and 26 in figure 5.)

Ship Motion Simulation. The mathematical formulation of the equations of motion (Item 8 in figure 5) of the vehicle is in standard form for ship motion and maneuvering computations. However, the determination of the hydrodynamic forces acting on the vehicle due to its state and motion is vastly different from displacement surface ship analysis. The two basic reasons for this are:

1. The ACV is supported by a cushion of air whose pressure is a function of the motion as well as fan and powering system dynamics. Therefore, relatively simple hydrostatics calculations which determine equilibrium conditions on a displacement vessel are replaced by a sophisticated pump and flow problem. This problem includes variable system geometry since both the seals and the water surface deform in response to cushion pressure changes.

2. The hydrodynamic interaction between vehicle and water surface is via a boundary condition on pressure rather than on normal velocity. Although this fact simplifies the mathematics of computation of the wave system and its interaction with the vehicle, it does not allow the use of the mathematics and empirical data developed for hydrodynamic exciting forces and restoring forces on displacement surface ships. The forces due to the water surface and cushion air system are therefore derived from the interaction between cushion fan and flow relations and the water surface profile. The water surface at each point is the sum of the wave height caused by the vehicle's pressure distribution, the vehicle generated waves, and the wave height present when the vehicle arrived, the seaway. If the vehicle is over land, the wave height calculations are skipped and the land elevations used. (Item 1 in figure 5.)

The seaway is composed of sinusoidal components in deep water running North toward the beach which runs East-West (Item 2 in figure 5.) As the waves approach the beach they gradually shorten, slow, steepen, and finally break. Values of the wave height at 35 points beneath the vehicle are continuously generated to input to the ship forces and motions. The seaway model is incidently capable of generating wave heights at any other point relative to the vehicle, if computer generated TV simulation of so complicated a picture should become feasible. At present, optical techniques are anticipated for the visual simulation.

The vehicle generated wave system is calculated by integrating the waves generated by the craft's moving pressure distribution over the last fifty time steps. This is perhaps the most technically challenging aspect of the simulation, and a large part of this report is devoted to its derivation and use. The waveheights generated by this mathematical model (Item 3 in figure 5) are input to the wave calculation.

The wave calculation (Item 4 in figure 5.) adds the wave heights due to seaway and motion to give the water surface profile under the craft as a function of time (Item 5 in figure 5).

The computation of cushion pressure forces on the vehicle (Item 19 in figure 5.) is simple, although the computation of the pressures is complex. Given the pressures in each cushion and assuming uniform pressure in each cushion, the horizontal plane forces and moments are given by the pressure and differences in wave height at the skirts. The vehicle is effectively climbing water hills. The analysis is identical ashore, except that the ground does not deflect in response to the vehicle motion. Heave, pitch, and roll excitation is derived from the cushion pressures and planform areas of each compartment.

<u>Air Cushion System Simulation</u>. The dynamics of the fans, thrust nozzles, cushion compartments and skirts make up the air cushion system model. The pressure in each compartment is determined by finding an equilibrium between

the cushion-skirt pressure and flow characteristics which are functions of the skirt gaps and vehicle motion, and the fan curves, which are a function of pressure, flow, and fan rpms. The fan curves and ducting are derived from the Bell PDSR (Item 26 in figure 5). The seal dynamics (Item 16 in figure 5) combine with the vehicle attitude and wave heights to produce the skirt gaps (Item 15 in figure 5). The fan and skirt gap pressure flow relations are combined to find the pressure in each cushion (Item 18 in figure 5). These pressures and flows are used to determine nozzle thrusts and intake air momentum drag (Item 20 in figure 5) as well as the forces due directly to the pressure distribution (Item 19 in figure 5).

Sensors Simulation. Sensors of ship motion and external environment available to the pilot are included in the mathematical model and output at each time step for D/A conversion. These sensors include:

1. Heading Angle. Rotation clockwise from North is integrated from the equations of motion and is called X(6) in the simulation (Item 9 in figure 5).

2. Ahead speed is output in feet per second from integrations of the equations of motion in the boat coordinate system. Small pitch angles are assumed and this value is not corrected for pitch. It is called XD(1) in the simulation (Item 10 in figure 5).

3. Sideslip speed is output in feet per second from integration of the equations of motion in the boat coordinate system. Roll angles are assumed to be small and this value is not corrected for roll for the indicator input. It is called XD(2) in the simulation (Item 10 in figure 5).

4. A height above water transducer is assumed attached to the hull bottom plating at the center of the ship. This measurement is available with height values at 34 other points from the simulation if needed. It is called HOW(8) in the simulation (Item 5 in figure 5).

5. Roll Angle. The angle of roll as seen in the ship's coordinate system is integrated from the equations of motion. It is called X(4) in the simulation (Item 9 in figure 5).

6. Pitch Angle. The angle of pitch is integrated from the equations of motion and is called X(5) in the simulation (Item 9 in figure 5).

7. The angle of the rudders is calculated as output from the rudder control (Item 23 in figure 5) and is called RANGLE in the simulation

8. The thrust nozzle angles are also calculated as output from directional control (Item 23 in figure 5) and are called PSI in the simulation.

9. The velocity of the wind relative to the boat is calculated from the input wind and the craft horizontal plane velocities (Items 13 and 12 in figure 5). It is called VA in the simulation. At the same time, the apparent wind angle, clockwise from the bow, is output. It is called ABETA.

10. The propeller pitch angles are output from the propulsion control logic (Item 6 in figure 5). They are called BETA(1) and BETA(2) in the simulation for starboard and port propellers.

11. The values of gas generator rpm are output from the propulsion forces calculation (Item 7 in figure 5). N1(1), N1(2), and N1(3) are the values for the starboard side turbines and N1(4) to N1(6) are the port side speeds. All engine calculations are done by a subroutine called ENGINE.

12. The values for starboard and port power turbine rpm are also calculated in the propulsion forces section (Item 7 in figure 5.) They are called N2(1) and N2(2). These port and starboard power turbine speeds may also be used to indicate fan and propeller rpms since they are connected by gearing.

13. The fuel flow rates in gallons per hour output from the propulsion forces section (Item 7 in figure 5). They are directly related to N1(1---6), the gas generator speeds. They are called GPH(1) to GPH(6). The time integral of these rates gives total fuel used.

Mission Environment Simulation. The motions and maneuvers of the craft are calculated from the equations of motion in a vehicle fixed coordinate system with origin at the pilot. However, the environment external to the vehicle is expressed in an earth fixed coordinate system with origin at the shoreline. These systems and their relationship are described in Section IV. The seaway and surf, the land, and the wind make up the external environment inputs to the vehicle. It is assumed that the seaway is made up of sinusoidal waves offshore which are combined with the vehicle-generated wave system to model the total height of the water beneath the craft. The steepening and breaking of the waves as they approach the beach is modeled. Over land the seaway and wave calculations are omitted and ground elevations used directly. The wind velocity and direction are input to the simulation.

SECTION III

MATHEMATICAL MODEL

INTRODUCTION

The mathematical model is intended for implementation on a Sigma 7 digital computer at the NAVTRAEQUIPCEN in Orlando, Florida. Digital to analog, and analog to digital interfaces exist for motion, control and sensor simulation. As the model has been developed, it has been programmed in FORTRAN IV and run on the Draper Laboratory IBM 360. It is assumed that this program will be of use to the simulation programmers. It should be pointed out, however, that it was intended as a test of the mathematical model as the pieces were developed and no attention has been paid to efficiency or running time. It is part of a mathematical rather than a programming task and was written by engineers rather than programmers. The real programming task, therefore, remains to be done.

EQUATIONS OF MOTION.

BASIC ASSUMPTIONS. The equations of motion for the ACV simulation are the standard six degree of freedom equations used for ship motion and control analysis. They are fully derived in the standard Naval Architecture texts, the most complete being Stability and Control of Ocean Vehicles by Martin A. Abkowitz, MIT Press 1969. For ACV modeling purposes the origin of coordinates is taken at the pilot location since that is the point of interest for the simulation. Pitch and roll angles are assumed small and the appropriate linearizations made in trigonometric functions of these angles.

The water and land surface are of course developed in an earth fixed level coordinate system with the vertical axis truly vertical. This creates slight mismatches between heights on the vehicle and ground or water elevations. However, for small roll and pitch the errors are negligible and not worth the computer time to correct.

Coordinate Transformations. The two coordinate systems employed for the simulation and the relationship between them are described below:

Definition of Terms

$\overline{es}_1$, $\overline{es}_2$, $\overline{es}_3$	unit vectors forward, to starboard, and down on the craft.
$\overline{eo}_1$, $\overline{eo}_2$, $\overline{eo}_3$	unit vectors North, East and down.
x_1, x_2, x_3	distances from ship based origin (Pilot cabin) in $\overline{es}_1$, $\overline{es}_2$, $\overline{es}_3$.
xo_1, xo_2, xo_3	distances from earth fixed origin (shoreline at mean sea level) in $\overline{eo}_1$, $\overline{eo}_2$, $\overline{eo}_3$.
$\dot{x}_j$, $\ddot{x}_j$	first and second time derivatives of x_j.
$\dot{xo}_j$	time derivative of xo_j.
x_4, x_5, x_6	rotations about $\overline{es}_1$, $\overline{es}_2$, $\overline{es}_3$ axes. (roll, pitch, yaw) x_6 is the heading angle measured clockwise from North. x_4 and x_5 are assumed small.
m	mass of vehicle.
I_1, I_2, I_3	moments of inertia, assumed to be principal moments, about $\overline{es}_1$, $\overline{es}_2$, $\overline{es}_3$ axes at ship based origin.
g	acceleration of gravity.
xs_{1c}, xs_{2c}, xs_{3c}	position of center of gravity in ship based coordinates.
FT_j	jth component of total force vector acting on vehicle exclusive of gravitational forces which are included in the equations of motion.

Coordinates

The transformation from earth to ship system is defined by:

$$\overline{es}_i = TSO_{ij}\,\overline{eo}_j \tag{1}$$

$$\overline{eo}_j = TSO_{ij}\,\overline{es}_i \tag{2}$$

Where,

$TSO_{11} = \cos x_6$	$TSO_{12} = \sin x_6$	$TSO_{13} = -x_5$
$TSO_{21} = -\sin x_6$	$TSO_{22} = \cos x_6$	$TSO_{23} = x_4$
$TSO_{31} = x_5 \cos x_6 + x_4 \sin x_6$	$TSO_{32} = x_5 \sin x_6 - x_4 \cos x_6$	$TSO_{33} = 1.$

The location of a point p whose location on the ship is defined by xs_1, xs_2, xs_3 relative to the pilot cabin is therefore expressed as follows in navigational coordinates:

$$\begin{aligned}
xo_{p1} &= xo_1 + xs_1 \cos x_6 - xs_2 \sin x_6 + xs_3\,(x_5 \cos x_6 + x_4 \sin x_6) \\
xo_{p2} &= xo_2 + xs_1 \sin x_6 + xs_2 \cos x_6 + xs_3\,(x_5 \sin x_6 - x_4 \cos x_6) \\
xo_{p3} &= xo_3 - xs_1 x_5 - xs_2 x_4 + xs_3.
\end{aligned} \tag{3}$$

If xs_1 and $xs_2 \gg xs_3$

$$\begin{aligned}
xo_{p1} &= xo_1 + xs_1 \cos x_6 - xs_2 \sin x_6 \\
xo_{p2} &= xo_2 + xs_1 \sin x_6 + xs_2 \cos x_6 \\
xo_{p3} &= xo_3 - xs_1 x_5 + xs_2 x_4 + xs_3.
\end{aligned} \tag{4}$$

Equations of Motion. The equations of motion are integrated using Euler integration. The acceleration of the pilot location is represented by:

$$\begin{aligned}\ddot{X}_1 = {} & TFX_1/VMASS - \dot{X}_5\dot{X}_3 + \dot{X}_6\dot{X}_2 + XS_{1c}\left|\dot{X}_5^{\,2} + \dot{X}_6^{\,2}\right| \\ & - XS_{2c}\left|\dot{X}_4\dot{X}_5 - \ddot{X}_6\right| - XS_{3c}\left|\dot{X}_4\dot{X}_6 + \ddot{X}_5\right|\end{aligned} \tag{5}$$

$$\begin{aligned}\ddot{X}_2 = {} & TFX_2/VMASS - \dot{X}_6\dot{X}_1 + \dot{X}_4\dot{X}_3 + XS_{2c}\left|\dot{X}_6^{\,2} + \dot{X}_4^{\,2}\right| \\ & - XS_{3c}\left|\dot{X}_5\dot{X}_6 - \ddot{X}_4\right| - XS_{1c}\left|\dot{X}_5\dot{X}_4 + \ddot{X}_6\right|\end{aligned}$$

$$\begin{aligned}\ddot{X}_3 = {} & TFX_3/VMASS - \dot{X}_4\dot{X}_2 + \dot{X}_5\dot{X}_1 + XS_{3c}\left|\dot{X}_4^{\,2} + \dot{X}_5^{\,2}\right| \\ & - XS_{1c}\left|\dot{X}_6\dot{X}_4 - \ddot{X}_5\right| - XS_{2c}\left|\dot{X}_6\dot{X}_5 + \ddot{X}_4\right|\end{aligned}$$

$$\begin{aligned}\ddot{X}_4 = {} & \Big[TFX_4 - \dot{X}_5\dot{X}_6\left|VI_3 - VI_2\right| - VMASS\ \Big|XS_{2c}(\ddot{X}_3 + \dot{X}_4\dot{X}_2 - \\ & \dot{X}_5\dot{X}_1) - XS_{3c}(\ddot{X}_2 + \dot{X}_6\dot{X}_1 - \dot{X}_4\dot{X}_3) + XS_{1c} \times XS_{2c} \\ & (\dot{X}_4\dot{X}_6 - \ddot{X}_5) - XS_{1c} \times XS_{3c}(\dot{X}_4\dot{X}_5 + \ddot{X}_6) + XS_{2c} \times XS_{3c} \\ & (\dot{X}_6^{\,2} - \dot{X}_5^{\,2})\Big|\Big] / VI_1\end{aligned}$$

$$\begin{aligned}\ddot{X}_5 = {} & \Big[TFX_5 - \dot{X}_6\dot{X}_4\left|VI_1 - VI_3\right| - VMASS\ \Big|XS_{3c}(\ddot{X}_1 + \dot{X}_5\dot{X}_3 - \dot{X}_6\dot{X}_2) \\ & - XS_{1c}(\ddot{X}_3 + \dot{X}_4\dot{X}_2 - \dot{X}_5\dot{X}_1) + XS_{2c} \times XS_{3c}(\dot{X}_5\dot{X}_4 - \ddot{X}_6) - XS_{1c} \times XS_{2c}(\dot{X}_5\dot{X}_6 + \ddot{X}_4) \\ & + XS_{3c} \times XS_{1c}(\dot{X}_4^{\,2} - \dot{X}_6^{\,2})\Big|\Big] / VI_2\end{aligned}$$

$$\ddot{X}_6 = \Big| TFX_6 - \dot{X}_4 \dot{X}_5 \left\{ VI_2 - VI_1 \right\} - VMASS \Big\{ XS_{1c}(\ddot{X}_2 + \dot{X}_6 \dot{X}_1 - \dot{X}_4 \dot{X}_3)$$

$$- XS_{2c}(\ddot{X}_1 + \dot{X}_5 \dot{X}_3 - \dot{X}_6 \dot{X}_2) + XS_{3c} \times XS_{1c} (\dot{X}_6 \dot{X}_5 - \ddot{X}_4)$$

$$- XS_{3c} \times XS_{2c}(\dot{X}_6 \dot{X}_4 + \ddot{X}_5) + XS_{1c} \times XS_{2c}(\dot{X}_5^{\,2} - \dot{X}_4^{\,2}) \Big\} \Big| \; / VI_3$$

The integration proceeds as follows:

$$DUM_N = \dot{X}_N \tag{6}$$

$$\dot{X}_N = \dot{X}_N + \ddot{X}_N \quad \Delta t$$

$$X_N = X_N + 0.5 \quad \left\{ DUM_N + \dot{X}_N \right\} \Delta t .$$

For the pilot location in navigational coordinates: (For navigational purposes heave velocity contributes negligibly to location)

$S_6 = \sin (X_6) \rightarrow$ sine of heading angle

$C_6 = \cos (X_6) \rightarrow$ cosine of heading angle

$\dot{X0}_1 = \dot{X}_1 \times C_6 - \dot{X}_2 \times S_6 \rightarrow$ North pilot velocity (7)

$\dot{X0}_2 = \dot{X}_1 \times S_6 + \dot{X}_2 \times C_6 \rightarrow$ East pilot velocity

$X0_1 = X0_1 + \dot{X0}_1 \Delta t \rightarrow$ New pilot North of start.

$X0_2 = X0_2 + \dot{X0}_2 \Delta t \rightarrow$ New pilot East of start

$$HPX01_N = X0_1 + HPXS1_N \times C_6 - HPXS2_N \times S_6 \tag{8}$$

$\rightarrow$ New North location of position N on hull for determination of seaway height.

The following characteristics and their values used at present are below:

VMASS	-	mass of vehicle 10879.5 slugs
X1SC	-	X location of cg -30.0 feet
X2SC	-	Y location of cg -18.0 feet
X3SC	-	Z location of cg + 8.0 feet
CSKRT	-	skirt stiffness parameter 0.0112
TC	-	skirt reaction time 8.0
ISYS	-	moment of inertia of rotating machinery 3.533
XIR	-	X location of rudders -67.1 feet
X2R (1, 2)	-	Y location of starboard, port rudders -4.0 feet -32.0 feet
X3R	-	Z location of rudders 0.0 feet
XIN	-	X location of nozzles -23.16 feet
X2N (1, 2)	-	Y location of nozzles -0.25 feet, -35.75 feet
X3N	-	Z location of nozzles -3.0 feet
X2P (1, 2)	-	Y location of propellers -4.0 feet -32.0 feet
X3P	-	Z location of propellers 0.0 feet
VWIND	-	Wind velocity 34.0 feet/sec
AWIND	-	Wind direction
RH0	-	density of air 0.00237 slugs/ft^3
VI1	-	vehicle moment of inertia about 1 axis = 5.672×10^6
VI2	-	vehicle moment of inertia about 2 axis = 1.629×10^7
VI3	-	vehicle moment of inertia about 3 axis = 2.057×10^7
XO1	-	initial location of vehicle pilot (North) - feet
XO2	-	initial location of vehicle pilot (East) - feet
SCAREA	-	planform area for windage = 3200 ft^2
FCAREA	-	frontal area for windage = 836 ft^2
LCUSH	-	length of cushion = 77 ft
AE	-	duct area = 123 ft^2
DIA	-	duct diameter = 11.25 ft
CHORD	-	duct chord = 4.67 ft
RSAREA	-	rudder (lifting area = 47.2 ft^2

SDOLD (1---27)	-	initial skirt depths - 4.5 for periphery, 4.0 for dividers
PFAN (1---4)	-	initial fan pressures - 300 psf
QFAN (---)	-	initial fan flows - 7000 ft^3/sec
QOC (1---4)	-	initial flow under skirts - 3000 ft^3/sec per compartment
PGC (1---4)	-	initial cushion pressure - 109. psf.

<u>Forces on the Vehicle</u>. The force and moment components on the vehicle are expressed as follows:

$$\begin{aligned}
TFX1 &= TPROP_1 + TPROP_2 - (TNOZ_1 + TNOZ_2)\ COS\ (PSI) \\
&+ RDRAG + XBDRAG + XMDRAG + WDRAGX + SDRX + XGRAV
\end{aligned} \tag{9}$$

$$\begin{aligned}
TFX2 &= -(TNOZ_1 + TNOZ_2)\ SIN\ (PSI) + YBDRAG + RLIFT + YMDRAG \\
&+ YDUCT + WDRAGY + SDRY + YGRAV
\end{aligned}$$

$$TFX3 = FX3CT + ZGRAV$$

$$\begin{aligned}
TFX4 &= FX4CT + ROLLA + ROLLMD + ROLLD + SIN(PSI)\ (TNOZ_1 \\
&+ TNOZ_2)\ X3N + WMX4 + RROLL + KGRAV + SDRR
\end{aligned}$$

$$\begin{aligned}
TFX5 &= FX5CT + PITCHA + PITCHM + RPITCH - COS\ (PSI) \\
&\quad (TNOZ_1 + TNOZ_2)\ X3N + (TPROP_1 + TPROP_2)\ X3P \\
&+ WMX5 + SDRP + MGRAV
\end{aligned}$$

$$\begin{aligned}
TFX6 &= -SIN\ (PSI)\ (TNOZ_1 + TNOZ_2)\ X1N + COS\ (PSI)\ (TNOZ_1 \times X2N_1 \\
&+ TNOZ_2 \times X2N_2) - X2P_1 \times TPROP_1 - X2P_2 \times TPROP_2 \\
&+ YAWBD + YAWMD + YAWD + RYAW + YAWDC \\
&+ WMX6 + SDRYM + NGRAV
\end{aligned}$$

where,

TFX1--- TFX6 = Total force and moment components in directions X_1 to X_6

$TPROP_{1,2}$ = Propeller thrusts

$TNOZ_{1,2}$ = Nozzle thrust

PSI = Nozzle angle

RDRAG = force on rudder in $\overline{es_1}$, direction (normally negative)

XMDRAG	=	Force due to intake air momentum change in $\overline{es}_1$ direction (normally negative)
XBDRAG	=	Aerodynamic force in $\overline{es}_1$ direction (normally negative)
WDRAGX	=	Force due to wave-cushion pressure interaction in $\overline{es}_1$ (normally negative)
SDRX	=	Skirt drag force in $\overline{es}_1$ direction (normally negative)
YBDRAG	=	Aerodyanmic force in $\overline{es}_2$ direction
RLIFT	=	Force on rudder in $\overline{es}_2$ direction
YMDRAG	=	Force due to intake air momentum change in $\overline{es}_2$ direction
YDUCT	=	Force on ducts in $\overline{es}_2$ direction
WDRAGY	=	Force due to wave-cushion pressure interaction in $\overline{es}_2$ direction
SDRY	=	Skirt drag force in $\overline{es}_2$ direction
FX3CT	=	Force on vehicle due to cushion pressure in $\overline{es}_3$ direction
FX4CT	=	Moment on vehicle due to cushion pressure in roll direction
ROLLA	=	Roll moment due to aerodynamic forces
ROLLMD	=	Roll moment due to intake air momentum change
ROLLD	=	Roll moment due to duct side force
X3N	=	Vertical position of nozzle
WMX4	=	Roll moment due to wave-cushion pressure interaction

SDRR = Roll moment due to skirt drag

FX5CT = Pitch moment due to cushion pressure

PITCHA = Pitch moment due to aerodynamic forces

PITCHM = Pitch moment due to intake air momentum change

RPITCH = Pitch moment due to forces on rudder

X3P = Longitudinal position of propellers

WMX5 = Pitch moment due to wave-cushion pressure interaction

SDRP = Pitch moment due to skirt drag

YAWBD = Yaw moment due to aerodynamic forces

YAWMD = Yaw moment due to intake air momentum change

YAWD = Yaw moment due to duct sideforce

RYAW = Yaw moment exerted by rudders

YAWDC = Yaw damping coefficient

WMX6 = Yaw moment due to wave-cushion pressure interaction

SDRYW = Yaw moment due to skirt drag

RROLL = Roll moment due to rudder forces

XGRAV = Gravity force in $\overline{es}_1$ direction ($-mgX_5$) (10)

YGRAV = Gravity force in $\overline{es}_2$ direction (mgX_4)

ZGRAV = Gravity force in $\overline{es}_3$ direction (mg)

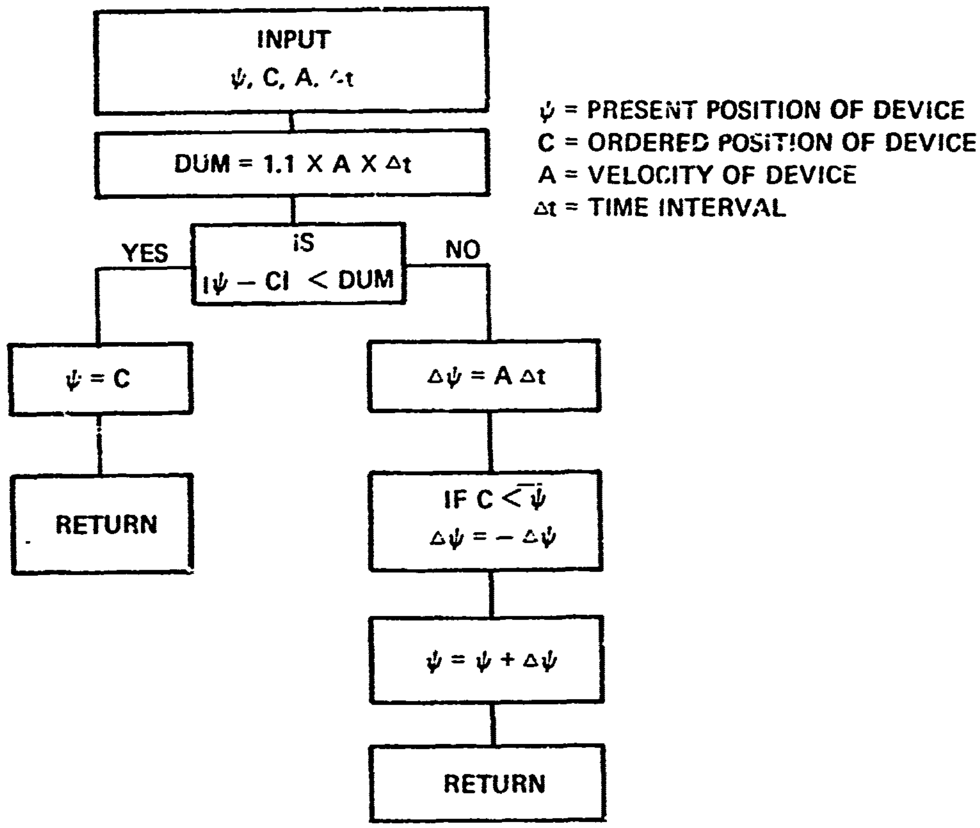
KGRAV = Gravity moment in $\overline{es}_4$ (ZGRAVxX2SC-YGRAVxX3SC)

MGRAV = Gravity moment in $\overline{es}_5$ (XGRAVxS3SC-ZGRAVxX1SC)

NGRAV = Gravity moment in $\overline{es}_6$ (YGRAVxX1SC-XGRAVxX1SC).

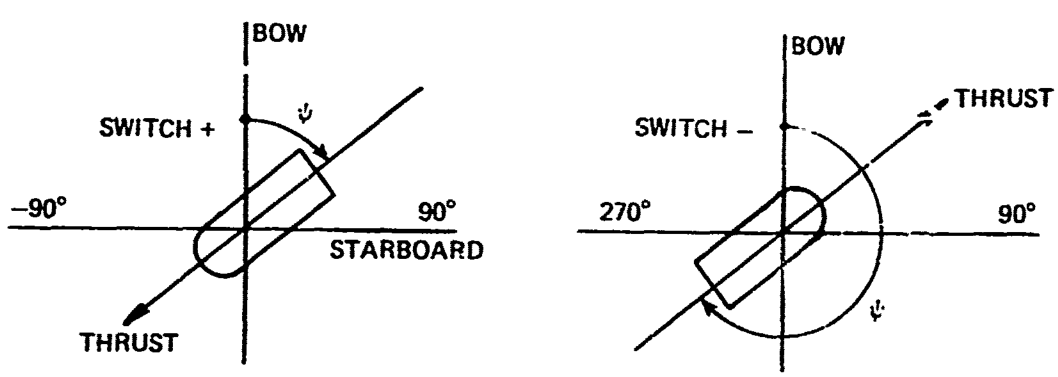
MODELING OF EFFECTORS AND CONTROLS

EFFECTOR CONTROL LOGIC. Angle Control Logic. For controls activated by the pilot the routine STEER is used to compute the response to pilot input. STEER works as follows:



The actuator (ψ) moves toward its ordered position (C) at rate (A).

The nozzle angle PSI is controlled in two modes. If the mode SWITCH is positive, angles between -90° and $+90^{\circ}$ are intended, and the thrust is aft.



If the mode switch is negative, angles between 90° and 270° are intended, and the thrust is forward. The ordered control, besides switch, is the steering wheel, which operates between -30° and $+30^{\circ}$. The steering wheel angle is ASTEER. When SWITCH is positive 3 x ASTEER is the ordered angle. When SWITCH is negative, 3 x ASTEER + 180° is the ordered angle. The logic is handled by a dummy variable PS which represents the nozzle angle PSI in the SWITCH + mode, and the nozzle angle less 180° in the SWITCH - mode. Therefore, the logic is:

PS = PSI (11)
If SWITCH is negative, PS = PSI - 180°
call STEER (PS, 3 x ASTEER, ACONST, DELT)
If SWITCH is negative, PSI = PS + 180°

ACONST, the angular velocity of the nozzle, is 50°/sec.

For the rudder angle RANGLE, the input from the pilot is RCONT, the desired rudder angle from the foot pedal controls. STEER is called with:

(RANGLE, RCONT, RCONST, DELT). RCONST, the rudder rate, is 30°/sec.

For the propeller pitch angles, the input from the pilot is APROP(1) and (2), the starboard or port pitch lever positions STEER is called with:

(BETA (N), APROP (N), PCONST, DELT). PCONST, the pitch rate, is 20°/sec.

Machinery Control. Variable definitions:

NIDES(1) --- N1DES(6) →	Desired gas producer control rpm set by pilot on each turbine.
$N^2GOV(1)$ --- $N^2GOV(2)$ →	Maximum power turbine control rpm set by pilot, starboard and port.
N1(1) --- N1(6) →	Actual gas producer rpms on each turbine.
N2(1) --- N2(2) →	Actual power turbine shaft rpm, starboard and port.
S1CON →	Time constant of N1 control (input, .3 sec^{-1}).
S2CON →	Time constant of N2 governor control (input, .5 sec^{-1}).

Machinery Control Logic. The pilot controls on the turbines consist of a desired fuel flow rate on each turbine (N1DES) and a governing (upper limit) control on the starboard and port power turbine output rpm (N2GOV). If N2GOV is exceeded the corresponding fuel rates are decreased.

Logic $J = 1$ (for starboard side), $n = 1, 2, 3$
$J = 2$ (for port side), $n = 4, 5, 6$

$$\text{if} \quad N2_j < N2GOVj \tag{12}$$

$$N1_n = N1_n + S1CON \left\{ N1DES_n - N1_n \right\} \Delta t$$

$$\text{if} \quad N2_j \geq N2GOVj$$

$$N1_n = N1_n + S2CON \left| N2GOV_n - N2_n \right| \Delta t.$$

Propeller Modeling. The thrust and torque characteristics of the propellers are based entirely on the Bell horizontal plane simulation. This is described in their Preliminary Design Summary Report, appendix to paragraph 2.2.2, "Three-Degree-Of-Freedom Maneuverability and Control Simulation." Many of the effector and control descriptions in the present mathematical model are taken from this Bell Aerospace simulation. In the notation of the present simulation:

For propeller thrust:

if $\beta j \geq 0.0$

$$TPROPj = \left| 338.\ \beta j + 4.36\ \beta j^2 - 0.1715\ X WIND^2 - 1.43\ \beta j XWIND \right| \left| NPROPj/1250 \right|^2 \tag{13}$$

if $\beta j < 0.0$

$$TPROPj = \left| 60.\ \beta j - .1715\ X\ WIND^2 \right| \left| NPROPj/1250 \right|^2.$$

For propeller power:

if $\beta j \geq 10.$, $C1 = 0.0$

if $\beta j < 10.$, $C1 = 50 \left| X\ WIND/127.5 \right| \left| 10 - \beta j \right|$

$$HPROPj = (NPROPj/1250.)^3. \left| 450. + 23.05\ \beta j + 2.56\ \beta j^2 + C1 \right| \tag{14}$$

where

βj	=	Pitch angle of propeller j
j	=	1 for starboard, 2 for port
$TPROP_j$	=	Thrust in pounds on propeller j
XWIND	=	Apparent head wind velocity

$NPROP_j$ = RPM of propeller j

$HPROP_j$ = Horsepower absorbed by propeller j.

The forces and moments exerted by the propellers are:

$$\text{in } ES_1 \text{ direction} \quad \sum_{j=1}^{2} TPROP_j \tag{15}$$

$$\text{in yaw direction} \quad \sum_{j=1}^{2} -TPROP_j \times X2P_j. $$

Thrust Nozzles. The thrust exerted by each nozzle is derived directly from the momentum flux of the air exciting from the nozzle. The effect of relative wind velocity on this force is considered small.

$$TNOZ_j = 2.44 \times 10^{-4} QNOZ_j^2 \tag{16}$$

where

TNOZ = Thrust in pounds on nozzle j

j = 1 for starboard, 2 for port

$QNOZ_j$ = Flow through nozzle j (ft/sec).

The flow through the nozzle is obtained directly from the fan pressure:

$$QNOZ_j = -346. \sqrt{|PFAN_j|} \text{ sgn } (PFAN_j) \tag{17}$$

where

$PFAN_j$ is the pressure developed by fan j in psf.

Rudders. The forces exerted on the rudders are based on the Bell PDSR simulation. First, the velocity at the rudders including vehicle motion, effect of the ducted propellers, and wind must be found. The square of the axial velocity at rudder j is:

If $TPROP_j \geq 0$

$$VDUCT2_j = (VA \cos\beta aa)^2 + TPROP_j/(\rho AE) \tag{18}$$

If TPROPj < 0

$$VDUCT2_j = (VA \cos\beta aa)^2 + TPROP_j/(2\rho AE)$$

where,

VA	=	Apparent wind velocity in ft/sec
βaa	=	Apparent wind angle in radians
$TPROP_j$	=	Thrust on propeller j
j	=	1 for starboard, 2 for port
	=	Air density in slugs/ft^3
AE	=	Duct area in ft^2 (123 ft^2).

The dynamic pressure available at the rudders is then

$$PDUCT_j = 1/2\, \rho\, VDUCT2_j. \qquad (19)$$

The rudder lift and drag coefficients for a rudder angle, RANGLE (in degrees) are:

$$\begin{aligned} CLIFT &= .053\ RANGLE \\ CDRAG &= .02 + .422 \times 10^{-3}\ RANGLE^2 \end{aligned} \qquad (20)$$

in the small angle of attack range. Stagnation is assumed to occur at 20^o:

If RANGLE > 20^o, CLIFT = 1.06.

The longitudinal and side forces due to the rudders are then:

$$\begin{aligned} RDRAG &= -CDRAG \times RSAREA \times (PDUCT_1 + PDUCT_2) \\ RLIFT &= CLIFT \times RSAREA \times (PDUCT_1 + PDUCT_2) \end{aligned} \qquad (21)$$

where RSAREA is the rudder area (47.2 ft^2).

Due to the location of the rudders relative to the pilot location origin yaw, roll and pitch moments are developed:

$$\begin{aligned} RYAW &= CDRAG \times RSAREA\ (X2R_1 \times PDUCT_1 + X2R_2 \\ &\quad \times PDUCT_2) + X1R \times RLIFT \\ RROLL &= -X3R \times RLIFT \\ RPITCH &= +X3R \times RDRAG. \end{aligned}$$

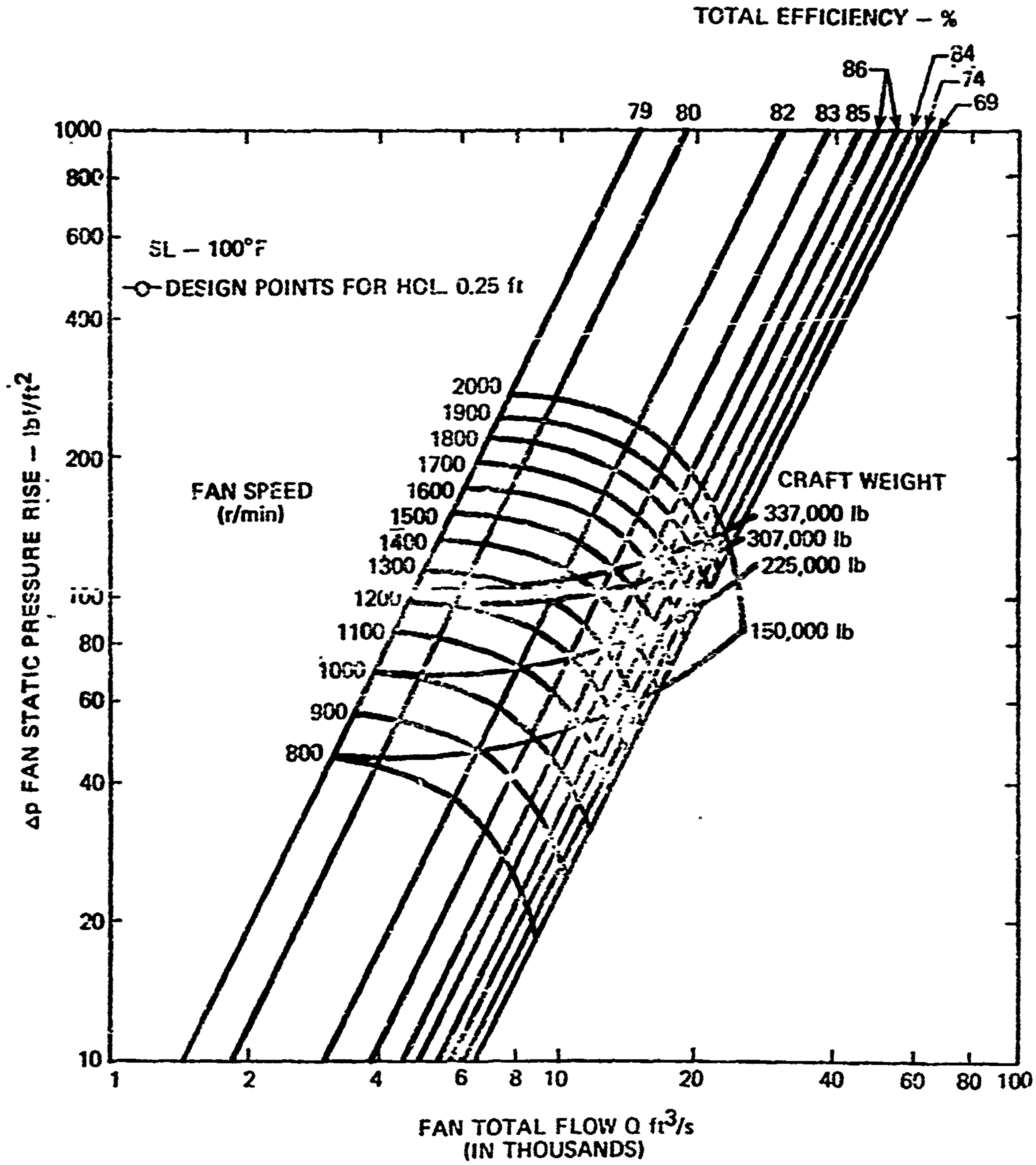


Figure 6. Fan Performance

Engine Modeling. Basis of Model. The turbine characteristics are based on the Bell Aerospace data shown in figure 6. A curve fit is used to find the optimum power turbine rpm, N_2, for gas generator rpm N_1. The gas generator rpm N_1 has been set by turbine control, as shown above. The shaft horsepower generated at optimum N_2 is found from a curve fit to figure 6 and the actual shaft horsepower from a parabolic fit to the off-optimum loss for the actual N_2. The torques required by fans and propellers are then computed at turbine rpm N_2 and the torque put out by the turbine is compared with that absorbed. Any excess accelerates the system.

Variable definitions.

N1(1) --- N1(6)	→ Gas generator rpm on each turbine. Turbines 1 to 3 are starboard.
N2OPT(1) --- N2OPT(6)	→ Optimum power turbine rpm N2 for each turbine N1.
SHPOPT(1) --- SHPOPT(6)	→ Horsepower generated by each turbine if operated at optimum N2.
SHP(1) --- SHP(6)	→ Actual horsepower generated by each turbine.
GPH(1) --- GPH(6)	→ Gallons per hour of fuel used by each turbine.
ETOR(1) --- ETOR(6)	→ Torque produced by each turbine in foot pounds.
TETOR(1) --- TETOR(2)	→ Total turbine torque, starboard and port.
N2(1) --- N2(2)	→ Power turbine rpm. N2(1) is starboard, N2(2) is port.
n, j	→ Dummy index variables. For engines n - 1, 2, 3 are starboard and correspond to j=1. n=4, 5, 6 are port and correspond to j=2.
NFAN(1) --- NFAN(2)	→ Starboard and port cushion fan rpm.
NPROP(1) --- NPROP(2)	→ Starboard and port propeller shaft rpm.
HPFAN(1) --- HPFAN(2)	→ Starboard and port power absorbed by cushion fans.
PFAN(1) --- PFAN(2)	→ Starboard and port fan pressures from cushion air dynamics.
QFAN(1) --- QFAN(2)	→ Starboard and port fan airflows from cushion air dynamics.

NETAF → Fan efficiency.

TNOZ(1) --- TNOZ(2) → Thrust from nozzles.

BETA(1) --- BETA(2) → β, propeller pitch angle in degrees.

XWIND → Relative head wind velocity in X_1 direction from Main program.

TPROP(1) --- TPROP(2) → Starboard and port propeller thrust.

HPPROP(1) --- HPPROP(2) → Horsepower absorbed by starboard and port propellers.

FTOR(1) --- FTOR(2) → Torque used by fans at N2 rpm, starboard and port.

PTOR(1) --- PTOR(2) → Torque used by propellers at N2 rpm, starboard and port.

DN2DT(1) --- DNDT(2) → Turbine power shaft accelerations in radians/sec^2.

ISYS → Equivalent moment of inertia of machinery as soon from a turbine power shaft.

Machinery Characteristics. The optimum turbine rpm N2 is determined for each turbine for its gas generator rpm N1 from: (Curve fit from figure 4).

$$N2OPT_n = 11672 - 1.90457\ N1_n + 1.21488 \times 10^{-4}\ N1_n^2. \qquad (22)$$

The shaft horsepower each turbine would generate at this optimum N2 is then determined:

$$SHPOPT_n = 14684. - 2.3976\ N1_n + 9.796 \times 10^{-5}\ N1_n^2. \qquad (23)$$

The actual horsepower produced by each turbine is then calculated:

$$SHP_n = SHPOPT_n \left\{0.1 + 1.8\ (N2j/N2OPT_n) - 0.9(N2j/N2OPT_n)^2\right\}. \qquad (24)$$

The fuel consumption rate is calculated for each turbine from the gas generator rpm

$$GPH_n = 757.6377 - .121784\ N1_n + 5.20379 \times 10^{-6} N1_n^2. \qquad (25)$$

The torque is then determined for each turbine:

$$ETOR_n = 3300\ SHPn/(2\pi N2J). \qquad (26)$$

The total starboard torque is:

$$TETOR(1) = \sum_{1-3} ETOR_n \qquad (27)$$

and port:

$$TETOR(2) = \sum_{4-6} ETOR_n.$$

The gear ratios in the shafting system are then used to find propeller and fan speeds:

$$\begin{aligned} NFANj &= .1297\ N2; & j &= 1,2 \\ NPROPj &= .6427\ NFANj & j &= 1,2. \end{aligned} \qquad (28)$$

The fan power absorbed is then calculated from the efficiency η,

$$HPFANj = PFANj\ QFANj/(540\ NETAF). \qquad (29)$$

From the fan and propeller powers the torque absorbed from the turbines derived:

$$\begin{aligned} FTORj &= 33000\ HPFANj\ /\ (2\pi N2j) \\ PTORj &= 33000\ HPROPj\ /\ (2\pi N2j). \end{aligned} \qquad (30)$$

The difference between the torque produced by the turbines (TETOR(1) and TETOR(2)) and that absorbed by fans and propellers leads to acceleration of the shafting:

$$DN2DTj = \left\{TETORj - FTORj - PTORj\right\} /\ ISYS. \qquad (31)$$

The shaft rpms are then updated:

$$N2j = N2j + DN2DTj\,\Delta t. \qquad (32)$$

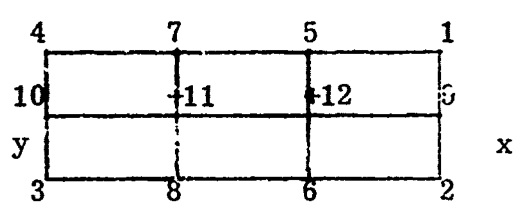
CUSHION SYSTEM MODEL

The determination of cushion pressure is governed by the fan pressure-flow relationship, the change of cushion volume with time due to waves or craft motion, and the pressure-flow relations beneath the skirts.

For speed in computation, internal functions are set up for computation of individual cushion volumes and skirt gap areas depending on the vehicles' present relation to the water or ground surface. These routines are described below. At each time step the volume of each air compartment is determined using the internal function HTAV and the height of the bottom plating above water, HOW, at the 35 points shown in figure 7. The depths of the skirts below this bottom plating is a function of cushion pressure and their determination is described below. The gap between skirts and water is calculated using the heights above water HOW and skirt depths SD at the peripheral points. These gaps are used with the internal functions QL and QS to determine skirt pressure-flow relationships. This volume and gap characteristic determination is described below.

With geometry determined, the pressure-flow relations are solved and the resulting forces on the vehicles are derived below.

Integration formulas. For the volume under each of the four cushion compartments an integration formula is established for the average height under the following configuration: (One of four compartments)



Assuming cubic curves in x and parabolic in y the internal function HTAV is established for the height for volume calculations:

$$\text{HTAV}(\text{AN1, AN2, AN3,---AN12}) = .0208333\ \{\text{AN1} + \text{AN2} + \text{AN3} + \text{AN4}\} + .0625\ \{\text{AN5} + \text{AN6} + \text{AN7} + \text{AN8}\} + .08333\ \{\text{AN9} + \text{AN10}\ \ + .25\ \ \text{AN11} + \text{AN12}\}. \quad (33)$$

The height of the hull bottom plating above the water or ground surface are calculated for each cushion and the result multiplied by the cushion planform area to obtain compartment air volume as a function of vehicle orientation and water or ground surface, $\eta(x_o, y_o, t)$.

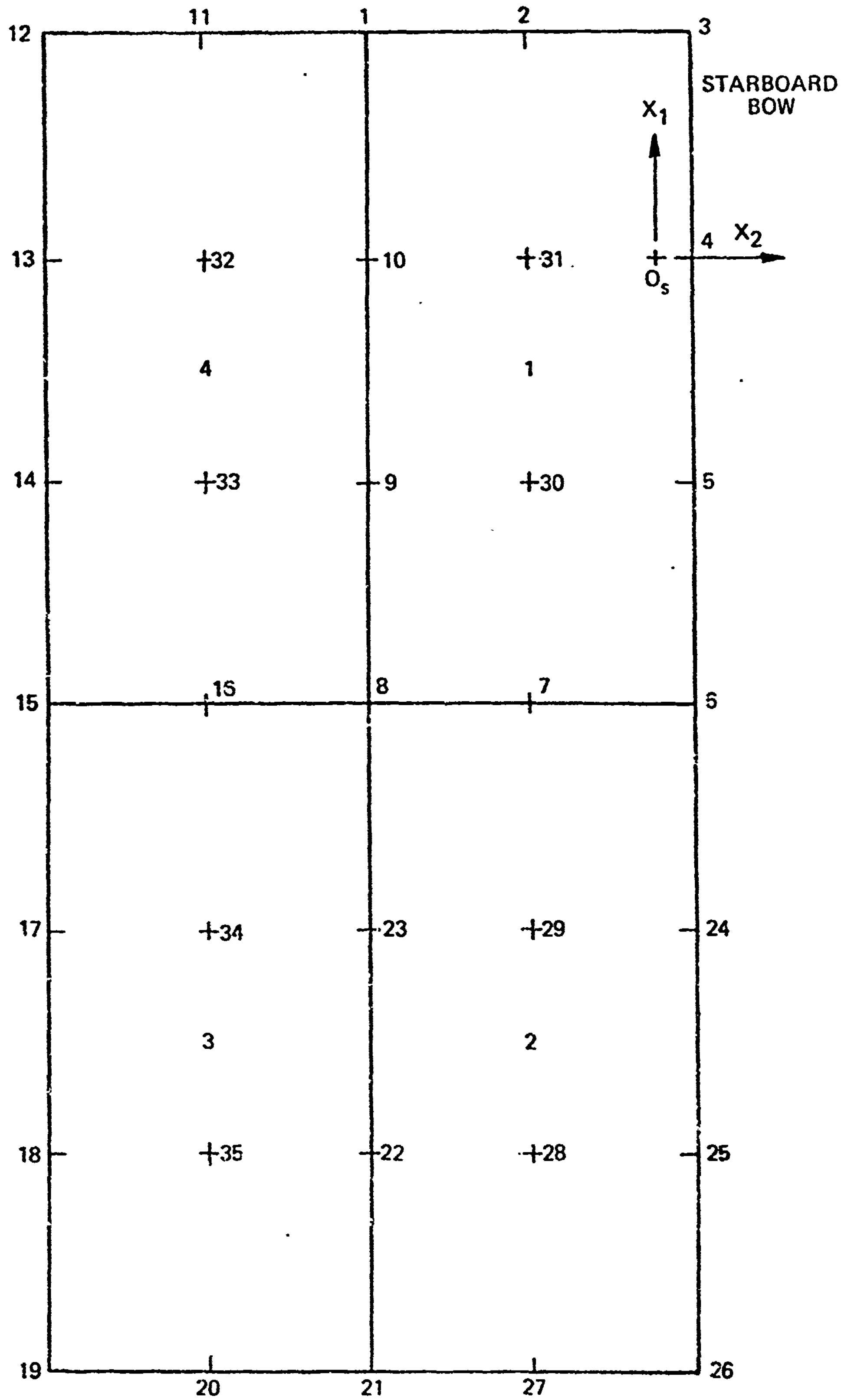
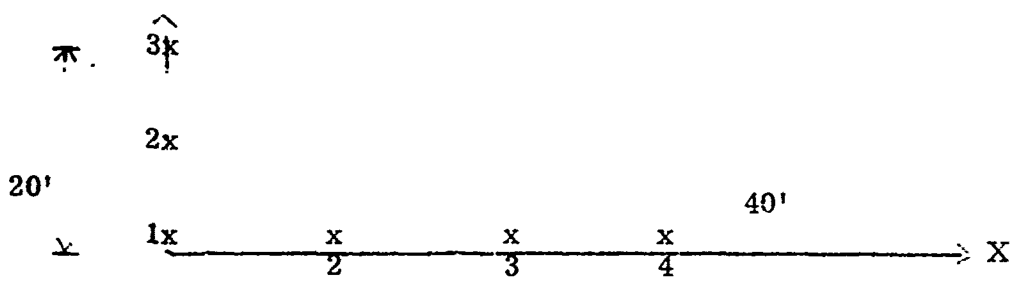


Figure 7. Location of Measurement Points

For calculation of pressure differences across the skirt gap a discharge coefficient of 0.6 is assumed:

$$\Delta p = \frac{Q^2}{A^2 \frac{2}{\rho} C_D} \qquad (34)$$

where Q is the flow rate under the skirts and A is the gap area under the skirt for this compartment.
Assuming skirt perimeter configurations for each compartment as follows:



functional routines are established for:

$$A\sqrt{2C_D/\rho}$$

In X a cubic gap is assumed: (35)

$$QL(ANS1...ANS4) = 94.16 \ \{ANS1 + 3xANS2 + 3xANS3 + ANS4\}.$$

In Y a parabolic gap is assumed: (36)

$$QS(ANS1, ANS2, ANS3) = 62.76 \ \{ANS1 + 4xANS2 + ANS3\}.$$

As the heights above water or ground of the appropriate points on the cushion shift periphery are input, the quantity

$$QL \text{ or } QS = A\sqrt{\frac{2C_D}{\rho}} \quad \text{results.}$$

Determination of Cushion Volumes and Skirt Gaps.

Cushion Volumes. The heights of each point on the bottom plating are determined measured from mean sea level:

$$HHULL_n = -X_3 + X_5 \ (HPXS1_n) - X_4(HPXS2_n) - HPXS3_n). \qquad (37)$$

$$HOW \quad = HULL_n - ETA_n . \qquad (38)$$

where

$$\left.\begin{array}{l} HPXS1_n \\ HPXS2_n \\ HPXS3_n \end{array}\right\}$$ Is the location of point n as measured from the pilot in es_1, $\bar{es}_2$, $\bar{es}_3$.

ETA_n - Is the height of water or ground at point n above mean sea level.

The volume of each cushion compartment ($VOLC_n$) is then determined with the heights above and the integration formula HTAV.

The change of cushion volume with time may then be calculated. This is usually called "wave pumping" and is the flow into the cushion required by its rate of change of volume.

$$QPUMP_j = (VOLC_j - VOLC_j\ (old))/\Delta t. \tag{39}$$

Skirt Gaps. An inner loop is set up for the calculation of the depth of the finger skirt periphery below the hard hull. The skirts are assumed to respond to pressure variations in the compartment according to the following logic:

$Y_n = 109. - PGC_n \rightarrow$ difference between equilibrium value and present compartment air pressure.

if $Y_n > 66.9$ $\quad Y_n = 66.9$ (40)
limits on skirt motion
if $Y_n < -66.9$ $\quad Y_n = -66.9.$

$$SDPRO_n = 4.5 + CSKRT \times Y_n - 8.325 \times 10^{-7}\ Y_n^3 \tag{41}$$

$\rightarrow$ cubic fit to steady skirt depth as a function of pressure.

$$SDDT_n = (SDPRO_n - SDOLD_n) \times TC \rightarrow \text{skirt depth} \tag{42}$$
approach to equilibrium.

$$SDDT_n = SDOLD_n + SDDT_n \times DELTAT \rightarrow \text{new skirt depth.} \tag{43}$$

The height of the skirt perimeter above (or below) water is calculated,

$$SAW_n = HOW_n - SD_n. \tag{44}$$

If SAW is negative, the skirt is touching water at point n and SAW_n will be used for skirt drag. Otherwise CLR_n, the airgap at this point is calculated:

$$CLR_n = SAW \tag{45}$$

If $\quad CLR_n \leq 0,\ CLR_n = 10^{-6}$

The clearance (CLR) is then used with functions QL and QS to determine the pressure-flow relation below the skirts.

Cushion Pressure Determination. The mathematical model developed for pressures and flow rates through the air cushion vehicle support system is based on assumptions of incompressible fluid flow and on discharge coefficients for flows through ducts and openings. In addition, inertial characteristics of the airmass are assumed to be small.

These assumptions greatly reduce the complexity of the mathematical model, but solution of the resulting state equations still entails the solution of six simultaneous nonlinear equations. These six equations are airflow continuity equations for the six "nodes" in the flow system as shown in figure 8. The sum of all the flows into each of the two "manifolds" downstream of the fans must equal zero. The flow rates into the manifolds depend on the fan rpm and the pressure difference between manifold and atmosphere. Likewise flow rates out are related to pressure differences between manifold and cushions.

Similar flow continity equations are written for each of the four cushions, where attention must be given to the rate of change of cushion volume due to vehicle motions and motion of the water surface. In figure 8 these effects are represented symbolically as a piston in each cushion.

For purposes of matrix manipulation in this section all pressures are stored in vector P:

P_1	Pressure in cushion 1 = PGC (1)
P_2	Pressure in cushion 2 = PGC (2)
P_3	Pressure in cushion 3 = PGC (3)
P_4	Pressure in cushion 4 = PGC (4)
P_5	Starboard fan pressure = PFAN (1)
P_6	Port fan pressure = PFAN (2).

Defining:

QINC(i)	flow into cushion i from manifold
QNOZ(i)	flow into manifold i through thrust nozzle
QFAN (i)	flow into manifold i through fan
QPUMP(i)	change of cushion volume i with time
QIC(i)	flow into cushion i from cushion (i-1)
Q(i)	flow into cushion i under its peripheral skirt.

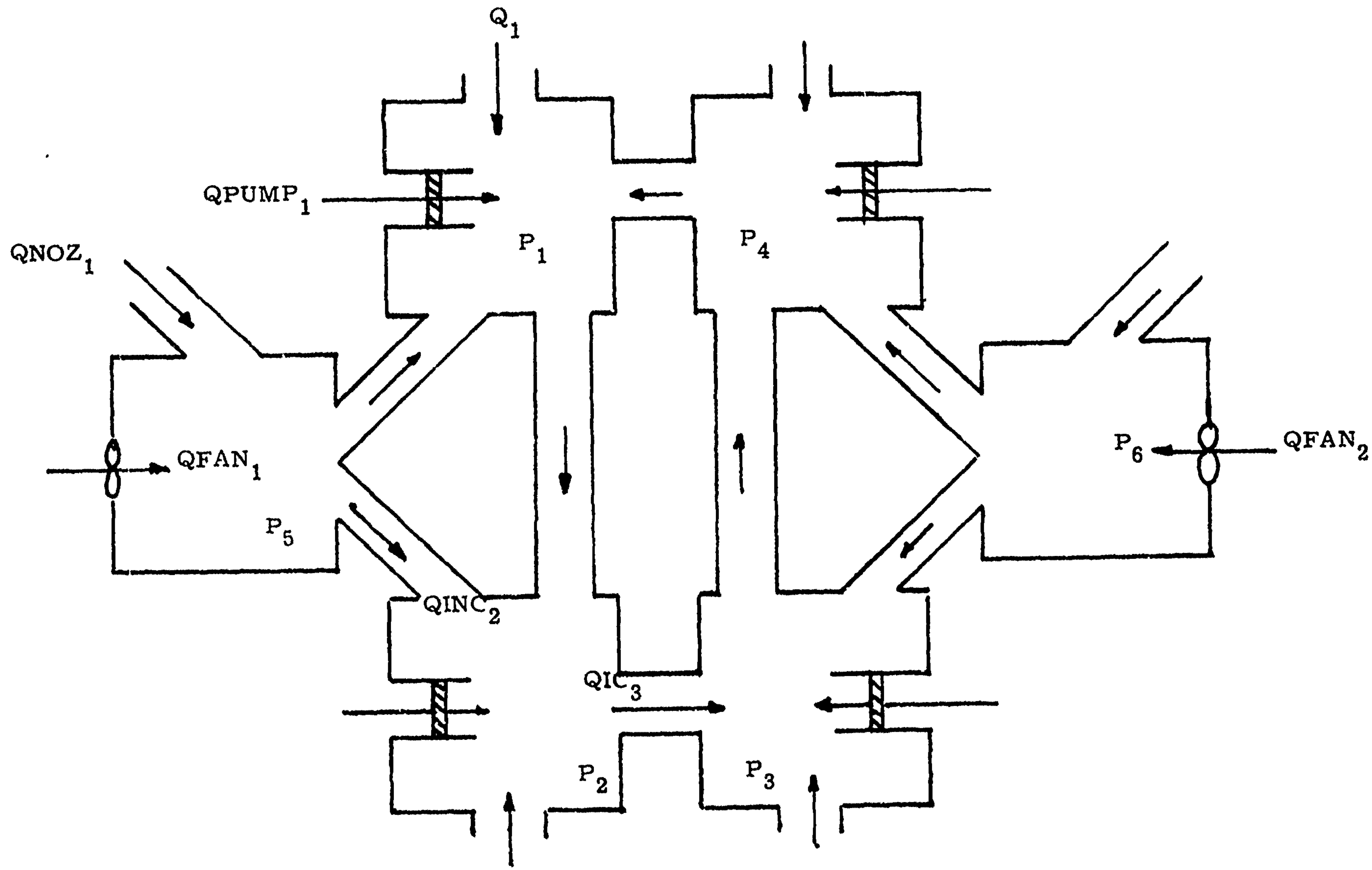


Figure 8. Cushion Pressure Schematic

The six continuity equations are:

For the manifolds:

$$\begin{aligned} -QINC(1) - QINC(2) + QNOZ(1) + QFAN(1) &= 0 \\ -QINC(3) - QINC(4) + QNOZ(2) + QFAN(2) &= 0. \end{aligned} \tag{46}$$

For cushions:

$$\begin{aligned} QINC(1) + QIC(1) - QIC(2) + QPUMP(1) + Q(1) &= 0 \\ QINC(2) + QIC(2) - QIC(3) + QPUMP(2) + Q(2) &= 0 \\ QINC(3) + QIC(3) - QIC(4) + QPUMP(3) + Q(3) &= 0 \\ QINC(4) + QIC(4) - QIC(1) + QPUMP(4) + Q(4) &= 0. \end{aligned} \tag{47}$$

Most of these flows may be expressed in terms of the pressures at the 6 nodes and the cross sectional area of the passages connecting them. The only exceptions are the fan flows which also depend on the fan rpms. The general form is:

$$Q \approx (\text{area} \sqrt{\frac{2}{\rho}} \sqrt{|\Delta p|} \ \text{sgn} (\Delta p). \tag{48}$$

The resulting expressions are:

$$\begin{aligned}
QINC(1) &= 589.\ \sqrt{|P(5) - P(1)|} \ \text{sgn}\ (P(5) - P(1)) \\
QINC(2) &= 589.\ \sqrt{|P(5) - P(2)|} \ \text{sgn}\ (P(5) - P(2)) \\
QINC(3) &= 589.\ \sqrt{|P(6) - P(3)|} \ \text{sgn}\ (P(6) - P(3)) \\
QINC(4) &= 589.\ \sqrt{|P(6) - P(4)|} \ \text{sgn}\ (P(6) - P(4)) \\
QNOZ(1) &= -346.\sqrt{|P(5)|} \ \text{sgn}\ (P(5)) \\
QNOZ(2) &= -346.\sqrt{|P(6)|} \ \text{sgn}\ (P(6)) \\
QIC(1) &= 675.\ \sqrt{|P(4) - P(1)|} \ \text{sgn}\ (P(4) - P(1)) \\
QIC(2) &= 338.\ \sqrt{|P(1) - P(2)|} \ \text{sgn}\ (P(1) - P(2)) \\
QIC(3) &= 675.\ \sqrt{|P(2) - P(3)|} \ \text{sgn}\ (P(2) - P(3)) \\
QIC(4) &= 338.\ \sqrt{|P(3) - P(4)|} \ \text{sgn}\ (P(3) - P(4))
\end{aligned}$$

$$Q(1) = -\text{area }(1)\cdot(14.5)\sqrt{|P(1)|}\ \text{sgn}\,(P(1))$$

$$Q(2) = -\text{area }(2)\cdot(14.5)\sqrt{|P(2)|}\ \text{sgn}\,(P(2))$$

$$Q(3) = -\text{area }(3)\cdot(14.5)\sqrt{|P(3)|}\ \text{sgn}\,(P(3))$$

$$Q(4) = -\text{area }(4)\cdot(14.5)\sqrt{|P(4)|}\ \text{sgn}\,(P(4)).$$

The fan characteristics are represented by the following expressions:

$$\text{QFAN}\,(1) \quad [-1280.\ \sqrt{|P(5) - 300.|}\ \text{sgn}\,(P5) - 300.) \qquad (49)$$

$$-31.6(P(5) - 300.)]\ \text{NFAN}(1)/2000.$$

$$\text{QFAN}(2) \quad [-1280.\ \sqrt{|P(6) - 300.|}\ \text{sgn}\,(P(6) - 300.)$$

$$-31.6(P(5) - 300.)]\ \text{NFAN}(2)/2000.$$

The four QPUMP values are determined by the motion of the craft and water surface and; thus, form an inhomogeneous part of each cushion equations.

Substitution of the above into equations gives six simultaneous nonlinear algebraic equations for P(i) given QPUMP, NFAN, and the escape area beneath the skirts. These equations are solved iteratively by a multi-dimensional Newton-Raphson procedure.

Symbolically the equations may be written:

$$[O_{NL}] \begin{Bmatrix} P_1 \\ \vdots \\ P_6 \end{Bmatrix} + \begin{Bmatrix} W_1 \\ \vdots \\ W_6 \end{Bmatrix} = 0 \qquad (50)$$

where $[O_{NL}]$ represents the six nonlinear operators on the vector P and W is a vector containing the inhomogeneous terms. If we introduce an error vector E, which should be zero when the equations are satisfied, we have:

$$\begin{Bmatrix} E_1 \\ \vdots \\ E_6 \end{Bmatrix} = [O_{NL}] \begin{Bmatrix} P_1 \\ \vdots \\ P_6 \end{Bmatrix} + \begin{Bmatrix} W_1 \\ \vdots \\ W_6 \end{Bmatrix}. \qquad (51)$$

With approximate values for the vector P, a linearization is possible giving:

$$\begin{Bmatrix} E_1 \\ \vdots \\ E_6 \end{Bmatrix} \qquad \begin{Bmatrix} E_1 \\ \vdots \\ E_6 \end{Bmatrix}_{\text{at } P = P_{guess}} + \left[\frac{dE_i}{dP_j}\right] \begin{Bmatrix} P_1 - P_{1guess} \\ \vdots \\ P_6 - P_{6guess} \end{Bmatrix}. \quad (52)$$

by setting E = 0 an updated approximation to P is obtained by solving the above set of coupled linear algebraic equations. This is iterated until suitable accuracy is obtained. In general one or two iterations are required.

The derivation of the error vector E and the matrix dE_i/dP_j is as follows:

$$E(1) = QPUMP(1) + 5.89x10^2 ASQRT(P(5) - P(1)) + 6.75x10^2 ASQRT(P(4) - P(1))$$
$$- 3.38x10^2 ASQRT(P(1) - P(2)) - 14.5\ AREA(1)\ ASQRT(P(1)) \quad (53)$$

$$dE(1)/dP(1) = -5.89x10^2\ DASQRT\ (P(5)-P(1)) - 6.75x10^2\ DASQRT\ (P(4)-P(1))$$
$$-3.38x10^2\ DASQRT\ (P(1)-P(2)) + AREA(1)\ DASQRT\ (P(1))$$

$$dE(1)/dP(2) = 3.38x10^2\ DASQRT\ (P(1)-P(2))$$

$$dE(1)/dP(4) = 6.75x10^2\ DASQRT\ (P(4)-P(1))$$

$$dE(1)/dP(5) = 5.89x10^2\ DASQRT\ (P(5)-P(1))$$

$$dE(1)/dP(1) = -dE(1)/dP(5) - dE(1)/dP(4) - dE(1)/dP(2)$$
$$-14.5\ AREA(1)\ DASQRT\ (P(1))$$

$$E(2) = QPUMP(2) + 5.89x10^2\ ASQRT\ (P(5)-P(2)) + 3.38x10^2\ ASQRT\ (P(1)$$
$$-P(2)) - 6.75x10^2\ ASQRT\ (P(2)-P(3)) - 14.5\ AREA(2) \cdot$$
$$ASQRT\ (P(2))$$

$$dE(2)/dP(1) = dE(1)/dP(2)$$

$$dE(2)/dP(3) = 6.75x10^2\ DASQRT\ (P(2)-P(3))$$

$$dE(2)/dP(5) = 5.89x10^2\ DASQRT\ (P(5)-P(2))$$

$$dE(2)/dP(2) = -dE(2)/dP(5) - dE(2)/dP(1) - dE(2)/dP(3)$$
$$-14.5\ AREA(2)\ DASQRT\ (P(2))$$

where:

ASQRT (X) $\sqrt{|X|}$ sign (X)

DASQRT (X) $\frac{1}{2\sqrt{|X|}} = \frac{d}{dx}\sqrt{X}$

$$E(3) = QPUMP(3) + 5.89x10^2\ ASQRT\ (P(6)-P(3)) + 6.75x10^2\ ASQRT\ (P(2) - P(3)) - 3.38x10^2\ ASQRT\ (P(3)-P(4)) - 14.5\ AREA\ (3)\ ASQRT\ (P(3))$$

$dE(3)/dP(2) = dE(2)/dP(3)$

$dE(3)/dP(4) = 3.38x10^2\ DASQRT\ (P(3)-P(4))$

$dE(3)/dP(6) = 5.89x10^2\ DASQRT\ (P(6)-P(3))$

$dE(3)/dP(3) = -dE(3)/dP(6) - dE(3)/dP(2) - dE(3)/dP(4) - 14.5\ AREA(3)\ DASQRT\ (P(3))$

$$E(4) = QPUMP(4) + 5.89x10^2\ ASQRT\ (P(6)-P(4)) + 3.38x10^2\ ASQRT\ (P(3) - P(4)) - 6.75x10^2\ ASQRT\ (P(4)-P(1)) - 14.5\ AREA(4)\ ASQRT\ (P(4))$$

$dE(4)/dP(1) = dE(1)/dP(4)$

$dE(4)/dP(3) = dE(3)/dP(4)$

$dE(4)/dP(6) = 5.89x10^2\ DASQRT\ (P(6)-P(4))$

$dE(4)/dP(4) = dE(4)/dP(1) - dE(4)/dP(3) - dE(4)/dP(6) - 14.5\ AREA(4)\ DASQRT\ (P(4))$

$$E(5) = -5.89x10^2\ ASQRT\ (P(5)-P(1) - 5.89x10^2 ASQRT\ (P(5)-P(2)) - 3.46x10^2 ASQRT\ (P(5)) - (NFAN(1)/2000.)^2 \left\{1.28x10^3 ASQRT\ (P(5)-300) + 3.16x10^1\ (P(5)-300)\right\}$$

$dE(5)/dP(1) = dE(1)/dP(5)$

$dE(5)/dP(2) = dE(2)/dP(5)$

$$dE(5)/dP(5) = -dE(5)/dP(1)-dE(5)/dP(2) - 3.46 \times 10^2 \text{ DASQRT } (P5)) -(NFAN(1)/2000)^2 \{1.28 \times 10^3 \text{ DASQRT } P(5) - 300)\}$$

$$\acute{E}(6) = -5.89 \times 10^2 \text{ ASQRT } (P(6) - (P(3)) - 5.89 \times 10^2 \text{ ASQRT } (P(6)-P(4) -3.46 \times 10^2 \text{ ASQRT } (P6)) - (NFAN(2)/2000.)^2 \{1.28 \times 10^3 \text{ ASQRT } (P(6) -300) + 3.16 \times 10^1 (P6) - 300)\} .$$

Forces and Moments due to wave system. Determination of the hydrodynamic forces acting on the craft is made by integrating the cushion pressure times the unit vector normal to the water surface below the craft. Since this surface tends to slope down from bow to stern, there will generally be a force opposing the craft's forward motion. Calculation of the overall shape of the water surface involves both wind generated ocean waves and shipwaves resulting from the craft's motion. These are treated in a later section. Under the assumption of uniform pressure within each chamber, this integral simplifies in evaluating the differences in water surface heights at opposite ends of chambers.

The hydrodynamic forces can then be evaluated as follows with ETA(i) referring to the ith planform point. ETA_i is the sum of the water amplitude at point i due to the seaway and the vehicle generated waves:

$$ETA_i = ETA_i \text{ (seaway)} + ETA_i \text{ (motion)} \qquad (54)$$

Wave Drag and Sway force, and Yaw Moment from Elevations.

In the Ahead Direction (X1):

$$WD1 = -PGC(1) * 20 * (ETA(1) + ETA(2) + ETA(3) - ETA(6) - ETA(7) - ETA(8))/3 \qquad (55)$$

$$WD1 = WD1 - PGC(2) * 20 * (ETA(6) + ETA(7) + ETA(8) - ETA(21) - ETA(27) - ETA(26))/3$$

$$WD1 = WD1 - PGC(3) * 20 * (ETA (8) + ETA(16) + ETA(15) - ETA(19) - ETA (20) - ETA(21))/3$$

$$WD1 = WD1 - PGC(4) * 20 * (ETA(1) + ETA (11) + ETA(12) - ETA(8) - ETA(16) - ETA (15))/3$$

In The Side Direction (X2):

$$WD2 = -PGC(1) * 40 * (ETA (3) + ETA(4) + ETA(5) + ETA(6) - ETA (1) -ETA(10) - ETA(9) - ETA(8))/4 \qquad (56)$$

$$WD2 = WD2 - PGC(2) * 40 * (ETA (6) + ETA(24) + ETA(25) + ETA(26) - ETA(8) - ETA(23) - ETA(22) - ETA(21))/4$$

WD2 = WD2 - PGC(3) * 40 *(ETA(21) + ETA (22) + ETA(23) +
ETA(8) - ETA(15) - ETA(17) - ETA (18) - ETA(19)/4

WD2 = WD2 - PGC(4) * 40 * (ETA(8) + ETA(9) + ETA(10) +
ETA(1) - ETA (14) - ETA(13) - ETA(14) - ETA(15))/4 .

The Moment About The Pilot's Location is: (57)

WM6 = PGC(1) * 40 * (ETA(6) + ETA(7) + ETA(8) - ETA(1) - ETA(2)
- ETA(3)) *8/3

WM6 = WM6 + PGC(2) * 40 * (ETA(26) + ETA(25) + ETA(24) + ETA(6) - ETA(21) -
ETA(22) - ETA(23) - ETA (8)) *50/4

WM6 = WM6 + PGC(3) * 40 * (ETA(19) + ETA(20) + ETA(21) - ETA(15) -
ETA(16) - ETA(8)) *28/3

WM6 = WM6 + PGC(3) * 40 * (ETA(21) + ETA(22) + ETA(23) + ETA(8) -
ETA (19) - ETA(18) - ETA(17) - ETA(15)) *50/4

WM6 = WM6 + PGC(4) * 20 * (ETA(15) + ETA(16) + ETA(8) - ETA(12) -
ETA(11) - ETA(1)) *28/3

WM6 = WM6 + PGC(4) * 40 * (ETA(8) + ETA(9) + ETA(10) + ETA(1) -
ETA (6) - ETA(5) - ETA(4) - ETA(3)) *40/4

WM6 = WM6 + PGC(1) * 40 * (ETA(6) + ETA(5) + ETA(4) + ETA(3)
- ETA(8) - ETA(9) - ETA(10) - ETA(1)) *10/4

WM6 = WM6 + PGC(2) * 20 * (ETA(21) + ETA(27) + ETA(26) - ETA(8) - ETA(7)
-ETA(6))* 8/3 .

Forces and Moments due to Cushion Pressure. Upward forces are exerted on the vehicle at the centroid of each cushion by the air pressure in the cushion. The force on each compartment is the cushion pressure (PGC) times the compartment area which is 800 square feet. Pitch and roll moments are also caused by these forces:

In the $\overline{es}_3$ direction, heave:

$$FX3CT = -\sum_{j=1}^{4} PGC(J) \times 800 \tag{58}$$

In the pitch direction, $\overline{es}_5$:

$$FX4CT = \sum_{j=1}^{4} PGC(J) \times 800 \times X_j$$

In the roll direction, $\overline{es}_4$:

$$FX4CT = \sum_{j=1}^{4} -PGC(J) \times 800Y_j$$

where X_j, Y_j is the location of the compartment center in plan relative to the ship based origin.

Forces and Moments due to Spray and Skirt Drag. The forces due to spray from the moving cushions and the cushion skirt drag are coupled in the experimental work on ACVs. We have no reliable basis to attempt a separation of these two effects analytically. Therefore, these effects have been approximated by losses proportional to horizontal velocity squared following experimental data in the Bell PDSR:

$$\overline{es}_1 \quad SDRX = -C_D X_1 \; |X_1| \tag{59}$$

$$\overline{es}_2 \quad SDRY = -C_D X_2 \; |X_2|$$

$$\overline{es}_4 \quad SDRR = -SDRY\ (H_P)$$

$$\overline{es}_5 \quad SDRP = SDRX\ (H_P)$$

$$\overline{es}_i \quad SDRYM = SDRY\ (XS1C) - SDRX\ (XS2C).$$

The drag coefficient C_D is assumed equal to 0.25 based on the Bell skirt drag and spray losses. H_P is the height of the origin above water.

ENVIRONMENT MODELING

The components of the vehicle environment modeled are the visual scene, wind, waves, and beach. The visual scene is created photographically from a camera mounted on the gantry. The wind direction and velocity are inputs to the program and are combined with the craft velocity vector to find apparent wind relative to the craft. Aerodynamic forces due to the apparent wind are calculated by curve fits to Bell experimental results. The wave surface is composed of the seaway, or waves that were present before the craft arrived, and vehicle generated waves.

WIND AND AERODYNAMIC FORCES. Apparent Wind. The wind velocity in feet per second (VWIND) and the wind direction in radians clockwise from North (AWIND) are inputs to the simulation. From these and the craft velocity vector the wind relative to the craft is calculated (apparent wind):

Head wind component: $$XWIND = \dot{X}_1 + VWIND \cos (X_6 - AWIND) \tag{60}$$

Side wind component (from starboard): $$YWIND = \dot{X}_2 - VWIND \sin (X_6 - AWIND) \tag{61}$$

Magnitude of apparent wind: $$VA = \sqrt{XWIND^2 + YWIND^2} \; . \tag{62}$$

The apparent wind direction, positive clockwise from straight ahead is then computed:

For YWIND + : $$\mathrm{Tan}^{-1}(YWIND/XWIND) \tag{63}$$

For YWIND − : $$-\pi + \mathrm{Tan}^{-1}(YWIND/XWIND) \; .$$

This angle in degrees: ABETA = BETAA $(180/\pi)$ has a range of -180 to +180.

Momentum Drag. The vehicle experiences reaction forces due to the ingest of air at relative velocity VA into the cushion fan system:

$$MDRAG = \rho (QFAN_1 + QFAN_2) \, VA. \tag{64}$$

This drag results in forces and moments about the pilot location:

$$\overline{es}_1: \quad XMDRAG = -MDRAG \cos (BETAA) \tag{65}$$

$$\overline{es}_2: \quad YMDRAG = -MDRAG \sin (BETAA) \tag{66}$$

$$\overline{es}_5: \quad PITCHM = XMDRAG \, (XS3C) \tag{67}$$

$\overline{es}_4$: ROLLMD = -YMDRAG (XS3C) (68)

$\overline{es}_6$: YAWMD = (12+XS1C)YMDRAG-XMDRAG(XS2C). (69)

<u>Windage.</u> The side force drag coefficient is curve fit from Bell experimental data shown in figure 9.

$$SDRAG = .01385\,|ABETA| - 7.69 \times 10^{-5}\,ABETA^2 \tag{70}$$

if, ABETA < 0, SDRAG = - SDRAG.

The frontal drag coefficient is curve fit from Bell experimental data shown in figure 10. This curve is fit with a parabola for angles less than 40°, with a cubic from 40° to 88°, and with a parabola above 88°. Port starboard symmetry is assumed:

if, $|ABETA| < 40^\circ$ (71)

$$FDRAG = -2.22 \times 10^{-4}\,ABETA^2 + 3.33 \times 10^{-3}\,|ABETA| + 0.5$$

if $40^\circ \leq |ABETA| < 88^\circ$

$$FDRAG = -2.48 \times 10^{-5}\,|ABETA|^3 + 5 \times 10^{-3}\,ABETA^2 - .324\,|ABETA| + 6.835$$

if, $|ABETA| > 88^\circ$

$$FDRAG = 1.04 \times 10^{-4}\,ABETA^2 - 3.518 \times 10^{-2}\,|ABETA| + 2.39.$$

The yaw moment coefficient about the boat center is based on the Bell experimental data shown in figure 11. This curve is fit with a straight line for $|ABETA| < 60^\circ$, with a cubic for $|ABETA|$ between 60° and 120°, and with a straight line from 120° to 180°. Again port-starboard symmetry is assumed.

if, $|ABETA| < 60^\circ$: $YDRAG = 1.67 \times 10^{-3}\,|ABETA|$ (72)

if, $60^\circ \leq ABETA \leq 120^\circ$: $YDRAG = 5.17 \times 10^{-7}\,|ABETA|^3 - 1.23 \times 10^{-4}\,ABETA^2 + 5.82 \times 10^{-3}\,|ABETA| + 8.27 \times 10^{-2}$

if, $ABETA > 120^\circ$: $YDRAG = 1.67 \times 10^{-3}$.

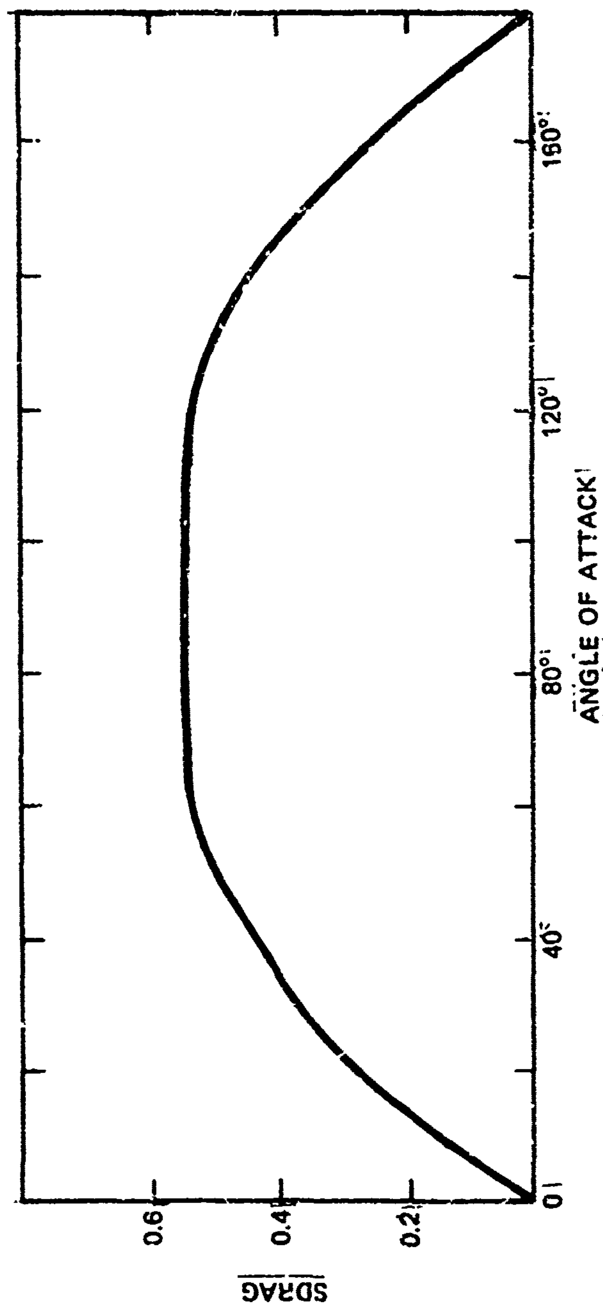


Figure 9. ACV Side Force Coefficient Bell Simulation (Figure 6.)

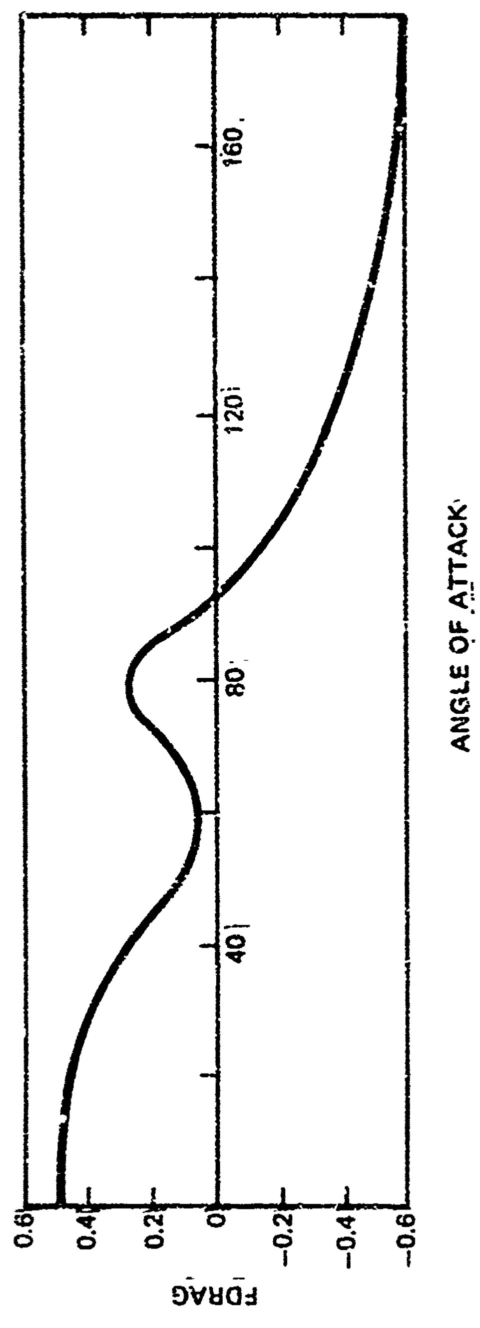


Figure 10. ACV Frontal Wind Drag Coefficient - Bell Simulation (figure 6)

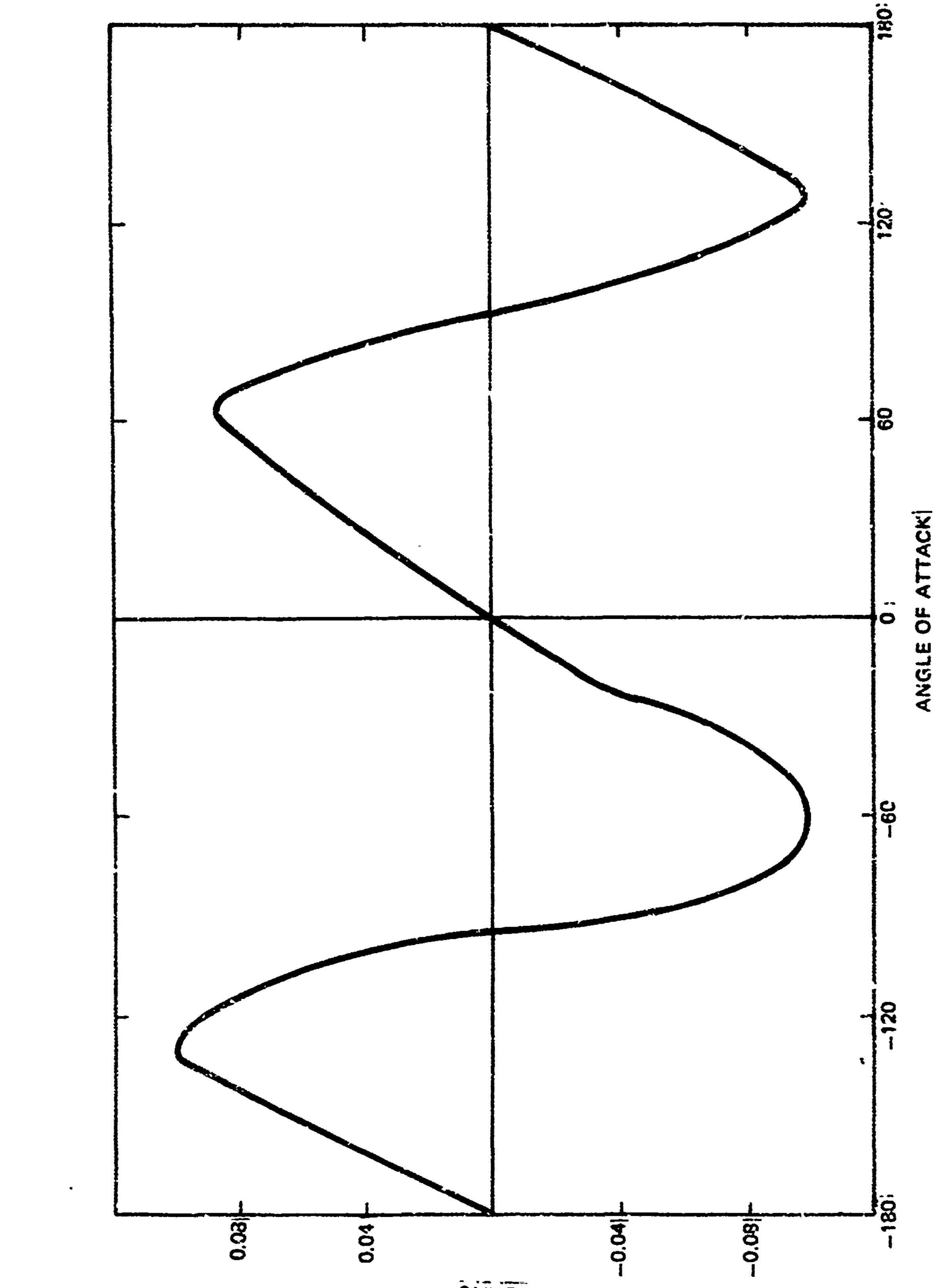


Figure 11. Yaw Moment - Bell Simulation (Figure 9)

The resulting windage forces and moments are:

$$\overline{es}_1: \quad XBDRAG = -FDRAG \times FCAREA \times 1/2\rho Va^2 \tag{73}$$

$$\overline{es}_2: \quad YBDRAG = -SDRAG \times SCAREA \times 1/2\rho Va^2 \tag{74}$$

$$\overline{es}_5: \quad PITCHA = XBDRAG \times XS2C \tag{75}$$

$$\overline{es}_4: \quad ROLLA = -YBDRAG \times XS3C \tag{76}$$

$$\overline{es}_6: \quad YAWBD = YDRAG \times SCAREA \times LCUSH \times 1/2\rho Va^2 + YBDRAG \times XS1C - XBDRAG \times XS2C. \tag{77}$$

Windage forces also occur on the propeller ducts. The angle of attack of the relative wind to the duct if first calculated including yaw velocity of the vehicle and propeller slipstream:

$$ADUCTj = \tan^{-1}\left\{\frac{Va \sin\beta aa + \Delta X \times XD(6)}{Va \cos\beta aa + 1.78\sqrt{TROPj}}\right\}. \tag{78}$$

Duct sideforce coefficient:

$$CDUCTj = -7.407 \times 10^{-3}\, ADUCTj + 1.333 \tag{79}$$

if $ADUCTj < 18,$ $\qquad CDUCTj = 6.667 \times 10^{-2}\, ADUCTj.$

Effective velocity at duct:

if $TPROPj \geq 0.0$

$$VDUCTj = \sqrt{\left\{Va \cos\beta aa + 1.78\sqrt{TPROPj}\right\}^2 + \left\{Va \sin\beta aa\right\}^2} \tag{80}$$

if $TPROPj < 0.0$

$$VDUCTj = \sqrt{\left\{Va \cos\beta aa - 0.89\sqrt{TPROPj}\right\}^2 + \left\{Va \sin\beta aa\right\}^2}.$$

Effective dynamic head at duct:

$$VDj = 1/2\rho VDUCTj^2. \tag{81}$$

The vehicle forces and moments due to the duct are then calculated:

$$YDUCT = -\sum_{j=1}^{2} CDUCTj \times DIA \times CHORD \times VDj \quad (82)$$

$$YAWD = YDUCT \times X1R$$

$$ROLLD = YDUCT \times X3R \quad (83)$$

The aerodynamic yaw damping moment is estimated by Bell:

$$YAWDC = -2.77 \times 10^6 \times XD(6). \quad (84)$$

SEAWAY DESCRIPTION. The composite small amplitude linear wave and finite amplitude solitary wave ocean surface model, as detailed in the June report, was subsequently found to be unworkable within a real time simulation environment. The essential difficulty derived from attempts to match in both time and space the surface elevations given by the two wave forms. A simpler model, judged to be adequate for engineering and simulation purposes, has been devised and is detailed below.

As a wave enters shallow water, roughly at a point where the water depth is one-half of the wave's length, the shoaling process begins. In general, the process reduces the wave's length, increases the height of the crest, and reduces the depth of the trough. Small amplitude wave theory can be used directly to predict the first effect within engineering accuracy and to partially account for the latter two. These affects are conveniently accounted for by using the Stokes second order correction for the wave profile.

Offshore of the point where shoaling begins, the first approximation to the ocean profile is

$$\overset{o}{\eta}_o = \frac{H_o}{2} \cos (K_o x - wt). \quad (85)$$

As the wave shoals, however, both the wave height and the wave number vary with distance, giving:

$$\overset{o}{\eta}_1 = \frac{H_o(x)}{2} \cos (\underline{\bar{K}}(x) - wt). \quad (86)$$

The wave height in shoaling water H(x) can be solved for in terms of the deep water wave height H_o, the local water depth h(x), and the local wave number K(x). It is (ref. Ippen, page 65):

$$H^o(x) = H_o \sqrt{\frac{2 \cos h^2 (K(x)\, h(x))}{2\, K(x)\, h(x) + \sin h(2K(x)\, h(x))}} \quad . \quad (87)$$

In equation 87, the local wave number K(x) is determined by solving (iteratively) the transcendental equation for the dispersion relation:

$$w^2 = g\,K(x) \tan h\,(K(x)\,h(x)\,). \tag{88}$$

The spatial variation as given by the $\overline{K}(x)$ term in the argument of the cosine in equation 86 requires special interpretation. In deep water the rate of change of wave phase in space is given by K_o; thus, the phase is given by

$$\int^x K_o dx' = K_o x \tag{89}$$

as shown in equation 85. For shoaling water, however, the analytic form for this rate of change is unknown and $\overline{K}(x)$, a function which must be computed for each beach-wave length combination, is defined as follows:

$$\overline{K}\,(x) = \int^x K(x')\,dx'. \tag{90}$$

As the water depth decreases, the slope of K(x) will increase giving a more rapid spatial variation of the cosine's argument in equation 86, equivalent to a shorter wave length.

Equation 85 and its inshore generalization, equation 86, gives a first approximation to the wave height; one in which the crests and troughs are displaced an equal distance from the mean surface elevation. A second order correction can be made by utilizing results from Stokes second order wave theory. This improvement is based on the first order solution. With crests occuring at both the crests and troughs of the first order wave, the effect is to increase the overall height of the crest and decrease that of the trough. This second order addition then may be written·

$$\overset{1}{\underset{1}{\eta}} = H^1(x) \cos\,[2\,(\overline{K}\,(x) - wt)\,]. \tag{91}$$

$$H^1(x) = \frac{\pi}{8}\;\frac{H_o(x)^2}{\frac{2\pi}{\overline{K(x)}}}\;\frac{\cos h(K(x)h(x)\,)(2+\cos h(2K(x)h(x)\,)}{\sin h^3(K(x)\,h(x)\,)}. \tag{92}$$

To prevent this second order contribution from erroneously dominating the first, $H^1(x)/H^o(x)$ is limited to a maximum value of 1/8. This condition prevents additional crests and is justified scientifically by recalling that this contribution is treated only as a small correction term.

As in the previous seaway description, the waves are assumed to form "spilling breakers" and disperse their energy gradually as they approach the shore. Critical wave height to water depth and wave height to wave length ratios are imposed as:

$$H^O(x) / h(x) \quad \leq 0.78 \tag{93}$$

$$H^O(x) / (2\pi / K(x)) \leq 0.142 . \tag{94}$$

Again, there is no wave remaining at the beach (by equation 93) and; therefore, there is no run up to account for. This departure from reality should not significantly alter the craft dynamics or training experience.

Seaway Calculations. Calculation of the wind-generated waves requires executing an initialization subroutine as well as a sequence of calculations within the time cycle loop in the simulation.

The main program reads a wave period, offshore wave height and a beach slope, and then calls the subroutine SEAWAY which does the following: Computes

WLEN	The offshore wave length.
WNKO	The offshore wave number.
DZ	The depth at which wave shoaling begins.
XSHOAL	The distance offshore at which shoaling beging (measured value of the X01 coordinate).

The remaining portion of the program determines values of the function K(x) as described by equation 90 in Seaway Description. This is done using a crude rectangular integrating routine with step sizes in the S direction equal to the deep water wave length. (As written only 100 values are permitted; this could be increased if necessary). During execution of SEAWAY, the subroutine WVNBR (for wavenumber) is frequently called. This routine solves the transcendental equation,

$$w^2 = gK(x) \tan h \, [K(x) \, h(x)]$$

relating the finite depth water wave frequency to its wave number K(x) and the water depth h(x). This is done using a Newton-Raphson root seeking technique for the equation (given w and h)

$$F(k) = w^2 - g K \tan h \, (Kh). \tag{95}$$

Once SEAWAY has been executed, no further calculations need be made prior to the beginning of the time loop.

Within the subroutine OCEAN, the wind wave heights are calculated at 35 pts and saved in the array ETA (I); I = 1,35.

Each of these points correspond to a point of water surface directly below a known point on the craft. Since the beach has been assumed to run east-west (X02 direction), the land to occupy X-1 ≥ 0, and the waves to approach the beach perpendicularly, the wave shape depends only upon the X01 coordinate and time. When given a point in space, the first step is to make certain that it is over water, not on the land.

If (HPX01(N)) 10, 50, 50

If on land, HPX01 ≥ 0 and ETA (N) is set to zero.

If at sea, a further check is made to see if the point is in the wave shoaling region or further offshore.

If offshore, the wave height and length are unaffected by the beach and ARGU; the argument of the cosing giving the shape of the wave is simply calculated.

If in the shoaling region, the wave height is adjusted according to the Seaway Description and the spatial contribution to the argument determined by linear interpolation within the table generated by subroutine SEAWAY.

Finally, the Stokes Second Order corrections are made on either the offshore or shoaling waves again according to SEAWAY description. This process is repeated for each of the 35 field points.

VEHICLE-GENERATED WAVES. The prediction of the surface wave pattern generated by the motion of the craft is a complex problem. The complexity lies in the fact that the craft's motion is not known beforehand, and the trajectory of motion in the horizontal plane could be very arbitrary. Difficult as the problem is, nevertheless, it is necessary to come up with a wave height production, because vehicle-generated waves not only provide the basic ingredient for calculating wave drag and moment, but also carry the effects of changing the cushion volume as well as the skirt clearance, factors which crucially affect the response of the craft.

The mathematical model for the prediction of vehicle-generated waves is based on linearized water-wave theory. The formulation and derivation of the solution of this hydrodynamics problem are detailed in Appendix A. The main result is that to obtain the wave height at a given point under the craft, a convolution integral in time must be evaluated. Input to the evaluation of the integrand and; thus, the integral consists of the recent history of the trajectory and orientation of the craft as well as the coordinates of the point where the wave height is desired. The integrand, or the kernel function of the integral, is a very complicated mathematical expression, evaluation of which during real time simulation is certainly beyond the present state of the art of computer technology. To make real time evaluation of the convolution integral possible, tabulation of the kernel function, which is a function of three variables, is necessary. With the table of kernel function loaded in memory, the computer can be programmed to interpolate values from the table and evaluate the integral on a real time basis.

Therefore, to implement the proposed mathematical model for the prediction of vehicle-generated waves, we can divide the required programming efforts into two stages. In Stage one, values of the kernel function are generated. This is an expensive and time-consuming process. The mathematical technique and associated programming procedures are thoroughly documented in Appendix B. These calculations, however, have to be done only once for a given water depth. A medium size table has been generated, for straight-ahead motion as well as a limited amount of sway and yaw. This table can be extended with little conceptual difficulty to accommodate the case of a general maneuver.

In Stage two, which occurs during the real time simulation process, the convolution integral which depends on the trajectory-history of the craft as input will be performed. The numerical procedure as well as the programming efforts involved in this stage are described in considerable detail in the main body of this report. Examples and test results are given to illustrate the application of the kernel table generated in Stage one. The presentation that is to follow consists of two parts: The first part describes the principles and mathematical details of the numerical procedure used in evaluating the convolution integral with specific reference to the origin of the equations. The second section consists simply of the FORTRAN IV version of the procedure described in the first section, put into effect. Results for several test runs of constant and variable speed conditions are given. Finally, to interface the vehicle-generated wave package with the rest of the motion simulation program, we must provide a routine that transfers and updates the array which stores the history of the craft's trajectory and orientation. This is described in Section III.

NUMERICAL PROCEDURE FOR REAL-TIME COMPUTATION OF VEHICLE-GENERATED WAVES. The nondimensional vehicle-generated wave-height, $\hat{\zeta} = \zeta(x', y')/(p_o/\rho g)$ can be obtained by performing the following convolution integral with respect to the nondimensional time, $\tilde{\tau} = t/\sqrt{a/g}$:

$$\hat{\zeta}(x', y') = \int_{-T_o}^{0} d\tilde{\tau}\, K^S(R_x(\tilde{\tau}), R_y(\tilde{\tau}), \tilde{\tau}) . \tag{96}$$

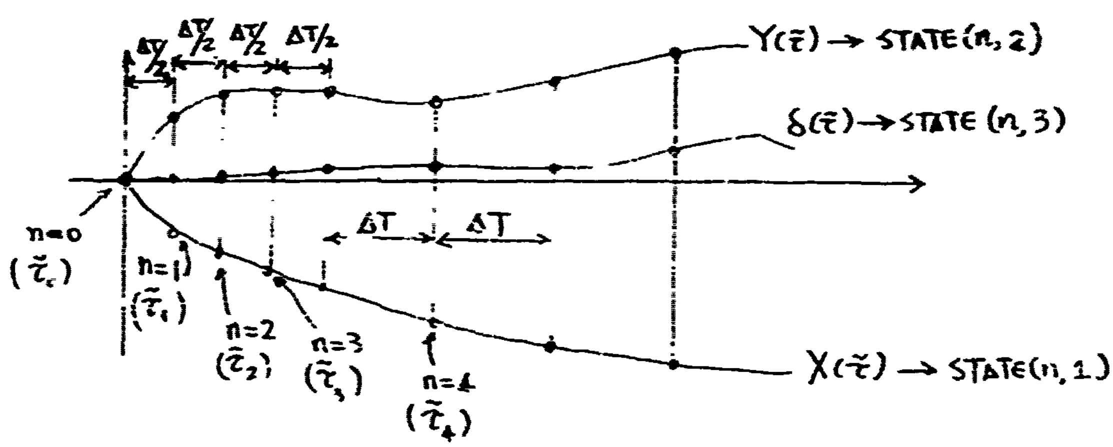
where (x', y') are the coordinates in the horizontal plane at which the wave height is desired, and K^S is the kernel function which will be tabulated and stored in a common block described later. To evaluate (96) we must also know the history of motion of the craft from $\tilde{\tau} = 0$ (i.e., the present) to $\tilde{\tau} = -T_o$, say some NT units of time interval, ΔT, back in the past. This is reflected through the functional dependence of $R_y(\tilde{\tau})$ and $R_x(\tilde{\tau})$ on $\tilde{\tau}$, as given by equation (7.4) of appendix A:

$$R_x(\tilde{\tau}) = [x' - X(\tilde{\tau})] \cos\delta(\tilde{\tau}) + [y' - Y(\tilde{\tau})] \sin\delta(\tilde{\tau}),$$

$$R_y(\tilde{\tau}) = [y' - Y(\tilde{\tau})] \cos\delta(\tilde{\tau}) - [x' - X(\tilde{\tau})] \sin\delta(\tilde{\tau}). \tag{97}$$

Here, $X(\tilde{\tau})$, $Y(\tilde{\tau})$, and $\delta(\tilde{\tau})$ are the history of the craft's trajectory and orientation, measured in the craft's present coordinate system, and must be known before we can calculate the $\tilde{\tau}$-integral. All length quantities shown have been nondimensionalized by the craft's half-length, a .

Let's assume we know $X(\tilde{\tau})$, $Y(\tilde{\tau})$, $\delta(\tilde{\tau})$ in a discrete manner as follows:



(98)

Here, we use the two-dimensional array STATE (n, m) to store the history of the trajectory. Note that if we integrate NT time intervals of ΔT each in the past, there are actually NT+3 ordinates because we use half-intervals in the recent past. Note also, by definition, STATE (o, n) for m=1, 2, 3.

Assume that we are given x', y' and the STATE's are defined, then the integrand $K^S(R_x(\tilde{\tau}), R_y(\tilde{\tau}), \tilde{\tau})$ at $\tilde{\tau}=\tilde{\tau}_n$, n=0, ... , NT+2, can be generated by calling a function subprogram KSINTP (RX, RY, N):

```
REAL KSINTP, F(50)
NN = NT + 2
DO 10  N=1, NN
Calculate RX, RY according to Equation (97)
using STATE (N,M), M=1,2,3 and x', y'.
AX = ABS (RX), AY = ABS (RY)
10 F(N) = KSINTP (AX, AY, N)
```

(99)

The integrand is now stored in the array F; the wave height, $\tilde{\zeta}$, can now be obtained simply by applying Simpson's rule with half-interval adjustments as below:

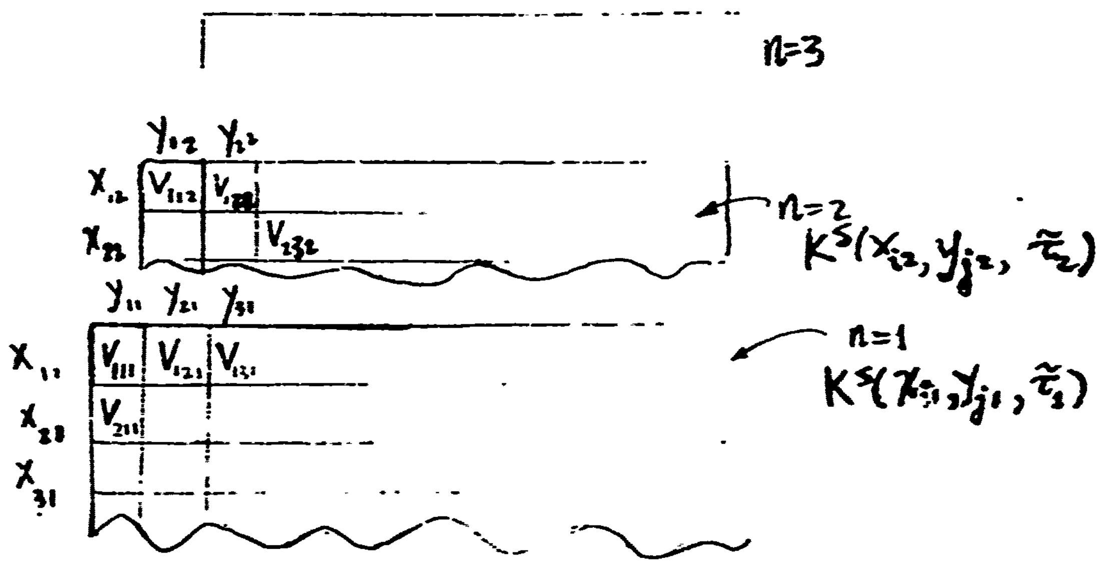
$$\hat{\zeta}(x',y') = \frac{\Delta T}{3}\Big\{ \frac{1}{2}F(0) + 2*(F(1)+F(3)) + F(2) + \frac{3}{2}F(4) + F(NN)$$
$$+ 4*[F(5)+F(7)+F(9)+\ldots+F(NN-1)] \qquad (100)$$
$$+ 2*[F(6)+F(8)+F(10)+\ldots+F(NN-2)]\Big\}$$

where F(0) = 0, and NN must be even.

We now proceed to describe the interpolation details of the function subprogram KSINTP, assuming that a table of the kernel function is available.

KSINTP (AX, AY, N) Routine. The Kernel function $K^S(x, y, \tilde{\tau})$ is a function of three variables. One may envision this as a three-dimensional array, with subscripts i, j, n, corresponding to grid points indices of the variables x, y, and $\tilde{\tau}$ respectively. The table is prepared in such a way that for given values of R_x, R_y, and $\tilde{\tau}$, only interpolations in the x and y directions are necessary. In other words, the $\tilde{\tau}$ grid of the K^S table is identical to the discrete time values defined in (98).

A schematic diagram showing the storage pattern of the values of the kernel function V_{ijn} is shown on the following page:



In order to completely define the table, the following information should be stored in a common block called/KRNL/ :

V_{ijn} = value of the kernel at the i, j grid point on the table corresponding to the time value $\tilde{\tau} = \tilde{\tau}_n$.

$\left.\begin{matrix} X_{in} \\ Y_{in} \end{matrix}\right\}$ = x and y grid point values of the $\tilde{\tau}_n$-table.

$(NX)_n$ = Number of x-grid points in the $\tilde{\tau}_n$-table.

$(NY)_n$ = Number of y-grid points in the $\tilde{\tau}_n$-table.

For example:

```
COMMON/KRNL/NX(50), NY(50), X(32, 50), V(32, 16, 50)
```

Each call of KSINTP (RX, RY, N) will therefore interpolate the value of the Kernel function from the x and y grid points in the $\tilde{\tau}_N$-table. The value is returned through the function's name. The following interpolation algorithm is based on a 3-point Lagrange formula.

Step 1: Search and find the indices p and q such that

$$X_{p,n} \le RX < X_{p+1,n} \quad \text{and} \quad Y_{q,n} \le RY < Y_{q+1,n}.$$

If $p = NX(N) - 1$, set $p = p-1$.

If $q = NY(N) - 1$, set $q = q-1$.

Step 2: Interpolate the value of kernel for 3 y-grid values at the x-value we want.

$$G_j = \Big[\prod_{i=p+1,\,p+2} \Big(\frac{RX-X_{in}}{X_{pn}-X_{in}}\Big)\Big] * V_{p,q+j,n} + \Big[\prod_{i=p,\,p+2} \Big(\frac{RX-X_{in}}{X_{p+1,n}-X_{in}}\Big)\Big] * V_{p+1,q+j,n}$$

$$+ \Big[\prod_{i=p+1,\,p} \Big(\frac{RX-X_{in}}{X_{p+2,n}-X_{in}}\Big)\Big] * V_{p+2,q+j,n}, \quad \text{for } j=o,1,2. \tag{101}$$

Step 3: Interpolate in the y-direction to obtain the Kernel value:

$$KSINTP = \sum_{k=0}^{2}\Big[\prod_{\substack{j=q \\ j\neq(q+k)}}^{q+2} \Big(\frac{RY-Y_{jn}}{Y_{q+k,n}-Y_{jn}}\Big) * G_k\Big] \tag{102}$$

Note that considerable simplifications in these formula will occur when the grid spacings are constant.

Subroutines for the Computation of Vehicle-Generated Waves. The "vehicle-generated wave package" which contains four subroutines is basically a FORTRAN IV version of the procedure described in the first section above. The name and purpose of each of the components inside the package are given below:

Routine LDTABL. This routine should be called once at the beginning of the calling program which uses the package. Its purpose is simply to load the common block/KRNL/ by reading the deck of cards which contains the values of the kernel function K^S.

Routine VGWAVE. This subroutine performs the convolution integral in time to obtain the wave height. Before entry, the STATE array must be defined for at least as many time steps as the kernel table is prepared.

In this routine, the procedure given by Equations (99) and (100) of the section above is performed by calling the interpolation routine KSINTP to generate the integrand and the routine SIMPSN to carry out the integration.

Routine KSINTP. A function subprogram which interpolates a value of the kernel function from the kernel table stored in the common block /KRNL/.

Routine SIMPSN. General purpose Simpson's rule integrator.

The above routines are given together with a sample main program which calls VGWAVE to obtain the wave height. In this particular application, the STATE array was generated by assuming that the craft is going at constant speed. Typical results are shown in figure 12 and figure 13 for a number of speeds. The Froude number corresponding to the case where the depth Froude number equals 1.0 is 0 · 707, above which waves diminish in height substantially. It has been observed that, in doing these computations, a nondimensional time step, ΔT, of 0 · 25 is too large for speed with Froude number greater than 5, although such step size appears to be more than sufficient for low speed cases. The kernel table used in these calculations was generated up to $T_o = 6.0$ (see Equation (96)) back in the past. This is insufficient for speed lower than $F_r = 0 \cdot 5$, while for high speed, a value of $T_o = 2.0$ is sufficient. For an accuracy of approximately 5%, a useful criterion for the truncation time T_o of the convolution integral (Equation (96)) is to keep integrating until $[R_x^2(T_o) + R_y^2(T_o)] \simeq 25.0$. Physically, this implies that effects caused by the craft more than 2.5 craft-lengths ago are neglected. Such capability can be easily incorporated in the final simulation program.

One interesting test computation performed as a check of the results was the case of zero forward speed. The results for integrating up to $T_o=6.0$ are shown in figure 14. This resembles and approaches the shape of the static wave profile, which is entirely consistent with the fact that if pressure impulses are applied continuously over the same spot on the free surface, the transients will eventually die out and one is then left with just the static profile.

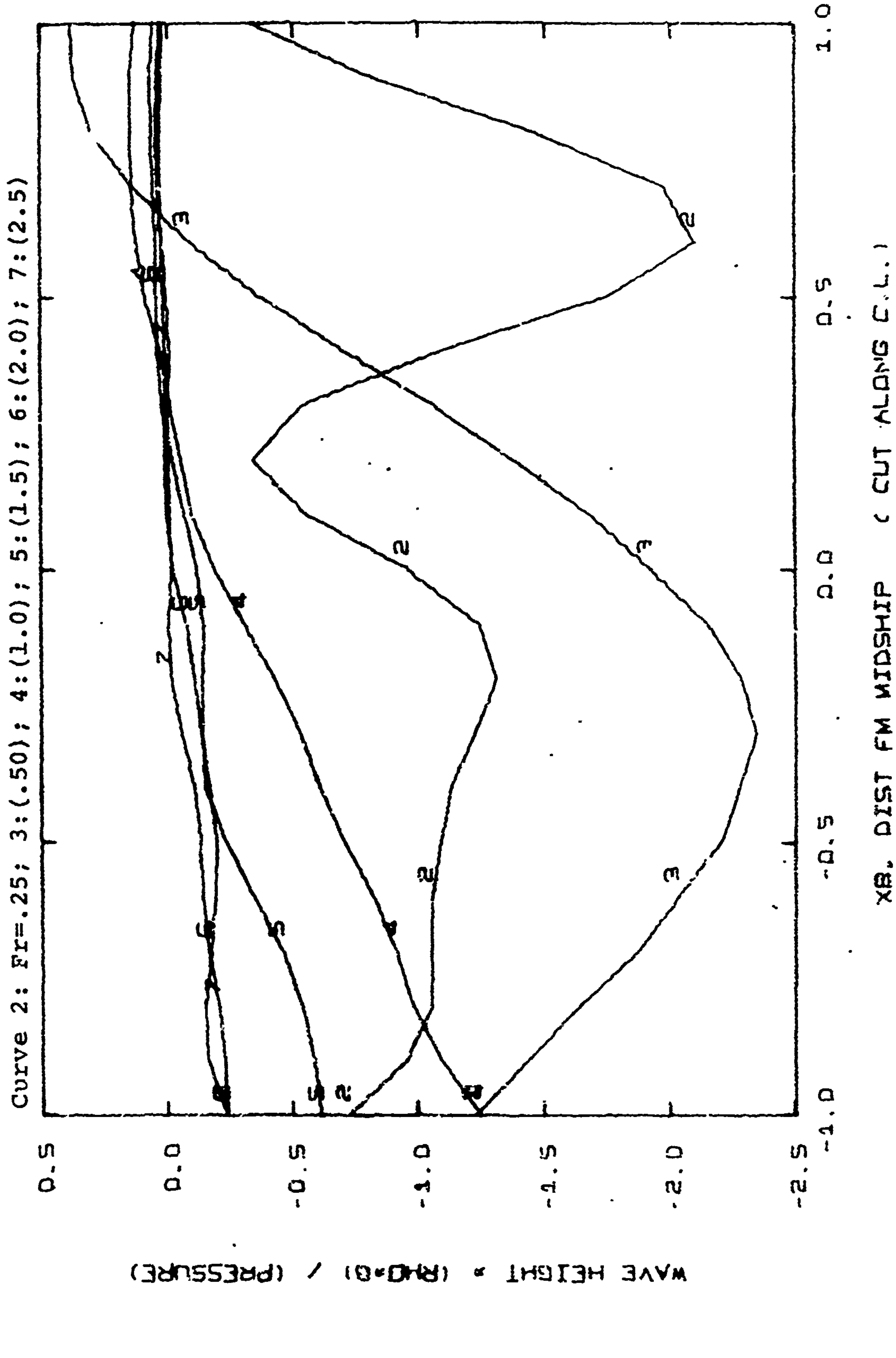


Figure 12. Craft at Constant Speed

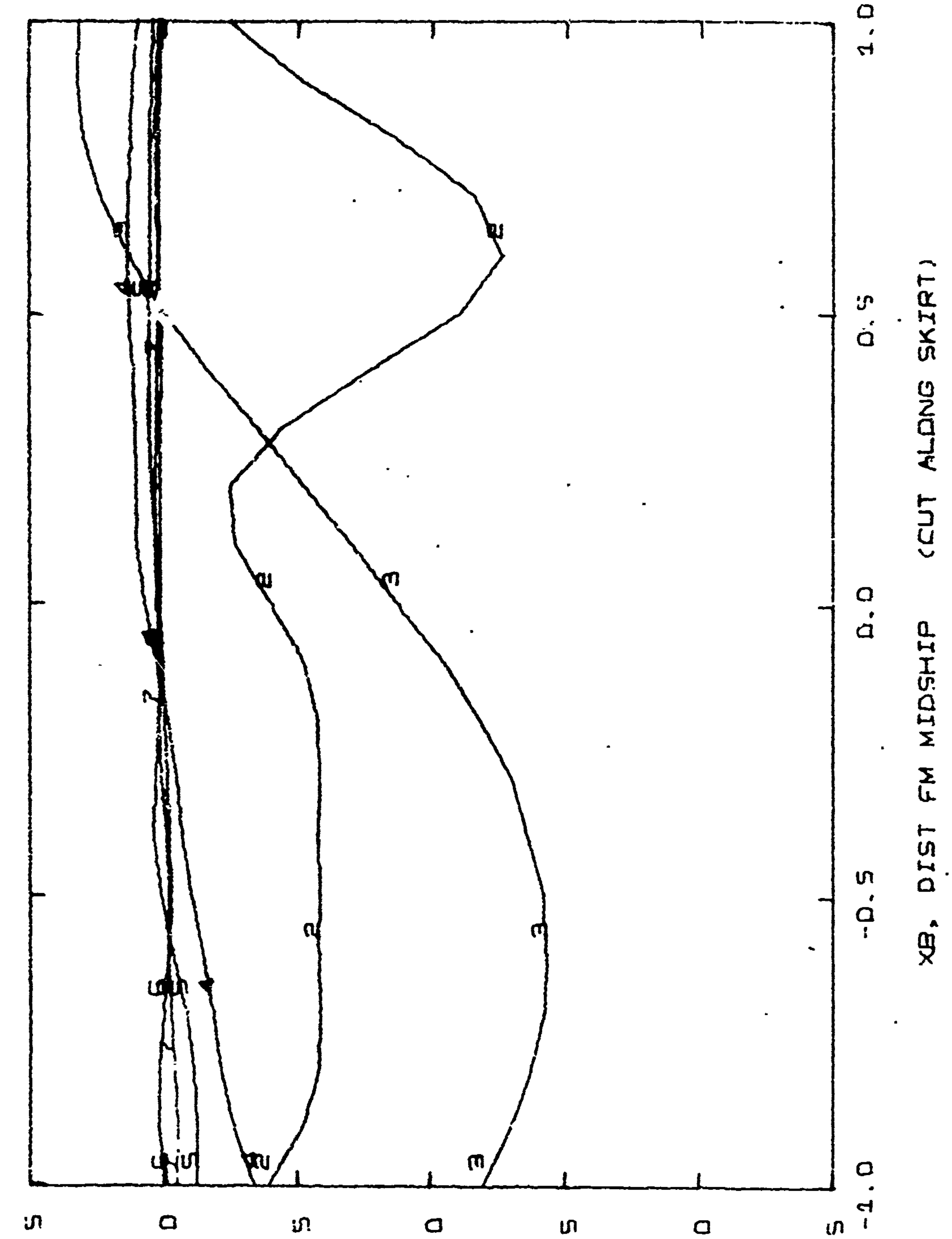


Figure 13. Craft at Constant Speed

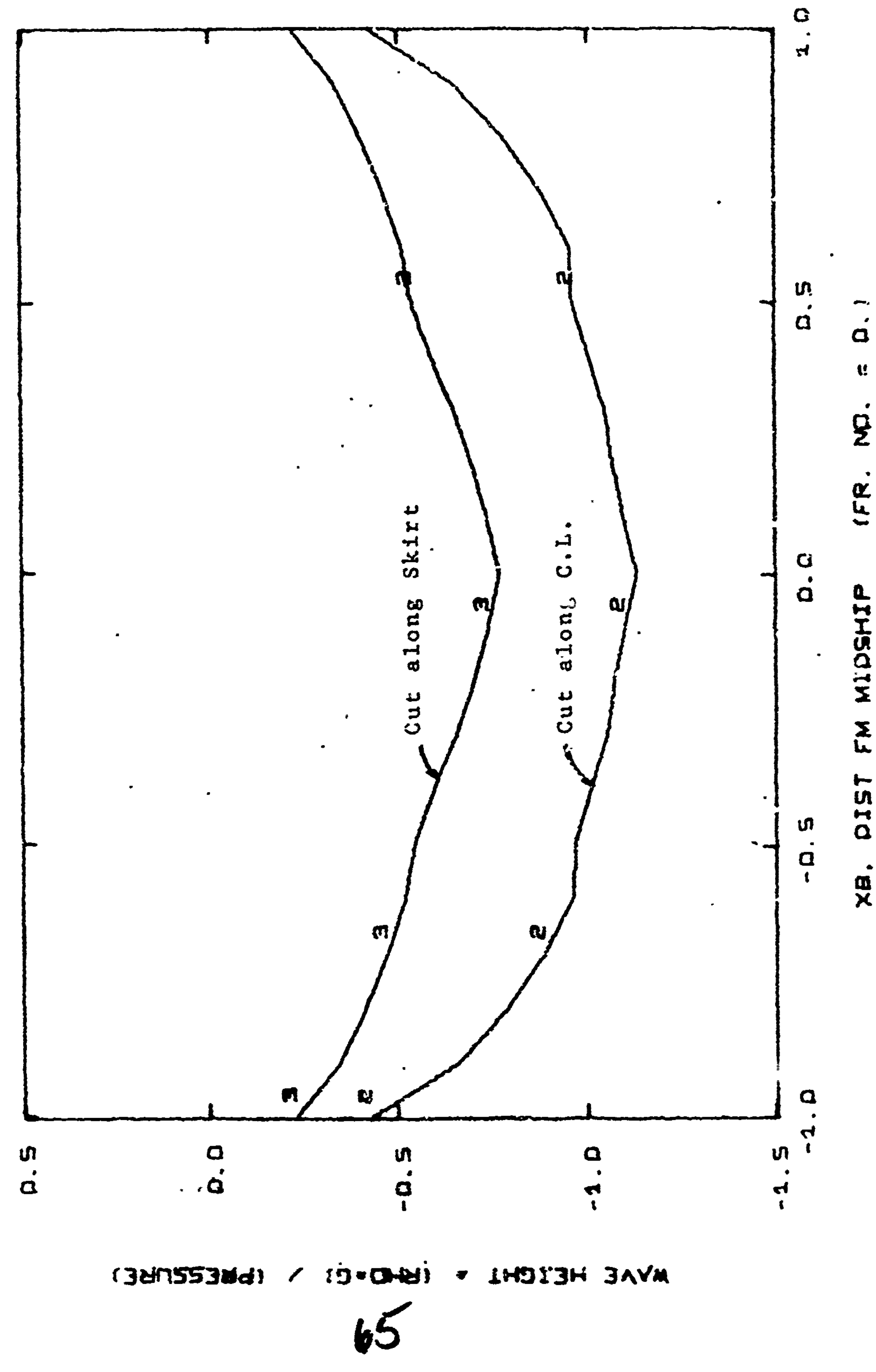


Figure 14.: Craft at Zero Forward Speed

```
C      SAMPLE PROGRAM FOR USING VEHICLE-GENERATED WAVES PACKAGE.
C     ***********                                              RWY 8/4/74.
C
C    PACKAGE CONTENTS ..  ROUTINES *VGWAVE*, *LDTABL*, *KINSTP*, *SIMPSN*,
C                         AND DECK OF CARDS CONTAINING KERNEL VALUES.
C
C    THE FOLLOWING COMMON - STATEMENTS SHOULD BE INCLUDED IN CALLING
C    PROGRAM.
         COMMON/RNPARM/ALPP,ST,DFLT,NTS,ERR, HT
         COMMON/STATUS/AT,BT,TIME,STATF(50,3)
         COMMON /KRNL/ NX(30), NY(30), X(40,30), Y(20,30), V(40,20,30),
       1   NTAU, T(30)
C
         DIMENSION FLDPT(50,2), HEIGHT(50), ETANON(50)
C
C    DEFINE A NUMBER OF CONSTANTS.
C    THESE CONSTANTS ARE NOT ACTUALLY USED, BUT SERVE TO DOCUMENT THE
C    GENERAL ENVIRONMENT IN WHICH THE KERNEL TABLE WAS PREPARED.
C    ALPP=DECAY FACTOR IN PRESSURE DISTRIBUTION.
C    ST = BEAM/LENGTH RATIO
C    HT = DEPTH OF WATER / HALF-LENGTH RATIO.
C    ERR = ERROR BOUND IN GENERATING TABLE.
         ALPP= .314159
         ST = .5
         ERR = .01
         HT = 1.
C
C    DEFINE PHYSICAL CONSTANTS OF WATER AND CRAFT.
C    PRESZ = NOMINAL PRESSURE OF CUSHION, IN LBS/(SQ. FT)
         RHO = 1.98
         GRAVTY = 32.17
         PRESZ = 109.375
         CONVRT =  PRESZ/(RHO*GRAVTY)
C
C
C    CALL *LDTABL* ONCE TO LOAD CARD DECK CONTAINING TABLE .
C    SEE ROUTINE *LDTABL* FOR DETAIL DEFINITION OF VARIABLES IN THE
C    COMMON /KRNL/.
C    THE TIME GRID WILL BE BE PRINTED OUT AUTOMATICALLY.
C    IPRINT SHOULD BE SET EQUAL TO 1 IF ONE WANTS TO PRINT OUT THE
C    VALUE OF THE ENTIRE TABLE,  OTHERWISE, SET IPRINT=0.
         IPRINT = 0
         CALL LDTABL (IPRINT)
         DELT = T(NTAU) -T(NTAU-1)
C
C
C    GENERATE SOME TYPICAL FIELD POINTS DATA.
C    NFP = NO. OF FIELD POINTS.
C    ARRAY *FLDPT* CONTAINS THE X AND Y COORDINATES OF THE POINT WHERE
C    THE WAVE HEIGHT IS DESIRED.
         NFP = 50
         DX = .1
         DY = .5
         DO 100  J=1,2
         DO 100  K=1, 25
         KK = (J-1)*25 + K
         FLDPT(KK,1) = 1. -(K-1)*DX
         FLDPT(KK,2) = (J-1)*DY
     100 CONTINUE
         WRITE(6,2001)  ((FLDPT(K,J), J=1,2), K, K=1, NFP)
    2001 FORMAT(//'  FIELD POINTS,  X AND Y '/ (2X, 2F10.4,I5))
C
C
         DO 400  M=1, 10
C
C    GENERATE SOME TRAJECTORY DATA AND STORE IN ARRAY *STATE*
C    IN THE FOLLOWING TEST-DATA GENERATION, CONSTANT SPEED IS ASSUMED.
C    NTS = NO. OF TIME STEPS FOR WHICH STATE DATA ARE AVAILABLE  , AND
C    MUST BE .GE. TO NTAU
         READ (5,3001)  FRNO
         IF ( FRNO .LT. .01 )  STOP
    3001 FORMAT ( F10.4)
```

```
      WRITE (6, 3002)  FRNO
 3002 FORMAT (     '1 FROUDE NO. =', F5.3 )
      SQ2 = SQRT(2.0)
      NTS = NTAU
      DO 200  N=1, NTS
      STATE(N,1) = -SQ2*FRNO*T(N)
      STATE(N,2) = 0.
  200 STATE(N,3) = 0.
      WRITE(6,2002) ((STATE(I,J),J=1,3), I=1,NTS)
 2002 FORMAT ( '0 STATES W.R.T. PRESENT COORDINATE-SYSTEM',/
     1     '   XOR          YOR          THETA ' /, (1X,3F10.5 ))
C
C
C   ***
C    WITH FLDPT AND STATE DATA DEFINED, WE ARE READY TO CALL *VGWAVE*
C   TO OBTAIN WAVE-HEIGHTS.
C
C     SAMPLE LOOP FOR USING ROUTINE *VGWAVE*, WHICH RETURNS A NON-
C   DIMENSIONALIZED WAVE-HEIGHT VALUE.
C   MULTIPLYING BY THE VARIABLE *CONVRT* WIIL CONVERT RESULT TO
C   PHYSICAL UNITS.
C
C   *****
      IPRINT = 0
      DO 300  I=1, NFP
C
C   IPRINT, IF SET EQUAL TO 1, WILL CAUSE ALL INTERPOLATED KERNEL-VALUES
C   TO BE PRINTED OUT WHEN *VGWAVE* IS CALLED. IF THAT IS NOT DESIRED,
C   SET IPRINT=0.
      ETANON(I) =-VGWAVE( FLDPT(I,1), FLDPT(I,2), IPRINT)
      HEIGHT(I) = CONVRT*ETANON(I)
C
  300 WRITE (6,3004)  I, FLDPT(I,1), FLDPT(I,2), HEIGHT(I), ETANON(I)
C   *****
C
 3004 FORMAT (    '0 FIELD POINT NO.',I3,' (X,Y)=(', 2F8.4,')',3X,
     1          '  WAVE HEIGHT =', F8.4,2X, 'LENGTH UNITS',
     2 ', (NON-DIMENSIONALIZED=', F10.6, ')' )
C
C  STATEMENT BELOW SERVES MERELY TO PUNCH RESULTS ON CARDS.
      WRITE (7, 3005)  FRNO, (I, FLDPT(I,1), FLDPT(I,2), ETANON(I),
     1  I=1, NFP)
 3005 FORMAT ( 'FP. NO. =', F8.4, / (I5, 5X, 2F10.4, F10.6))
  400 CONTINUE
C
      STOP
      END
```

```
      FUNCTION VGWAVE( XFP, YFP, IPRINT)
      COMMON/RNPARM/ALPP,ST,DELT,NTS,ERR, HT
      COMMON/STATUS/AT,BT,TIME,STATE(50,3)
      COMMON /KRNL/ NX(30), NY(30), X(40,30), Y(20,30), V(40,20,30),
     1  NTAU, T(30)
      REAL KSINTP
      DIMENSION F(50)
C
C  THIS SUBROUTINE PERFORMS THE CONVOLUTION INTEGRAL IN TIME TO
C  OBTAIN THE WAVE-HEIGHT AT THE FIELD POINT WITH COORDINATES (XFP, YFP)
C  BEFORE ENTRY, THE ARRAY *STATE* DESCRIBING THE TRAJECTORY, MUST BE
C  DEFINED FOR AT LEAST *NTAU* TIME STEPS.
C  THE COMMON BLOCK /KRNL/ IS ALSO ASSUMED TO HAVE BEEN LOADED ALREADY.
C
      IF ( IPRINT .EQ. 1 ) WRITE (6, 1002)
 1002 FORMAT(//3X, 'J', 8X, 'RX(TAU)', 4X, 'RY(TAU)',6X, 'TAU', 4X,
     1 ' KERNEL' )
C
      F(1) = 0.
      DO 30  N=1, NTAU
C
C  COMPUTE THE QUANTITIES, RX(TAU), AND RY(TAU).
C
      DT = STATE(N,3)
      CD = COS(DT)
      SD = SIN(DT)
      DX = XFP - STATE(N,1)
      DY = YFP - STATE(N,2)
      AT = CD*DX + SD*DY
      BT = CD*DY - SD*DX
      RX = ABS(AT)
      RY = ABS(BT)
C
C  GENERATE KERNEL VALUES BY CALLING KSINTP.
C
      JJ = N+1
      F(JJ) = KSINTP(RX, RY, N)
       IF ( IPRINT .NE. 1 ) GO TO 30
      WRITE (6, 1003)  N, AT, BT, T(N), F(JJ)
 1003 FORMAT ( 1X, I4, 2X,2F11.3, 2X, F8.4 , 2X, F10.6 )
   30 CONTINUE
C
C  CALL INTEGRAION ROUTINE
      CALL SIMPSN (F, DELT, NTAU+1, ANS)
      VGWAVE = ANS
      RETURN
      END
```

```
      REAL FUNCTION KSINTP(RX, RY, N)
C     INTERPOLATION ROUTINE FOR KERNEL VALUE.
      COMMON /KRNL/ NX(30), NY(30), X(40,30), Y(20,30), V(40,20,30),
     1  NTAU, T(30)
      INTEGER P, Q
C
C     STEP1.. SEARCH FOR THE INDICES P AND Q.
C
      II = NX(N)
      DO 10  I=3, II
      P=I
      IF ( X(P,N) .GE. RX)  GO TO 15
   10 CONTINUE
      GO TO 50
   15 P=P-2
      JJ = NY(N)
      DO 20  J=3, JJ
      Q=J
      IF ( Y(Q,N) .GE. RY)  GO TO 25
   20 CONTINUE
      GO TO 50
   25 Q=Q-2
C
C     STEP2, INTERPOLATE VALUES OF 3 Y-GRID VALUES AT THE VALUE OF X GIVEN.
C
      RP = RX-X(P,N)
      RP1= RX-X(P+1,N)
      RP2 = RX-X(P+2,N)
      PP1 = X(P,N)-X(P+1,N)
      PP2 = X(P,N)-X(P+2,N)
      P12 = X(P+1,N)-X(P+2,N)
      F1= RP1*RP2/(PP1*PP2)
      F2= -RP*RP2/(PP1*P12)
      F3 = RP*RP1/(PP2*P12)
C
      G0 = F1*V(P,Q   ,N) + F2*V(P+1,Q   ,N) + F3*V(P+2,Q   ,N)
      G1 = F1*V(P,Q+1,N) + F2*V(P+1,Q+1,N) + F3*V(P+2,Q+1,N)
      G2 = F1*V(P,Q+2,N)+ F2*V(P+1,Q+2,N) + F3*V(P+2,Q+2,N)
C
C     STEP3 , CROSS INTERPOLATE IN THE Y-DIRECTION.
C
      RQ = RY-Y(Q,N)
      RQ1 = RY-Y(Q+1,N)
      RQ2 = RY - Y(Q+2,N)
      QQ1 = Y(Q,N) - Y(Q+1,N)
      QQ2 = Y(Q,N) -Y(Q+2,N)
      Q12 = Y(Q+1,N)-Y(Q+2,N)
      KSINTP= RQ1*RQ2/(QQ1*QQ2)*G0 - RQ*RQ2/(QQ1*Q12)*G1 +
     1 RQ*RQ1/(QQ2*Q12)*G2
      RETURN
   50 KSINTP=0.
      RETURN
      END
```

```
      SUBROUTINE LDTABL(IPRINT)
C  ROUTINE FOR LOADING  KERNEL TABLES.
     COMMON /KRNL/ NX(30), NY(30), X(40,30), Y(20,30), V(40,20,30),
    1  NTAU, T(30)
C  T(N) = TIME VALUE CORRESPONDS TO THE N-TH TANLE.
C  NTAU = NO. OF TIME TABLES.
C  X = X-GRID POINT VALUES.  Y = Y-GRID POINT VALUES.
C  NX(N) = NO. OF X-GRID POINTS IN TABLE N.
C  NY(N) = NO. OF Y/GRID POINTS IN TABLE N.
C  V = KERNEL VALUE.
C
 1001 FORMAT ( I5)
 1002 FORMAT (25X, F10.5)
 1003 FORMAT ( 32X, I5 /(8F10.5))
 1004 FORMAT (8X, 9F8.4)
 1012 FORMAT (//'  KERNEL TABLE FOR TIME =', F10.4 )
 1013 FORMAT ( '  RX GRID, NO. OF GRID POINTS =', I5, /, 32X, I5/
    1 (8F10.5 ))
 1023 FORMAT ( '  RY GRID, NO. OF GRID POINTS =', I5, /, 32X, I5/
    1 (8F10.5 ))
C
      READ (5,1001) NTAU
      DO 10  N=1, NTAU
      READ (5, 1002) T(N)
   11 READ (5, 1003) II, (X(I,N), I=1, II)
   12 NX(N) = II
      READ (5,  1003)  JJ, (Y(J,N), J=1, JJ)
   13 NY(N) = JJ
      IF ( IPRINT .NE. 1 ) GO TO 15
C
      WRITE(6, 1012) T(N)
      WRITE(6, 1013) II, (X(I,N), I=1, II)
      WRITE (6, 1023)  JJ, (Y(J,N), J=1, JJ)
C
   15 DO 20  J=1, JJ
      READ (5, 1004) ( V(I,J,N), I=1, II)
      IF ( IPRINT .NE. 1 ) GO TO 20
      WRITE(6, 1004) ( V(I,J,N), I=1, II)
   20 CONTINUE
   10 CONTINUE
      WRITE (6, 1005)  NTAU, ( T(N), N=1, NTAU)
 1005 FORMAT ( 2X, I5, ' TIME GRID VALUES AS DEFINED BELOW..',
    1 /, (8F10.4 ))
      RETURN
      END
```

```
      SUBROUTINE SIMPSN (Y, H, N, R)
      REAL     Y(1)    , SE,SO, R
      M=(N-3)/2
      SO=0.
      SE= Y(2)
      IF (N .EQ. 3) GO TO 15
      DO 10  I=1, M
      SO = SO + Y(2*I+1)
   10 SE = SE + Y(2*I+2)
   15 R= H*( Y(1) +Y(N) + 4.*SE + 2.*SO )/3.
      RETURN
      END
```

The Interface Between Vehicle Kinematics and the Computation of Vehicle-Generated Waves. Determination of the vehicle-generated wave system requires a knowledge of the past path and orientation of the vehicle. This section outlines two mathematical manipulations required to provide the vehicle-generated wave package the necessary information regarding the vehicle's trajectory: (1) conversion to the coordinate system used in the vehicle-generated wave subroutines and updating the records with most recent motions; and (2) providing this information in dimensionless form and at time increments as needed by the vehicle-generated wave subroutines.

The first operation is accomplished by the subroutine UPDATE whose argument list contains newly determined incremental changes of the vehicle's position and orientation in the horizontal plane, as measured in the coordinate system fixed at the pilot's location. This routine updates the dimensional records of the vehicle motion which is maintained in the coordinate system used in the analysis of vehicle generated waves. The pilot based system has X1 forward, X2 starboard and X6 (yaw angle) measured positive from the X1 axis toward the X2 axis; the system used in the determination of vehicle-generated waves has X forward, Y port and θ yaw angle measured positive from the X axis toward the Y axis (figure 15a). Because the coordinate system in which the motion records are maintained is continuously changing (fixed to moving craft) the entire record needs to be updated at each time step. This can be seen clearly in (figure 15b).

During a time interval between time t and time $(t+\Delta t)$ the integrated equations of motion will give an incremental change of X1, X2, and X6, say DX1, DX2 and DX6. From these subroutines UPDATE first determines corresponding changes DX, DY, DD according to:

$$DX = DX1 + 30.0 - 34.9857 \cos (DX6 + 0.54042026)$$

$$DY = -DX2 - 18.0 + 34.9857 \sin (DX6 + 0.54042026)$$

$$DD = -DX6$$

The equations reflect geometrical relations best explained by reference to figure 15a. The next operation refigures the motion history in the new coordinate system located DX ahead, DY to port and rotated DD to port from the old system. This record is kept in the array DSTATE (100,3) where the columns, left to right, contain the x position, y position and yaw angle of the vehicle at a number of time intervals into the past corresponding to the row number. (Note that row "0" isn't needed since the three values there are by definition zero.) During each time step, the coordinates of a previous point on the trajectory makes one step down the column. The oldest entry is always lost. Using expressions for the shift and rotation of a coordinate system's origin:

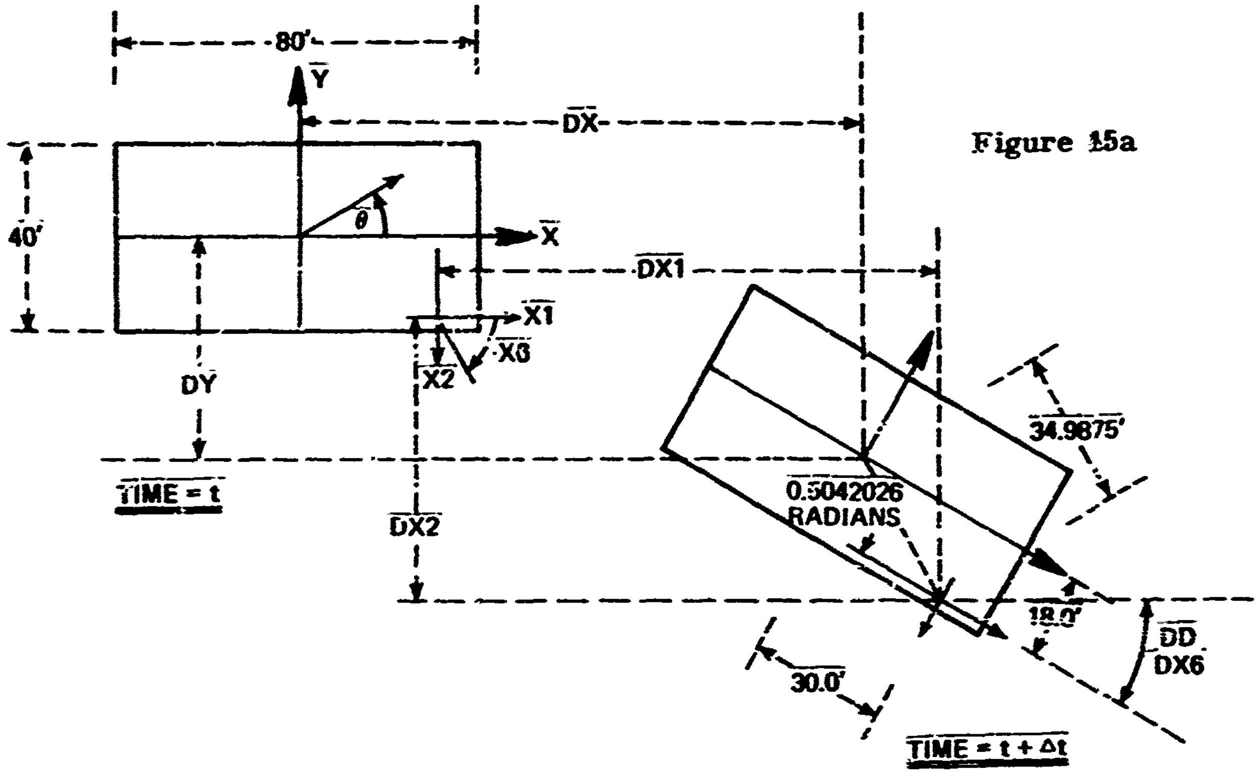


Figure 15a

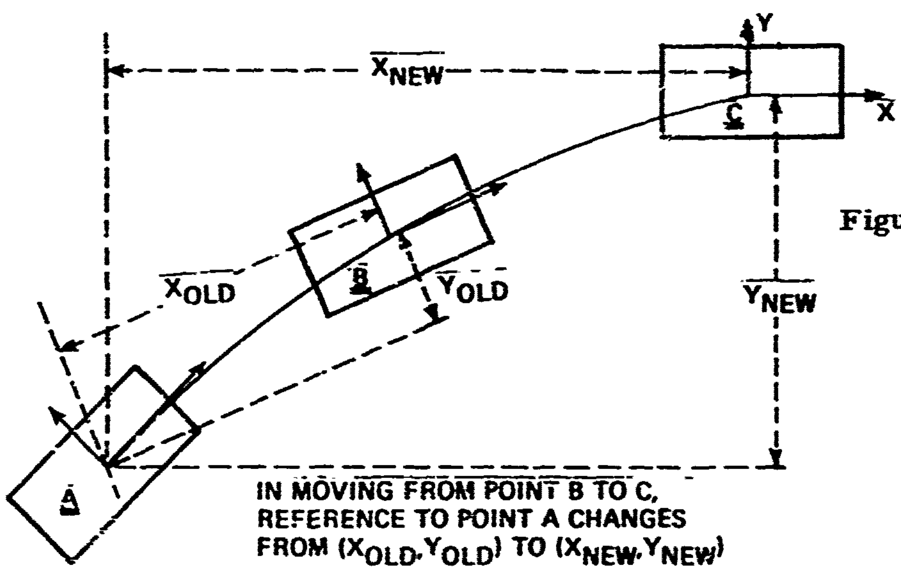


Figure 15b

Figure 15. Update of Vehicle Motion

$$X_{new} = (X_{old} - DX) \cos (DD) + (Y_{old} - DY) \sin (DD)$$

$$Y_{new} = -(X_{old} - DX) \sin (DD) + (Y_{old} - DY) \cos (DD)$$

$$\theta_{new} = \theta_{old} + DD$$

where

$$X_{new} = \text{DSTATE } (i + 1, 1), \; X_{old} = \text{DSTATE } (i, 1),$$

$$Y_{new} = \text{DSTATE } (i + 2, 2), \; Y_{old} = \text{DSTATE } (i, 2),$$

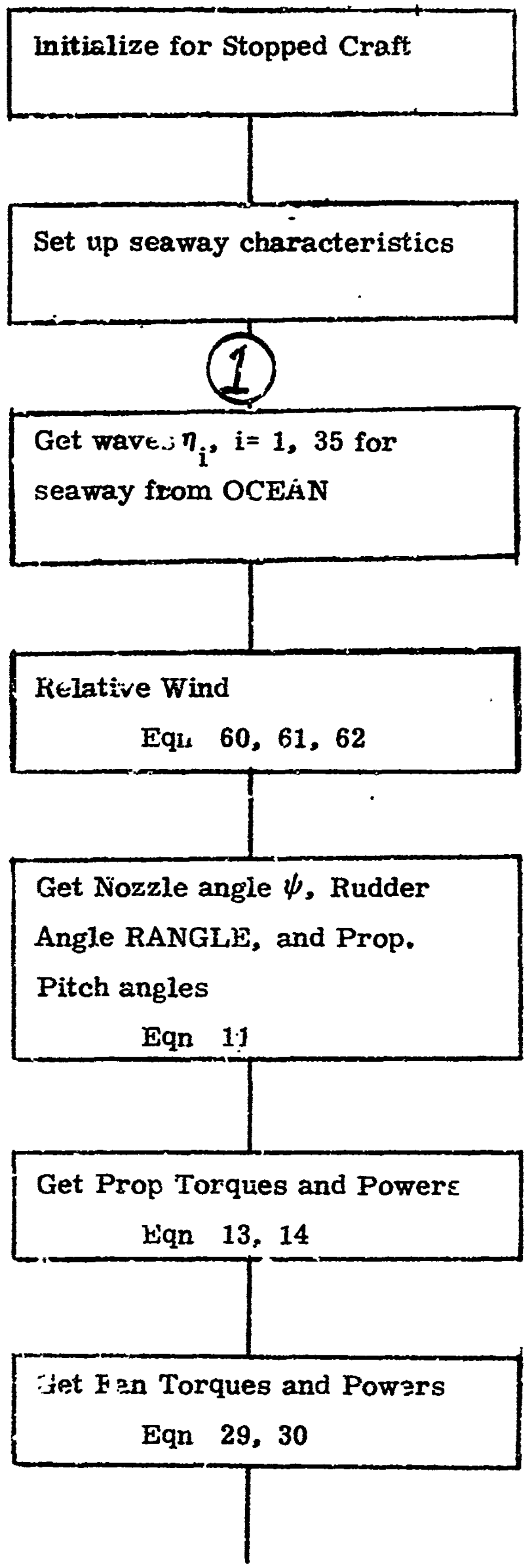
$$\theta_{new} = \text{DSTATE } (i + 2, 3), \; \theta_{old} = \text{DSTATE } (i, 3).$$

The array DSTATE is updated during each time step of the main simulation and entries are maintained in feet and radians.

The second operation is performed by the subroutine INTERP. This routine uses array DSTATE to produce array STATE, which differs in two ways. First, the entries are made in "natural" length units "a" (vehicle half length) and in radians. Second, the time increment is changed to correspond to the dimensionless time used in the vehicle-generated wave package and the time increment between all entries is not the same. Between entries in rows one through four the time increment is 1/8 of a "natural" time unit $\sqrt{a/g}$ (a = vehicle half length; g = gravitational acceleration). Between rows four and five (and thereon) the time increment is of a time unit.

SECTION IV

LOGIC FLOW FOR ACV MATHEMATICAL MODEL



Use engine characteristics for engine torque output
Eqn 22 to 27

↓

Update engine RPMS
Eqn 31, 32

↓

Calculate cushion volumes
Eqn 37, 38, 33

↓

Calculate wave pumping $QPUMP_i$, i = 1, 4
Eqn 39

↓

Calculate skirt depths
Eqn 40, 41, 43, 43

↓

Calculate skirt heights above water; CLR_i, i = 1, 35
Eqn 44

↓

Calculate flow escape areas
Eqn 35, 36

↓

Calculate cushion pressures and flows
Eqn 48 to 52

Calculate vehicle generated wave heights & sum with seaway
Eqn 54

Wave forces
Eqn 55, 56, 57

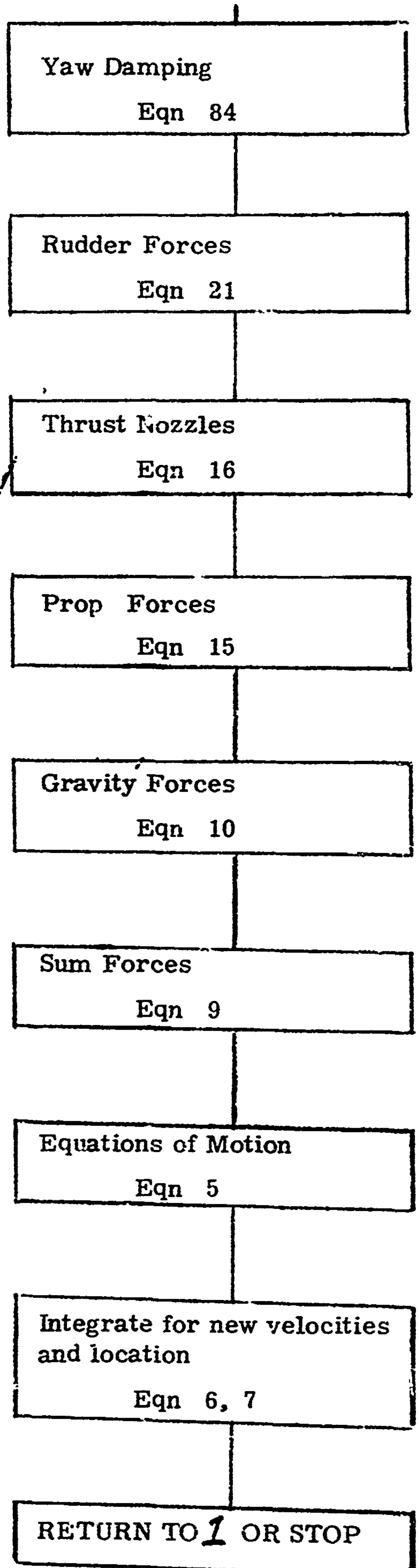
Pressure forces
Eqn 58

Spray and seal forces
Eqn 59

Momentum Drag
Eqn 64 to 69

Wind Forces
Eqn 70 to 77

Duct Forces
Eqn 78 to 83


Yaw Damping
Eqn 84
Rudder Forces
Eqn 21
Thrust Nozzles
Eqn 16
Prop Forces
Eqn 15
Gravity Forces
Eqn 10
Sum Forces
Eqn 9
Equations of Motion
Eqn 5
Integrate for new velocities and location
Eqn 6, 7
RETURN TO 1 OR STOP

REFERENCES

Doctors, L. and Sharma, S.: The Wave Resistance of an Air Cushion Vehicle in Acceleration Motion, University of Michigan, Dept of NAME, Report 099, Dec., 1970.

Haskind, M. D.: Oscillation of a Ship on a Calm Sea, Bulletin de l'Academie des Sciences de l'URSS, Classe des Sciences Techniques, 1946 No. 1, pp. 23-24. Translated in Soc. Naval Arch. Marine Engrs., Tech. & Res. Bull. No. 1-12, 45-60 (1953).

Lunde, J.K.: On the Linearized Theory of Wave Resistance for a Pressure Distribution Moving at Constant Speed of Advance of the Surface of Deep or Shallow Water, Skipsmodelltanken, Norges Tekniske Hogskole, Trondheim, Medd. No. 8, 48 pp. (1951).

Stoker, J.J.: Water Waves, Interscience Publisher, 1957.

Wehausen, J. W. and Laitone, E. V.: Surface Waves, Handbuch der Physik, vol. 9, pp. 446-778, Springer-Verlag, Berlin, 1960.

APPENDIX A

SOLUTION TO THE HYDRODYNAMIC PROBLEM OF A PRESSURE DISTRIBUTION MOVING WITH ARBITRARY SPEED ALONG AN ARBITRARY PATH

The following summarizes all the important mathematical details on the formulation and solution of the problem of a pressure distribution moving in any arbitrary manner in the horizontal plane. The craft is allowed to have any speed and any heading angle. The assumptions are that:

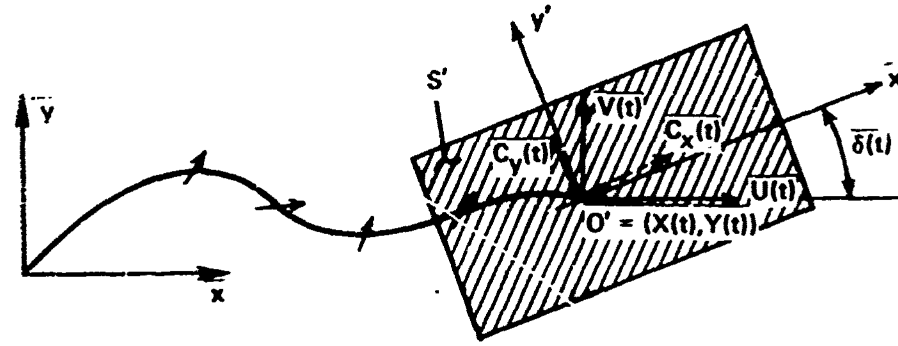
a. The fluid is inviscid
b. The flow is irrotational and potential; and,
c. The waves generated by the motion of craft is small enough so that the linearized free-surface condition can be used.

1. Coordinate Systems and Other Definitions

x, y, z are coordinates based on inertial frame of reference

x', y', z' are coordinates referred to a moving system which is mounted on the craft. Location and orientation of this system is given by $X(t)$, $Y(t)$, $\delta(t)$

z axes point upwards



Trajectory is given parametrically as function of time, t:

$$X(t) = \int_0^t U(\tau)\, d\tau = \int_0^t [C_x(\tau) \cos \delta(\tau) - C_y(\tau) \sin \delta]\, d\tau, \tag{103}$$

$$Y(t) = \int_0^t V(\tau)\, d\tau = \int_0^t [C_x(\tau) \sin \delta(\tau) + C_y(\tau) \cos \delta]\, d\tau, \tag{104}$$

$$C_x = U \cos \delta + V \sin \delta, \tag{105}$$

$$C_y = - U \sin \delta + V \cos \delta . \tag{106}$$

Relations between the two systems are:

$$\xi = \xi' \cos\delta(\tau) - \eta' \sin\delta(\tau) + X(\tau). \qquad (107)$$

$$\eta = \xi' \sin\delta(\tau) + \eta' \cos\delta(\tau) + Y(\tau).$$

and

$$\xi' = [\xi - X(t)] \cos\delta + [\eta - Y(t)] \sin\delta, \qquad (108)$$

$$\eta' = -[\xi - X(t)] \sin\delta + [\eta - Y(t)] \cos\delta.$$

2. Formulation in the Inertial Coordinate System

The pressure distribution in the moving coordinates is assumed to be given as $p'(\xi', \eta', \tau)$ in general, in the inertial system, $p(\xi, \eta, t)$. The region in which of $p(\xi, \eta, t)$ is nonzero, changes with time. The fluid is assumed to be of a constant depth h, and density ρ. Define the velocity potential $= \Phi(x, y, z, t)$, then continuity equation gives $\nabla^2\Phi = 0$.

Boundary Conditions

On Free Surface F: (z = 0)

$$\Phi_{tt}(x, y, 0, t) + g\Phi_z(x, y, 0, t) = -p_t(x, y, t)/\rho. \qquad (109)$$

On Bottom, β: (z = -h)

$$\Phi_z(x, y, -h, t) = 0. \qquad (110)$$

Here, g is the acceleration of gravity.

These conditions occur very frequently and derivations are available from any text book on free-surface hydrodynamics. See for instance, J. J. Stoker (1957, p. 191).

Initial Conditions

$$\Phi_t(x, y, 0, 0) = -g\zeta(x, y, 0) - p/\rho\,(x, y, 0) = 0, \qquad (111)$$

$$\Phi_z(x, y, 0, 0) = \zeta_t(x, y, 0) = 0. \qquad (112)$$

I. E. No initial disturbances except a static deformation of the free surface. Here ζ is the wave-height.

3. Solution in terms of the Time-Dependent Green Function

As usual, one starts with Green's Theorem for Φ_t,

$$4\pi\Phi_t(x, y, z, t) = \iint_{F \cup B \cup \Sigma R} [G(x, y, z; \xi, \eta, \zeta; t, \tau)\Phi_{t_\nu} - \Phi_t G_\nu]\, dS. \qquad (113)$$

where

$\frac{\partial}{\partial \nu} = (\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta}) \cdot \hat{n}$; G is a Green Function with a $1/r$ singularity, and Σ_R is a large vertical cylinder enclosing all the disturbances. It can be shown that the integral over Σ_R vanishes in the limit as R, the radius of the cylinder, approaches infinity, and after a number of operations* on (113), we may write:

$$4\pi\phi(x,y,z,t) = \frac{1}{\rho g} \iint_{-\infty}^{\infty} d\xi\, d\eta \int_0^t d\tau G_t(x,y,z,\xi,\eta,0,\tau,t) p_t(\xi,\eta,\tau). \tag{114}$$

In arriving at (114), we have made use of the inital and boundary conditions of ϕ and have assumed that G satisfies the following properties:

$$G = 1/r + H(x,y,z;\ \xi,\eta,\zeta,t,\tau),$$
$$\nabla^2 G = 0, \text{ for } (x,y,z) \neq (\xi,\eta,\zeta);\ \nabla^2 H = 0,$$
$$G_\zeta - \frac{1}{g} G_{tt} \Big|_{\zeta=0} = 0,$$
$$G_\zeta(x,y,z;\ \xi,\eta,-h;\ t,\tau) = 0,$$
$$G(x,y,z;\ \xi,\eta,\zeta,\ t,t) = G_t(x,y,z;\ \xi,\eta,\zeta,t,t) = 0. \tag{115}$$

Basically, G satisfies the free-surface, the bottom boundary conditions, and two initial conditions when the source is brought into existence at time equal to τ. The Green function was first derived by Lunde (1951), [by Haskind (1946) for the infinite depth case] and is available from Wehausen & Laitone (1960, p. 494).

In the present notation, the Green Function takes the following form:

$$G(x,\ y,\ z;\ \xi,\eta,\zeta;\ t,\ \tau_0) = 1/r + 1/r_2 - 2\int_0^\infty e^{-kh} \frac{\cosh k(h+\zeta)}{\cosh hk} \cosh k(z+h)\, J_0(kR)\, dk + 2\int_0^\infty dk \sqrt{gk} \frac{\cosh k(z+h)}{\cosh^2 kh \sqrt{\tanh kh}} \int_{\tau_0}^t d\tau \sin[\gamma(k)(t-\tau)] \cosh k(h+\zeta)\ J_0(kR) \tag{116}$$

*see for instance, J. J. Stoker, Water Waves, 1.9, p. 187, Interscience Publisher, 1957.

where

$$r^2 = (x-\xi)^2 + (y-\eta)^2 + (z-\zeta)^2,$$

$$r_2^2 = (x-\xi)^2 + (y-\eta)^2 + (z+\zeta)^2,$$

$$R^2 = (x-\xi)^2 + (y-\eta)^2,$$

$$\gamma(k) = \sqrt{gk \tanh kh}$$

and J_0 is the Bessel function of order zero. For our application, we are concerned only with $\zeta = 0$, and G therefore simplifies to

$$G(x, y, z; \xi, \eta, 0; t, \tau_0) = 2\int_0^\infty dk \frac{\cosh k(z+h)}{\sinh kh} [1 - \cos \gamma(k)(t-\tau_0)]$$

$$J_0(kR)\, dk. \tag{117}$$

Differentiating this and substituting it into equation (114), we get the following equation for $\phi(x, y, z, t)$:

$$\Phi(x, y, z, t) = \frac{-1}{2\pi\rho} \int_0^t d\tau \iint_{S(\tau)} d\xi\, d\eta\, p_t(\xi, \eta, \tau) \int_0^\infty dk$$

$$k \frac{\cosh k(z+h)}{\cosh kh} \frac{\sin\gamma(t-\tau)}{\gamma} J_0(kr) \tag{118}$$

4. <u>Solution in the Moving Coordinate System</u>

Since $p(\xi, \eta, \tau)$ can be expressed as $p'(\xi', \eta')$ in the moving coordinates, in which the integrals over (ξ, η) do not have limits depending on time, it is more convenient to write (118) in terms of x', y', z'. Using the coordinate transformation developed earlier, and noting that in the moving coordinate system:

$$\frac{\partial p}{\partial \tau} = \frac{\partial p'}{\partial \tau} + \frac{\partial p'}{\partial \xi'}\frac{\partial \xi'}{\partial \tau} + \frac{\partial p'}{\partial \eta}\frac{\partial \eta'}{\partial \tau}, \tag{119}$$

it can be shown that after some fairly straight forward calculations:

$$\Phi(x', y', t) = \frac{-1}{4\pi^2\rho} \int_0^t d\tau \iint_{-\infty}^{\infty} dw\, du \frac{\sin\gamma(t-\tau)}{\gamma} \frac{\operatorname{ch} k(z+h)}{\operatorname{ch} kz} \iint_S d\xi' d\eta'$$

$$\left\{\frac{\partial p'}{\partial \tau} - i\, p'(\xi', \eta', \tau) \div [w(U(\tau) - (\xi' \sin\delta + \eta' \cos\delta)\dot{\delta}) + u(V(t)\right.$$

$$\left. + (\xi' \cos\delta - \eta' \sin\delta)\dot{\delta})]\right\} \cdot e^{iw[X(x', y', t) - X(\xi', \eta', \tau)]}$$

$$+ iu[Y(x', y', t) - Y(\xi', \eta', \tau)], \qquad i = \sqrt{-1} \tag{120}$$

where

$$X(x', y', t) = X' \cos\delta(t) - Y' \sin\delta(t) + X(t)$$

$$Y(x', y', t) = X' \sin\delta(t) + Y' \cos\delta(t) + Y(t). \qquad (121)$$

In arriving at (120), we have used an integral representation of $J_0(kR)$ to convert the k-integral to the w-u double integral. Here

$$w^2 + u^2 = k^2.$$

Equation (120) is the solution to the problem of an arbitrarily moving and spinning craft. This is the analog of Equation (2-21) of the report by Doctors and Sharma (1970), now extended for 3 degrees of freedom instead of just rectilinear motion.

Note, if the pressure in the moving system is nontime dependent, the first term $\frac{\partial p}{\partial \tau}$ drops out. Note also if $\dot{\delta} = 0$, $\delta \neq 0$, this reduces to a solution corresponding to purely translational motion with a drift angle but no spinning. If we had formulated the problem in the moving coordinates, Equation (120) will be the solution. A good check will be to start with (120) and see if the free-surface condition in the rotating system will be satisfied. The free-surface condition is the version linearized with respect to disturbance caused by p; no linearization with respect to the smallness of angular rotation is necessary.

5. General Expression for Wave-Height

To obtain the wave-height, one simply uses equation (111) now, [RHS≠0] which relates the wave-height to the time-derivative of the potential, but must bear in mind the meaning of the time differentiation in a moving frame. Upon performing an integration by parts of the $\frac{\partial p}{\partial \tau}$ term, with respect to τ, we obtain the following simplified expression:

$$\zeta(x', y', t) = \frac{-1}{4\pi^2 \rho g} \iint_{-\infty}^{\infty} dw\, du \iint_{S'} d\xi'\, d\eta'\, p'(\xi', \eta', \tau) \cos \gamma t \, \mathrm{Exp}\{ iw[X(x'y't) - \xi']$$

$$+ iu[Y(x', y', t) - \eta']\} - \frac{-1}{4\pi^2 \rho g} \iint_{-\infty}^{\infty} dw\, du \iint_{S'} d\xi'\, d\eta' \int_0^t d\tau$$

$$p(\xi', \eta', \tau)\, \gamma \cdot \sin \gamma(t-\tau) \cdot \mathrm{Exp}\{ iw\, [X(x', y', t) - X(\xi', \eta', \tau)]$$

$$+ iu\, [Y(x', y', t) - Y(\xi', \eta', \tau)]\}. \qquad (122)$$

with X(x, y, t) and Y(x', y', t) defined as before.

This expression has the advantage that one only needs to keep track of X(t), Y(t), and $\delta(t)$, instead of, in addition, $C_x(t)$, $C_y(t)$, and $\delta(t)$. Equation (122) is the wave-height seen in the craft's frame, wave generated as a result of a given history of motion prescribed by X(t), Y(t), and $\delta(t)$.

To evaluate some of these integrals analytically, we must now know the functional form of $p'(\xi', \eta', t)$.

6. Functional Form of the Pressure Distribution

Equation (122) is a quintuple integral. The (ξ', η') integration corresponds essentially to taking the two-dimensional Fourier transform in space of the pressure-distribution function $p(\xi', \eta', \tau)$. The (w, u) integrals correspond to the inversion of the transforms, and the time integral represents an integration of the effects created by the craft's trajectory and orientation in the past. It should be noted that for the simulation problem being considered at hand, it is impossible to calculate the time integral ahead of time. This is so because the trajectory and orientation of the craft depends on the actions taken by the pilot, actions which are unknown before the simulation.

Because of the immense generality of the problem and complexity of the integrals, it will be to our advantage to be able to perform the (ξ', η') integrals analytically. We suggest here adopting the hyperbolic-tangent distribution used by Doctors and Sharma (1970). The functional form of $p(\xi', \eta', \tau)$ is given by

$$p(\xi', \eta', \tau) = p_0 \frac{P(\tau)}{4} \left[\tanh \alpha\left(\frac{\xi'}{a} + 1\right) - \tanh \alpha\left(\frac{\xi'}{a} - 1\right)\right]$$

$$\times\left[\tanh \beta\left(\frac{\eta'}{a} + \frac{b}{a}\right) - \tanh \beta\left(\frac{\eta'}{a} - \frac{b}{a}\right)\right] \qquad (123)$$

where α, β are diffusion parameters in the longitudinal and the transverse directions respectively; the larger they are, the closer the shape is to that of a step distribution. a, b in (123) are the half-lengths and half beams of the craft respectively. $P(\tau)$ is a time modulation factor which can be and will later be set equal to 1.0 if the pressure is assumed to be nontime dependent in the craft's frame, and p_0 is the nominal pressure. For this distribution, p_0 is given by

$$p_0 = \Delta / 4ab \qquad (124)$$

where Δ is the displacement of the craft.

With this assumption of the functional form of $p(\xi', \eta' \tau)$ the (ξ', η') integrals can be evaluated analytically, as done earlier by Doctors and Sharma (1970, pp. 37-40). The result is

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(\xi', \eta',) e^{i\,[(w\cos\delta + u\sin\delta)\xi' + (-w\sin\delta + u\cos\delta)\eta']}\, d\xi'\, d\eta'$$

$$= \frac{\pi^2\, p_0 P(\tau)}{\alpha\beta} \; \frac{\sin aw' \cdot \sin bu'}{\sin h(\frac{\pi w'}{2\alpha}) \sin h\,(\frac{\pi u'}{2\beta})}\;, \tag{125}$$

where the notations

$$w' = w\cos\delta + u\sin\delta,$$

$$u' = -w\;\sin\delta + u\cos\delta\;, \tag{126}$$

have been used, bearing mind w and u are variables of integration in the next step.

Next, substituting (125) back into (122) and changing variables of integration from (w, u) to (w', u') as defined in (126) we get

$$\zeta\,\frac{(x', y', t)}{p_0/4pg} = -P(0)\, K^C\,[R_x(t),\; R_y(t),\; t]$$

$$-\int_0^t d\tau P(\tau)\, K^S\,[R_x(t-\tau),\; R_y(t-\tau),\; t-\tau]\,, \tag{127}$$

where the Kernel functions are defined as follows:

$$\begin{matrix} K^C \\ K^S \end{matrix}(R_x,\; R_y,\; t) = \iint_{-\infty}^{\infty} dw'\, du'\; e^{iw'R_x + iu'R_y} \begin{pmatrix}\cos\gamma t \\ \sin\gamma t\end{pmatrix} \frac{\sin(aw')\sin(u'b)}{\alpha\beta\, \mathrm{sh}\,(\frac{\pi}{2\alpha}w')\mathrm{sh}(\frac{\pi}{2\beta}u')}$$

with (128)

$$\gamma = \sqrt{gk\tanh kh}\;, \qquad k^2 = w'^2 + u'^2\,.$$

It can be seen that K^C and K^S are functions of three variables for given values of α and β. Also, the quantities $R_x(t-\tau)$, $R_y(t-\tau)$ are merely symbolic since they do not depend explicitly on the combination $(t-\tau)$. Physically, they have the meaning of being the x- and y- coordinates of the point (x', y') in the craft's system at time equals t, expressed in the coordinate system of the craft at time equals to τ. For a given set of (x', y') at time t, $R_x^{(t-\tau)}$ and $R_y^{(t-\tau)}$ may then be calculated as follows:

$$R_x = [X(x', y', t) - X(\tau)]\cos\delta(t) + [Y(x', y', t) - Y(\tau)]\sin\delta(\tau),$$

$$R_y = -[X - X(\tau)]\sin\delta(\tau) + [Y - Y(\tau)]\cos\delta(\tau),$$

where (x, y) are given by Equations (121).

It is worthwhile to point out at this point that a good choice of the values for α and β are $\alpha=\beta=5.0$. As discussed in Doctor's and Sharma's report, this choice though appeared arbitrary, provide two advantages. First, it effectively eliminates the unrealistic hollows and humps of the wave resistance curve at low speeds. Second, the computational efforts involved in evaluating the wave-number integrals can be reduced considerably because, as can be seen from Equation (128), the decay factor is dependent on $\frac{\pi}{2\alpha}$ and $\frac{\pi}{2\beta}$ for the (w-u) integrals.

7. Nondimensional Expression for the Free-Surface Elevation

Equation (127), developed earlier for the free-surface elevation, consists of two terms. The term associated with the function K^C can be thought of as the wave system generated as a result of the initial building up of the cushion pressure. This system of waves can be neglected if the forward motion of the craft does not occur "immediately" after the build up. If the first term is neglected, what is left then is a convolution integral in time involving the Kernel function K^S. This integral can be evaluated if the quantities X(t), Y(t), and δ(t) are given.

In actual computations, it is much more convenient to use a nondimensional form of Equation (127) and measure time t with the present being equal to zero. Therefore nondimensionalizing all physical lengths by the craft's half-length a, and time by $\sqrt{a/g}$, for example,

$$\begin{aligned} \tilde{x} &= x'/a \\ \tilde{w} &= w'a \\ \tilde{t} &= (t-\tau)/\sqrt{a/g} \\ \tilde{\gamma} &= [\tilde{k} \tanh \tilde{k} (h/a)\cdot]^{1/2}, \text{ etc.,} \end{aligned} \tag{129}$$

(127) becomes:

$$\begin{aligned} &\tilde{\zeta}(\tilde{x},\tilde{y},\tilde{t}=0) = \zeta(x',y',t)/(p_0/4\rho g) \\ &\simeq \int_{-T_0}^{0} d\tilde{\tau} P(\tilde{\tau})\, K^S(\tilde{R}_x, \tilde{R}_y, \tilde{\tau}) \end{aligned} \tag{130}$$

where

$$K^S(\tilde{R}_x, \tilde{R}_y, \tilde{t}) = \iint_{-\infty}^{\infty} d\tilde{w}\, d\tilde{u}\, e^{i(\tilde{w}\tilde{R}_x+\tilde{u}\tilde{R}_y)}\cdot\tilde{\gamma}\sin\tilde{\gamma}\tilde{t}\cdot\frac{\sin\tilde{w}\sin\tilde{u}\,(\frac{b}{a})}{\alpha\beta\,\mathrm{sh}(\frac{\pi\tilde{w}}{2\alpha})\,\mathrm{sh}(\frac{\pi\tilde{u}}{2\beta})}. \tag{131}$$

with

$$\begin{aligned} a\tilde{R}_x(\tilde{\tau}) &= [X'-X(\tau)]\cos\delta(\tau) + [Y'-Y(\tau)]\sin\delta(\tau), \\ a\tilde{R}_y(\tilde{\tau}) &= -[X'-X(\tau)]\sin\delta(\tau) + [Y'-Y(\tau)]\cos\delta(\tau). \end{aligned} \tag{132}$$

In writing down (132), we have assumed that $X(\tau)$, $Y(\tau)$, $\delta(\tau)$, are given in the craft's present coordinate system. In other words, $X(\tau=0) = Y(\tau=0) = \delta(\tau=0) = 0$, $X(\tau)$ and $Y(\tau)$ are negative if the craft is moving forward and swaying to port; and $\delta(\tau)$ is positive if the craft is turning to starboard. In so doing, we can dispense with the inertial coordinate system used earlier in formulating the problem and it is then only necessary to know the history of the trajectory and orientation, measured relative to the present position, rather than the absolute value measured in an inertial frame.

APPENDIX B

THE KERNEL FUNCTION, $K^S(R_x, R_y, t)$

In this appendix, mathematical and numerical techniques for evaluating the kernel function $K^S(R_x, R_y, t)$ are described. Consider the double integral defined by eqn (131) of appendix A.

$$K^S(R_x,R_y,t) = \iint_{-\infty}^{\infty} dwdu \; e^{i[wR_x+uR_y]} \cdot \gamma \sin \gamma t \cdot \frac{\sin w \cdot \sin \frac{b}{a} u}{\alpha\beta sh\left(\frac{\pi w}{2\alpha}\right) sh\left(\frac{\pi w}{2\beta}\right)} .$$

Here for simplicity of notation, we omit all the titles above the dimensionless variables with the understanding that all quantities under consideration in this appendix are dimensionless.

THE K^S INTEGRAL IN POLAR COORDINATES. Assuming $\alpha = \beta$, i.e., the diffusion parameter is the same in both longitudinal and transverse directions and writing the (w, u) integral in polar coordinates (k, θ), we have

$$K^S(R_x,R_y,t) = 4\int_0^{\infty} kdk\gamma\sin\gamma t \int_0^{\pi/2} d\theta \; \frac{\cos(kR_x\cos\theta)\sin(k\cos\theta)\cos(kR_y\sin\theta)\sin(k\mu\sin\theta)}{\alpha^2 \sin h\,(k'\cos\theta)\, \sinh\,(k'\sin\theta)} \tag{133}$$

where

μ	=	b/a	=	beam to length ratio	(134)
α'	=	$\pi/2\alpha$	=	a decay factor, characteristic of the distribution	
k'	=	$\alpha' k$.			

Next, we define:

$$R_{1j} = [(R_x+1)^2+(R_y \pm \mu)^2]^{1/2}, \qquad j = 1,2$$

$$R_{2j} = [(R_x-1)^2+(R_y \pm \mu)^2]^{1/2}, \qquad j = 1,2 \tag{135}$$

$$\psi_{ij} = \tan^{-1}\left(\frac{R_y \pm \mu}{R_x \pm 1}\right) , \qquad \psi = [0., 2\pi].$$

R_{ij} and ψ_{ij} are, basically, polar coordinates of (R_x, R_y) referred to the four corners of the craft. K^S can now be written as follows:

$$K^S(R_x, R_y, t) = \int_0^\infty \frac{kdk}{2} \gamma \cdot \sin\gamma t \cdot [\sum_{i=1}^{2} \sum_{j=1}^{2} (-)^{i+j} Q(k, R_{ij}, \psi_{ij})] \tag{136}$$

where

$$Q(k, R, \psi) = \frac{1}{2} \int_0^{\pi/2} d\theta \frac{\cos[kR\cos(\theta - \psi)] - \cos[kR\cos(\theta + \psi)]}{\sin h(k' \cos\theta) \sin h(k' \sin\theta)} \tag{137}$$

Our immediate task, evidently, is to be able to compute the theta-integral defined by (137).

THE THETA-INTEGRAL $Q(k, R, \psi)$. It can be seen from (137) that the Theta-Integral is a highly oscillatory integral, particularly for large values of kR. The magnitude of the integrand, however, is exponentially small for large values of k'. To avoid handling this oscillatory integral numerically, it was found that the following Bessel series expansion yields excellent accuracy with minimal amount of numerical computations:

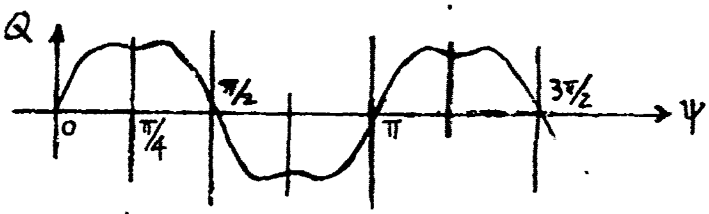
$$Q(k, R, \psi) = \sum_{n=1,3,5...}^{\infty} (-)^n C_n(k') J_{2n}(kR) \cdot \sin 2n\psi \tag{138}$$

where J_{2n} is the Bessel function of the first kind, of order 2n, and

$$C_n(k') = 4\int_0^{\pi/2} \frac{\sin 2n\theta d\theta}{\sin h(k'\cos\theta) \sin h(k'\sin\theta)}, \quad k' \neq 0. \tag{139}$$

This coefficient, in general, has to be evaluated numerically, but is a function of two parameters only, n, k'. The Bessel functions can be computed by standard mathematical library-routines.

Because n is odd in (138), one sees immediately that $Q(k, R, \psi)$ is an even function of about $\psi = \pi/4$; $5\pi/4$; $9\pi/4$, and $11\pi/4$, and odd about $\psi = 0$, $\pi/2$; π; $3\pi/4$. For example, we expect the behavior of Q as a function ψ as follows:



The behavior of Q in $[0, \pi/4]$ defines its entire behavior completely. The number of terms that one needs to take in (138) will depend on the magnitude of kR. To estimate the number of terms required, we note that for a given value of z, a large-n approximation for $J_n(z)$ is:

$$J_n(z) \approx \frac{1}{\sqrt{2\pi n}} \left(\frac{ez}{2n} \right)^n \qquad (140)$$

Hence, n can be so chosen that

$$\frac{1}{2} \log (2\pi n) + n \log \left(\frac{ez}{2n} \right) < \log \epsilon \qquad (141)$$

where ϵ is a measure of the truncation error for terminating the series with a finite number of terms, e.g., $\epsilon = 10^{-6}$. It is also evident from (140) that n in (138) should be greater than (ekR / 4).

Analysis of the coefficient $C_n(k')$ indicates that $Q(k, R, \psi)$ behaves approximately as follows:

$$Q(k,R,\psi) \approx \frac{-1}{k' \sin h\, k'} \sum_{n=1,3,5\ldots}^{\infty} C_n^*(k')\, J_{2n}(kR) \sin (2n\psi), \qquad (142)$$

where $C_n^*(k')$ is "almost" a constant function of k'. Note that Q decays like $1/k'e^{k'}$. The advantage of (138) lies in that the oscillatory behavior of Q is kR is completely contained in the Bessel functions.

It may be shown that $C_n(k')$ has the following asymptotic form for large value of n:

$$\lim_{n\to\infty} C_n(k') = \frac{1}{k'\sinh k'} \left\{ \pi \tanh \left(\frac{n\pi}{k'}\right) - 4e^{k'} \frac{\pi}{4} \cdot \sin \frac{n\pi}{2} \sum_{s=1}^{\infty} \frac{e^{-s\frac{k'\pi}{2}}}{(2s+1)^2 + \frac{(2n)^2}{k'}} \right\} \qquad (143)$$

which indicates that $C_n(k')$ approaches a constant. Such behavior is to be expected because this integral is reminiscent of the more well-known integral

$$\int_0^{\pi/4} \frac{\sin n\theta}{\sin 2\theta}\, d\theta = \frac{\pi}{2}, \text{ for all n = odd.} \qquad (144)$$

It was found that this asymptotic formula is exceedingly good for $n > 10$. In practice, for low values of n, Simpson's rule was applied to evaluate $C_n(k')$ numerically over a range of values of k', (k'= [.001, 125.]). For each value of n, the function $C_n(k')$ is fitted with a spline curve, and the spline coefficients are stored on cards. This idea results in a substantial saving of computation time. For large values of n, $(n > 10)$ the formula (143) is used. Only the first term in the approximation was necessary. Having determined the number of terms required in the series, and having generated the $C_n(k')$, n = 1,3,5,... either by interpolation, or by using (143), we reduce the calculation of $Q(k, R, \psi)$ to merely a summing process as defined by (138).

The routine package, Q24, described above puts into practice the technique described herein. It also takes advantage of the fact in the double sum of (136):

$$\sum_{i=1}^{2} \sum_{j=1}^{2} (-)^{i+j} Q(k, R_{ij}, \psi_{ij}) = \sum_{n=1,3,5\ldots}^{\infty} C_n(k') \cdot \left[\sum_{i=1}^{2} \sum_{j=1}^{2} J_{2n}(kR_{ij}) \sin(2n\psi_{ij})\right]. \quad (145)$$

Thus redundant calculations of $C_n(k')$ can be avoided.

NUMERICAL COMPUTATION OF THE K-INTEGRAL. To obtain the values of K^{θ}, it is necessary to integrate the Q-function with respect to k. In practice, the infinite upper limit has to be replaced by a finite value. Since $Q(k, R, \psi)$, according to equation (142), behaves approximately as $[k'e^{k'}]^{-1}$ for large k', we may estimate the truncation error ϵ, incurred as a consequence of stopping the integration at k_o, by the following formula:

$$\epsilon \leq \frac{1}{\alpha 2} \int_{k_o}^{\infty} k\gamma \left|\sin\gamma t\right| \frac{dk}{k' \sinh k'} \simeq \left(\frac{2}{\pi}\right)^2 \left(\frac{2\alpha}{\pi}\right)^{1/2} \int_{k'_o}^{\infty} \sqrt{k'}\, e^{-k'} dk',$$

$$= \left(\frac{2}{\pi}\right)^2 k_o^{1/2} e^{-\frac{\pi}{2\alpha} k_o} . \quad (146)$$

For a value $\epsilon = 10^{-4}$, k_o amounts to 31.8, which yields a maximum value of $k_oR = 750$, if R is less than or equal to 23.5. The value k_oR is important because it determines the number of terms required in the Bessel series defined by equation (138).

The integrand of the k-integral defies simple analytical description. Oscillation arises from both the sin αt term as well as from the function Q(kR). It was observed that the period of oscillation of the Q-function is approximately 2π, with respect to the variable kR. This is merely an approximation because this function is hardly sinusoidal. In order to minimize the number of evaluations of the integrand, which requires a considerable amount of computational efforts, it is necessary to have some knowledge of the approximate separation between, say, successive peaks. The following model was found useful in subdividing the interval $k=[0., k_o]$ into a number of subintervals. Let K* satisfy the relation:

$$k^* / \tanh k^* (h/a) = 1.5 \quad (147)$$

then for $k < k^*$, an approximate "period" of oscillation p_r is given by

$$p_r = 2\pi / [R^* + \sqrt{\tilde{h}}\ t], \tag{148}$$

where $R^* = [R_x^2 + R_y^2]^{1/2}$, $\tilde{h}$ is the water-depth/half-length ratio, and t is the nondimensional time parameter in the kernel. For $k > k^*$,

$$p_r = 2\pi / [R^* + t/\sqrt{k}\ ] \tag{149}$$

In each of the subintervals, determined either by (148) or (149), the Gauss-Legendre quadrature formula is applied. This was found to be the most efficient means of performing the k-integral numerically, without having to have to sacrifice accuracy. For an absolute accuracy of 10^{-4} quadrature formula of order 10 for $k < k^*$ and of order 7 for $k < k^*$ were found to be acceptable for this purpose.

TABULATION OF THE KERNEL FUNCTION, $K^S(R_x, R_y, t)$. In the discussions above, we have outlined the numerical procedures for evaluating K^S. The computations are so involved that real time evaluation of K^S is beyond the question. The approach we suggest is to calculate and tabulate this function before the time of simulation. This table, which is three-dimensional, can be stored in memory and then an interpolation procedure, as detailed in Section IV, can be used to obtain the proper value of K^S. For convenience, we recall the definition of K^S:

$$K^S(R_x, R_y, t) = \frac{1}{\alpha 2} \int_0^\infty dk\, k\gamma \sin\gamma t \left[\sum_{i=1}^{2} \sum_{j=1}^{2} (-)^{i+j} Q(kR_{ij}, \alpha k, \psi_{ij}) \right], \tag{150}$$

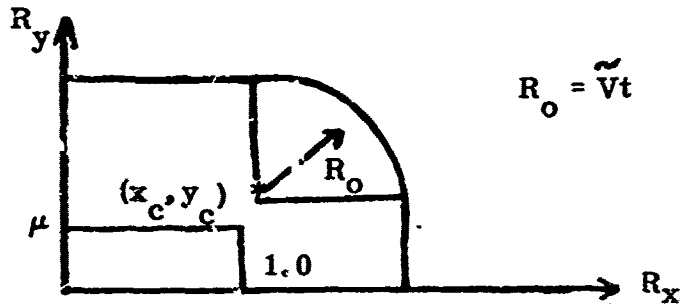
where

$$Q(kR, k', \psi) = -\sum_{n=1,3,5\ldots}^{\infty} C_n(k')\, J_{2n}(kR) \sin 2n\psi, \tag{151}$$

$$\gamma(k) = \sqrt{k \tanh k(h/a)}, \tag{152}$$

and R_{ij}, ψ_{ij}, are given in terms of R_x, R_y by equation (135) above. The physical meaning of K^S is as follows: If an impulse pressure-distribution of the shape given below is applied on the free surface at time equal to zero in the form of a delta function, $K^S(R_x, R_y, t)$ describes the response at the time t, of the free surface located at (R_x, R_y). The free-surface elevation, ζ, is given by $-(p_o/p_g)K^S$. Therefore, K^S is proportional to the wave height and contains all the response characteristics of a fluid with a constant but finite depth. From this interpretation, one sees immediately that the region of nonzero disturbance will grow in time. According to linearized water-wave theory, we know that the wave front, the wave with the longest wave-length, will propagate the fastest and that the maximum speed is at most $\sqrt{gh}$, where h is the water depth.

In nondimensional form, this speed is: $\tilde{V} = (h/a)^{1/2}$. Thus, for a given value of t, one may estimate the size of the region in which K^S is nonzero as follows:



If the pressure distribution had been a step function $(x_c, y_c)=(1.0, \mu)$. For the distribution that we have used, a good choice is (4., 2.5).

The program TABKS, a listing of which is attached, was developed for the purpose of generating the Kernel function K^S. Referring to (150), we see that the important physical parameters are:

$\tilde{h}$ = water depth to half-length ratio
α = diffusion parameter of the distribution
μ = beam/length ratio of craft

In addition to these quantities, we must specify the array of values of R_x, R_y and t, for which the function K^S is to be calculated. The computer program TABKS ready in the array for t, TMS, the array for R_x, RS, and the array for Ry RY. k_o, defined by equation (146) is then determined by calling the routine FINDKO. Next, for each combination of the values of R_x, R_y and t, routine INTDK2 is called to evaluate the k-integral according to the steps described above. The results, i.e., the values of K^S, are then punched out on cards (File #7). Two options are available for using INTDK2. If the parameter IACY is set equal to 1, an accuracy of approximately $\pm 10\%$ will be used as criterion for calculating the k-integrals. This option is useful when some rough ideas about the behavior of K^S is desired. If IACY is set equal to 2, accurate calculations of more than 1% accuracy will be performed, and ϵ in (146) will be set equal to 10^{-4} for the determination of k_o. Evidently, in order to obtain the integrand for the k-integral, the theta-integral, $Q(k, R, \psi)$ must be evaluated. This is achieved by calling the routine Q24 (see the fourth to last statement in routine WNING), which is a major component of the Q24 package detailed in the last section of this appendix. On an IBM-370 model 65 machine, typical computation time for each value of K^S ranges from 0.5 to 5.0 seconds depending on the magnitude of the values of R and t (see equation (148)).

In figures B-1 to B-13, we have plotted graphically the behavior of K^S vs. R_x, for different values of R_y. Each graph corresponds to a different value of the nondimensional time. These values were generated using:

$\tilde{h} = 1.0$; $\alpha' = \pi/10$; $\mu = 0.5$.

which is applicable to the ACV being considered. The hump speed corresponds to this particular geometrical configuration as approximately 15.0 knots. The table was generated for time-values of t=0.125, 0.25, 0.375, 0.5, 0.75, 1.0, (0.25), 6.0. In the notation of Section IV, ΔT is 0.25. The range of R_x values was chosen to include the entire region of nonzero disturbance. However, the range of R_y is only [1., 1.5]. Hence, this table is not sufficient for the application of the general case of a radical maneuver, which requires R_y to have the same range as R_x. However, a small amount of sway and yaw is permissible.

The following points will be useful for future preparation or extension of the kernel table:

a. A grid spacing of R_x, or R_y of 0.1 is quite sufficient if linear instead of parabolic interpolation is used in the procedure described in Section IV. More sparse grid-spacing can be used if higher order interpolation function is used. This becomes a tradeoff problem between the storage requirement and the available computational time for performing the convolution integral during real time simulation.

b. The value of the kernel function, K^S, varies from 0. to a maximum value of $\pm$ 1.75. In general, it is sufficiently accurate to round off the value of K^S to the fourth decimal place. Therefore, instead of using a REAL*4 variable to store K^S, an integer variable of 2 bytes, i.e., INTEGER*2, is sufficient. This will ameliorate the situation of storage requirement.

c. Storage space can also be saved if one makes use of the fact for small values of t, the table-size is smaller. Hence, a three-dimensional array such as V(50,50,40) (see Section IV) will be wasteful in storage allocation. A better alternative will be, for example V1(30,30), V2(32,32), V3(36,36)...V40(50,50). This would amount to approximately 76,000 half-words, or 38,000 full words of four bytes each.

```
C    ***** *****          PROGRAM  *TASKS*
C    PROGRAM FOR CALCULATING THE KERNEL FUNCTION KS(RX, RY, T)
C    ***** *****
C
      COMMON/RNPARM/ALPP,ST,DELT,NTS,FLDPT(50,2),NFP,ERR,HT
      COMMON/STATUS/AT,BT,TIME,STATE(50,3),KERNEL(50,50)
      COMMON/CNSTAN/PI,PI2,MACCY,FACTOR
      COMMON/PRIPUN/IR,IW,NOPRT
       COMMON /SAVE/ R(4), SI(4), QF(4), NORDR(4)
      COMMON /LIMIT/ AKU, ALPHA, BREAK, AL2
      REAL*8 PI, O24, ANS(20)
      REAL*4 KERNEL,LENGTH
       DIMENSION RX(50), RY(30), TMS(30), P(4,2)
      DIMENSION SOLN(65), FRR(65), FSI(65)
C
      CALL ERRSET (208,999,-1,1)
      PI = 3.1415926535898
      PI2 = .5*PI
      MACCY=6
      FACTOR = 2.
C
      NOPRT = -1
      ST=0.5
      ALPP = .314159
      HT=1.
      ALPHA = PI2/ALPP
      AL2 = .25/(ALPHA**2)
      BREAK =1.5
      XC = 4.
      YC = 2.5
      IACY = 1
      ERR = .05
C
      CALL PREP
C
C    VARIABLES DEFINITION..
C    ST=BEAM/LENGTH RATIO
C    ALPP = PI/(2.*ALPHA), WHERE ALPHA IS NON-DIMENSIONAL , I.E. (LARRY
C          DOCTOR'S ALPHA)*A , WHERE A= HALF-LENGTH OF CRAFT.
C          RS, ALPHA)*A
C    HT = DEPTH OF HALF-LENGTH RATIO.
C    ERR = ERROR ALLOWED IN THE K-INTEGRAL
C
C    TMS = TIME = NON-DIMENSIONAL TIME VALUE,  PRESENT= 0.
C        = ACTUAL TIME INTERVAL * SQRT(G/A)
C    RX = X-COORDINATE OF FIELD POINT
C    RY = Y-COORDINATE OF FIELD POINT
C    IACY IS A PARAMETER CONTROLLING ACCUARCY OF THE CALCULATIONS.
C    IACY=1, FOR ROUGH CALC.
C    IACY=2 FOR GOOD ACCURACY.
C
C    READ IN ARRAY OF FIELD POINTS
      READ (5,1001) NT, (TMS(K), K=1,NT)
       READ(5,1001) NX, (RX(I), I=1,NX)
      READ(5,1001) NY, (RY(I), I=1,NY)
C    READ IN ACCURACY CONTROL SIGNAL
```

```
      READ (5, 1033) IACY
 1001 FORMAT (I5/(F10.4))
 1002 FORMAT (F10.4)
 1033 FORMAT ( I5)
      IF ( IACY .EQ. 2 ) ERR = .01
C
C    CALCULATE APPROX. UPPER AK- LIMIT FOR GIVEN ERR.
      CALL FINDKO (AKU, ERR, ALPHA )
C
      DO 100  K=1, NT
      TIME = TMS(K)
      RD2 = HT*TIME**2 *1.1
      RD = SQRT(RD2)
C
C  PUNCH OUT GRID SYSTEM FOR RECORD.
      WRITE (6,1012) TIME       , (RX(J), J=1, NX)
 1012 FORMAT (1H0, ' NON-DIMENSIONALIZED TIME VALUE =', F10.5 , /,
     1 1X, 'RX(AT) -',3X, 9F12.4 /( 12X,9F12.4))
      WRITE (7, 1013) TIME
 1013 FORMAT ( '  KERNEL-TABLE FOR TIME =', F10.5 )
      WRITE (7,1015) NX, ( RX(I), I=1, NX)
 1015 FORMAT ( 'RX GRID, NO. OF GRID POINTS, NX=' , I5 /(8F10.5))
      WRITE (7,1016) NY, ( RY(J), J=1, NY)
 1016 FORMAT ( 'RY GRID, NO. OF GRID POINTS, NY=' , I5 /(8F10.5))
      WRITE (6,1024)
 1024 FORMAT (2X, ' RY=BT')
C
      DO 300  J=1, NY
      BT = RY(J)
      P(1,2) = BT+ST
      P(2,2) = BT-ST
      P(3,2) = P(1,2)
      P(4,2) = P(2,2)
      CALL TIMING( IJK)
C
      DO 200  I=1, NX
      AT = RX(I)
      P(1,1) = AT+1.
      P(2,1) = P(1,1)
      P(3,1) = AT-1.
      P(4,1) = P(3,1)
C
C  SET-UP CORNER COORIDINATES.
       DO 120 M=1,4
       R(M) = SQRT(P(M,1)**2+P(M,2)**2)
      SI(M) = 0.
      IF ( ABS(P(M,1)) .GT. 1.E-10 .OR. ABS(P(M,2)) .GT. 1.E-10 )
     1SI(M) = ATAN2(P(M,2), P(M,1))
  120 CONTINUE
C
      FRR(I) = SQRT (AT**2 + BT**2)
      IF ( FRR(I) .LT. 0.5 ) FRR(I) = 0.5
      FSI(I) = 0.
      IF ( ABS(BT) .GT. 1.E-10 .OR. ABS(AT) .GT. 1.E-10 )
     1FSI(I) = ATAN2(BT, AT)
C
C  CHECK IF GRID-POINT EXCEEDS WAVE-FRONT.
      IF ( BT .LE. YC  .AND.  AT .GE. (XC+RD)) GO TO 310
      IF ( AT .LE. XC  .AND.  BT .GE. (YC+RD)) GO TO 310
      IF (((BT .GE. YC) .AND. (AT .GE. XC)) .AND.
```

```
     1   (((BT-YC)**2 + (AT-XC)**2) .GT. RD2)) GO TO 310
C
C  CALCULATE THE INTEGRAL BY CALLING *INTDK*
C  VALUE RETURNED IN ARRAY *SOLN*
      CALL INTDK2(FRR(I), SOLN(I), IACY )
      GO TO 200
C
  310 SOLN(I) =0.
  200 KERNEL(I,J)= SOLN(I)
C
      CALL TIMING (IJL)
      SECS = .01*(IJL-IJK)
C
C  PRINT-OUT LOOP.
C
      DO 400  M=1, 10
      M1 = (M-1)*9+1
      M2 =M1+8
      IF ( M2 .GT. NX)  M2= NX
C
      IF ( M .NE. 1 ) GO TO 420
                          WRITE (6,1003)RT,M,(SOLN(L), L=M1, M2)
 1003 FORMAT (1X, F7.4, I6, 9F12.4 )
                          WRITE (7,1005)J,M, (SOLN(L), L=M1, M2)
 1005 FORMAT ( I5, I3, 9F8.5)
      GO TO 410
C
  420 WRITE (6,1006)M (SOLN(L), L=M1, M2)
 1006 FORMAT (8X, I6, 9E12.4)
      WRITE (7,1007)M,(SOLN(L), L=M1, M2)
 1007 FORMAT ( I8, 9F8.5)
C
  410 IF ( M2 .EQ. NX) WRITE(6, 1004) SECS
 1004 FORMAT ( 1H+, 121X, F8.2 )
      IF ( M2 .GE. NX) GO TO 300
  400 CONTINUE
C
  300 CONTINUE
  100 CONTINUE
      STOP
      END
```

```
      SUBROUTINE INTDK2(RR, ANS, KPT)
      REAL*8 VK, AKO, ANS1, ANS2, AK
      REAL*8 ANS3, ANS4
      COMMON/RNPARM/ALPP,ST,DELT,NTS,FLDPT(50,2),NFP,FPP,HT
      REAL*4 KERNEL
      COMMON/STATUS/AT,BT,TIME,STATE(50,3),KERNEL(50,50)
      REAL*8 PI
      COMMON/CNSTAN/ PI,PI2,MACCY,FACTOR
      COMMON/PRIPUN/IR,IW,NOPRT
      COMMON /LIMIT/ AKU, ALPHA, BREAK, AL2
      REAL*8 WW,ZZ, W, Z
      COMMON /LAGUER/ ZZ(68), WW(68), Z(30), W(30), KHIGH, KLOW
      DIMENSION AUX(20), KOUNT(10)
      DIMENSION ANSS(10)
      COMMON /SAVE/RAD(4), PSI(4),QF(4),NORDR(4)
C     REAL*8 WNING
      EXTERNAL WNING
C
C  SUBROUTINE FOR CALCULATING(ALPHA)**-2 TIMES THE INTEGRAL OF
C  K=0. TO K=INFINITY OF..
C  K*GAMMA*SIN(GAMMA*T) *Q24( RIJ, SIIJ, K) * DK
C     WHERE GAMMA = SQRT(K*TANH(K*HT)) .
C  RR = DISTANCE OF FIELD POINT FROM CRAFT.S ORIGIN..  TIME=T.
C  AKU SOMETIMES REPLACES THE INFINITE UPPER LIMIT.
C  ANS = VALUE OF K-INTEGRAL,  R(I,J) AND SI(I,J) IS TRANSMITTED
C  THROUGH THE COMMON BLOCK /SAVE/
C
C  REGION OF INTEGRATION IS DIVIDED INTO TWO PORTIONS.
C  DIVISION POINT IS GIVEN BY *BREAK*
C  SET *KPT* EQUAL TO 1 FOR ROUGH CALCULATIONS.
C  SET *KPT* EQUAL TO 2  FOR REFINED CALCULATIONS.
      N=1
      ANS =  0.
      AKUP=0.
      PERIOD = 2.*PI/(RR+SQRT(HT)*TIME)
      GO TO (1,2), KPT
C
C  BLOCK 1 ****
    1 CONTINUE
C
C  REGION LESS THAN BREAK, FREQUENCY= R+SQRT(HT)*TIME
C  REGION BIGGER THAN BREAK, FREQUENCY = (R+TIME/SQRT(AK))
C
      DO 150 L=1,20
      AKLO= AKUP
      AKUP = AKLO + PERIOD
      IF ( AKLO .GE. BREAK ) GO TO 160
      IF ( AKUP .GT. BREAK ) AKUP= BREAK+ .00001
      KOUNT(1)=6
      CALL  Q66   (AKLO, AKUP, WNING, ANSS(1))
      ANS = ANS+ ANSS(1)
      IF ( NOPRT .NE. 1 ) GO TO 150
      WRITE (6, 1002) L, AKLO, PERIOD, ANSS(1)
 1002 FORMAT ( I20, 2F10.4, F12.6 )
  150 CONTINUE
      GO TO 161
```

```
C
  160 AKUP= BREAK
  161 BNS = 0.
      DO 50  L=1, 65
      AKLO = AKUP
      PERIOD= 2.*PI/(RR+TIME/SQRT(AKLO))
      AKUP= AKLO+PERIOD
      IF ( AKLO .GE. AKU) GO TO 60
      IF ( AKUP .GT. AKU ) AKUP = AKU+ .00001
C
      KOUNT(1) = 4
      CALL QG4(AKLO, AKUP, WNING, ANSS(2))
      BNS = BNS + ANSS(2)
C
      IF ( NOPRT .NE. 1) GO TO 50
      WRITE (6, 1002) L, AKLO, PERIOD, ANSS(2)
   50 CONTINUE
   60  ANS = AL2*(ANS+BNS)
      RETURN
C
C
C  BLOCK 2*****
    2 CONTINUE
      DO 350  L=1,20
      AKLO= AKUP
      AKUP = AKLO + PERIOD
      IF ( AKLO .GE. BREAK ) GO TO 360
      IF ( AKUP .GT. BREAK ) AKUP= BREAK+ .00001
      KOUNT(1) =10
      CALL  QG10  (AKLO, AKUP, WNING, ANSS(1))
      ANS = ANS+ ANSS(1)
      IF ( NOPRT .NE. 1 ) GO TO 350
      WRITE (6, 1002) L, AKLO, PERIOD, ANSS(1)
  350 CONTINUE
      GO TO 361
C
  360 AKUP = BREAK
  361 BNS =0.
      DO 450 L=1, 65
      AKLO = AKUP
      PERIOD= 2.*PI/(RR+TIME/SQRT(AKLO))
      AKUP= AKLO+PERIOD
      IF ( AKLO .GE. AKU) GO TO 460
      IF ( AKUP .GT. AKU ) AKUP = AKU+ .00001
C
      KOUNT(1) = 7
      CALL QG7(AKLO, AKUP, WNING, ANSS(2))
      BNS = BNS + ANSS(2)
C
      IF ( NOPRT .NE. 1) GO TO 450
      WRITE (6, 1002) L, AKLO, PERIOD, ANSS(2)
  450 CONTINUE
  460  ANS = AL2*(ANS+BNS)
C
      RETURN
      END
```

```
      SUBROUTINE FINDKO (AKU, ER, ALPHA)
C  DETERMINE THE APPROX. UPPER LIMIT FOR THE INTEGRATION WITH RESPECT TO K
C  VALUE RETURNED THRU AKU.
      E = ER/( SQRT(ALPHA)*.5079491)
      AKU = 5.
      DAK = 0.5
      DO 10 I=1,30
      IF ((  SQRT(AKU) *EXP(-AKU)) .LE. E)  GO TO 30
   10 AKU = AKU+DAK
   30  AKU = AKU*.636620*ALPHA
C  AKU IS K-SUB. 0.
      WRITE (6,1001) ER, AKU
 1001 FORMAT ( '0   ER=',  E10.3, '  AKU=', F10.3 )
      RETURN
      END
```

```
      FUNCTION WNING( AA )
C  EVALUATES THE INTEGRAND OF THE K-INTEGRAL.
C  NOTE.. THE INPUT- PARAMETERS ARE.. AA, TIME, ALPHA, HT, AND
C                                      R(4) AND SI(4) IN SAVE.
      REAL*4 KERNEL
      REAL*8 Q24
      COMMON/RNPARM/ALPP,ST,DELT,NTS,FLDPT(50,2),NFP,ERR,HT
      COMMON/STATUS/AT,BT,TIME,STATE(50,3),KERNEL(50,50)
      COMMON /SAVE/RAD(4), PSI(4),QF(4),NORDR(4)
      GAMMA = SQRT(AA*TANH(AA*HT))
      WNING = AA*GAMMA* SIN( GAMMA*TIME) * Q24(AA, ALPP)
C  NOTE THAT Q24 TAKES REAL*4 ARGUMENTS.
      RETURN
      END
```

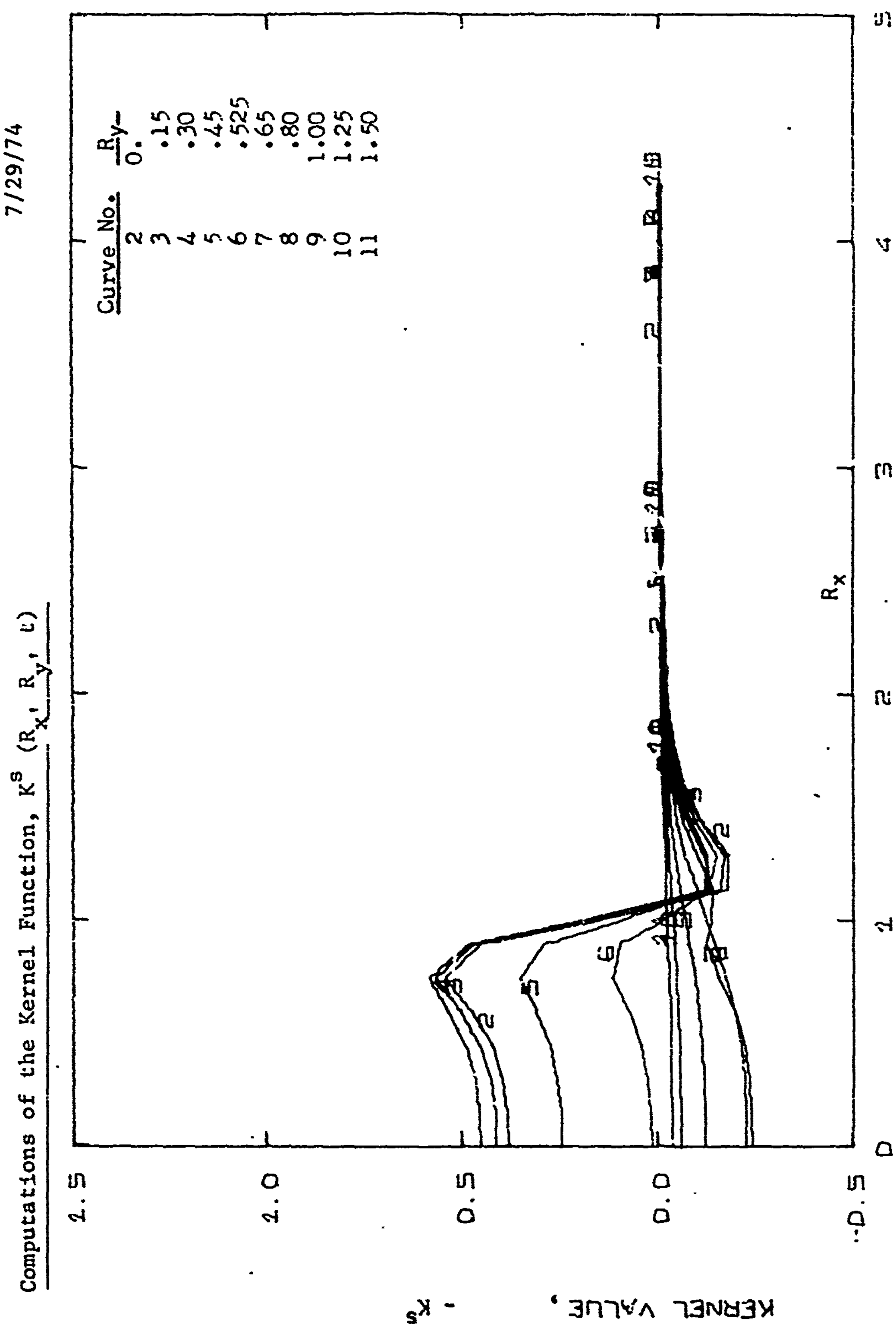


Figure B.1 Kernel-Table for TIME = D.25000

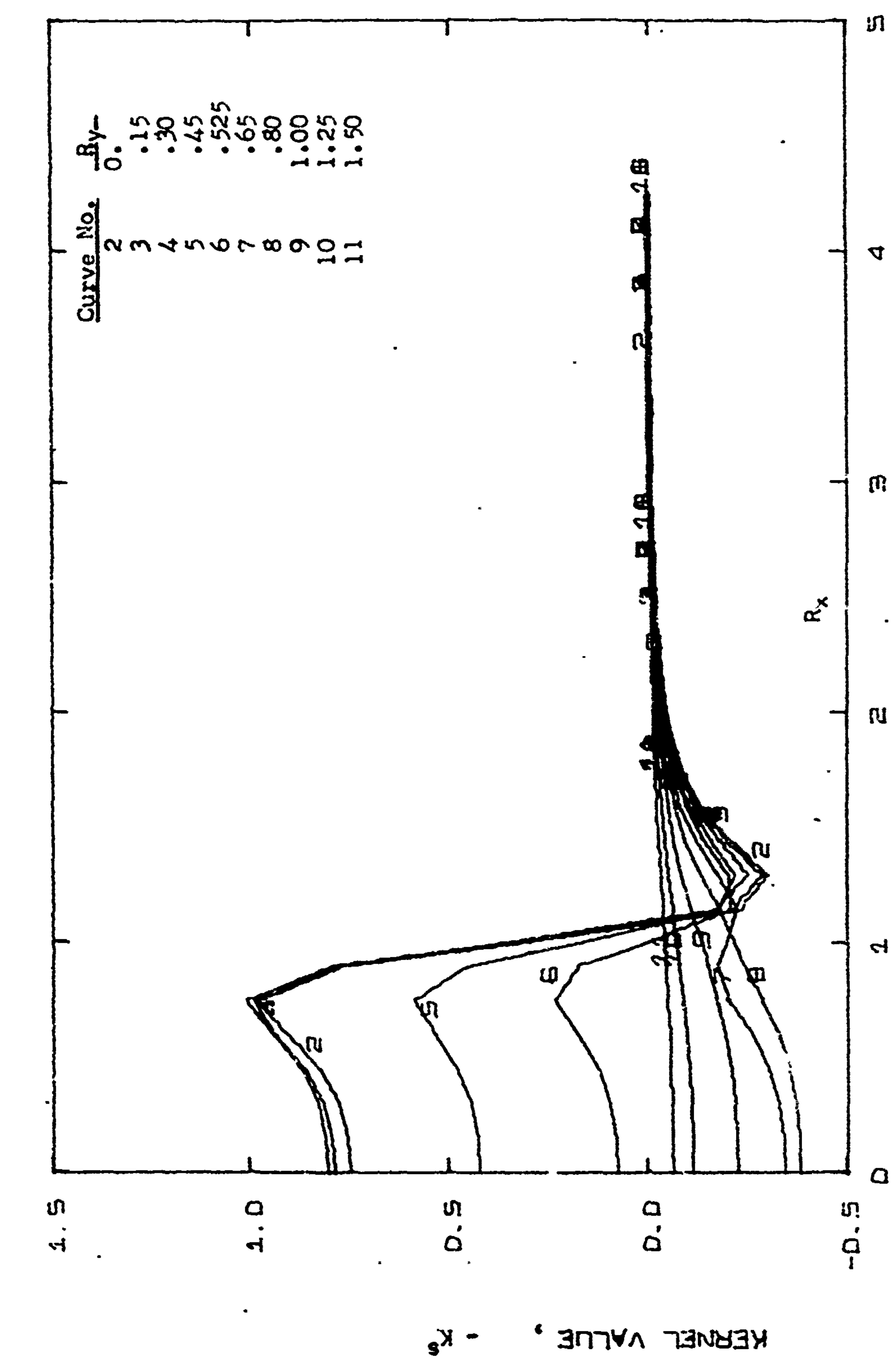


Figure B.2 Kernel-Table for TIME = 0.50000

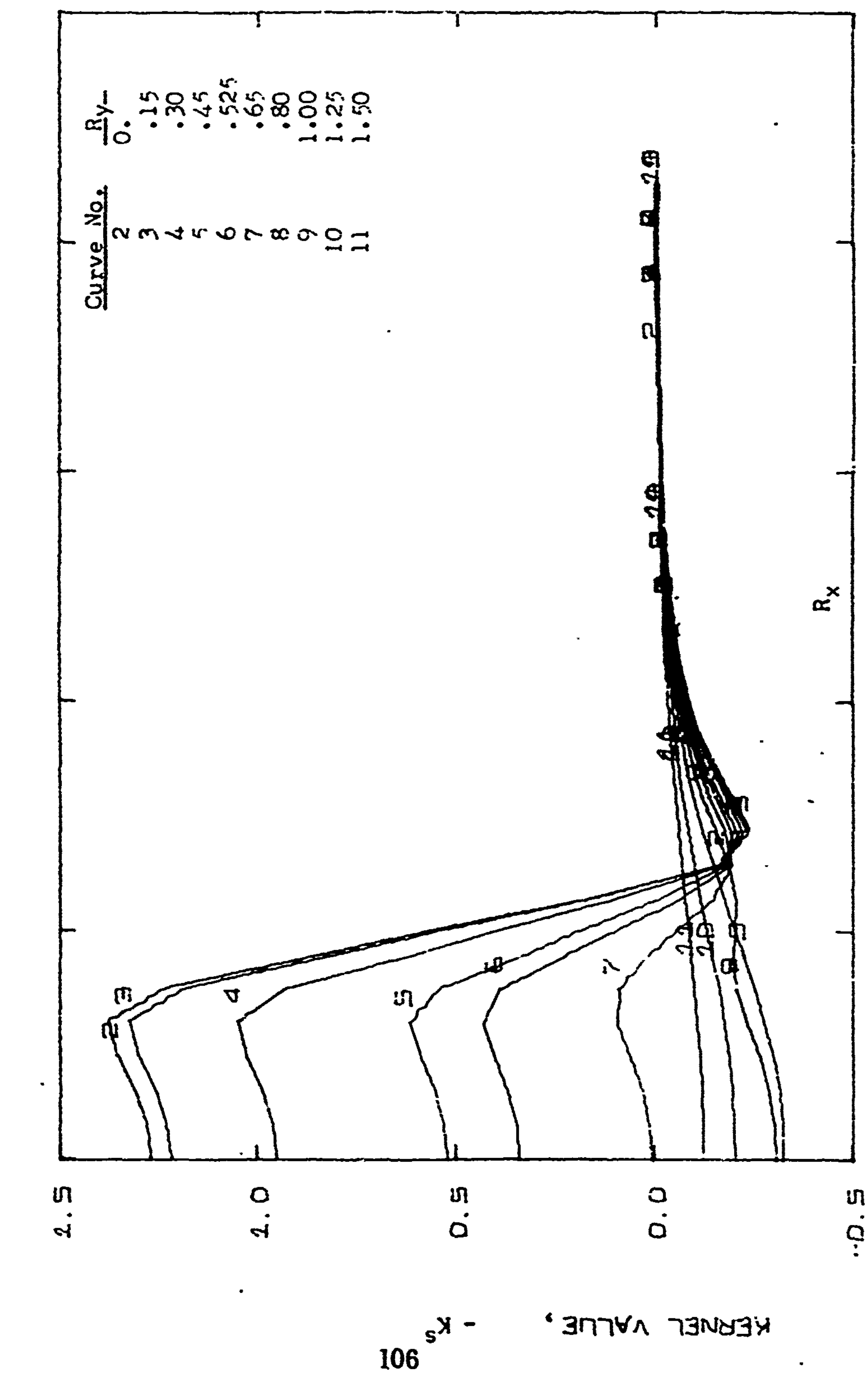


Figure B.3 Kernel-Table for TIME = 1.00000

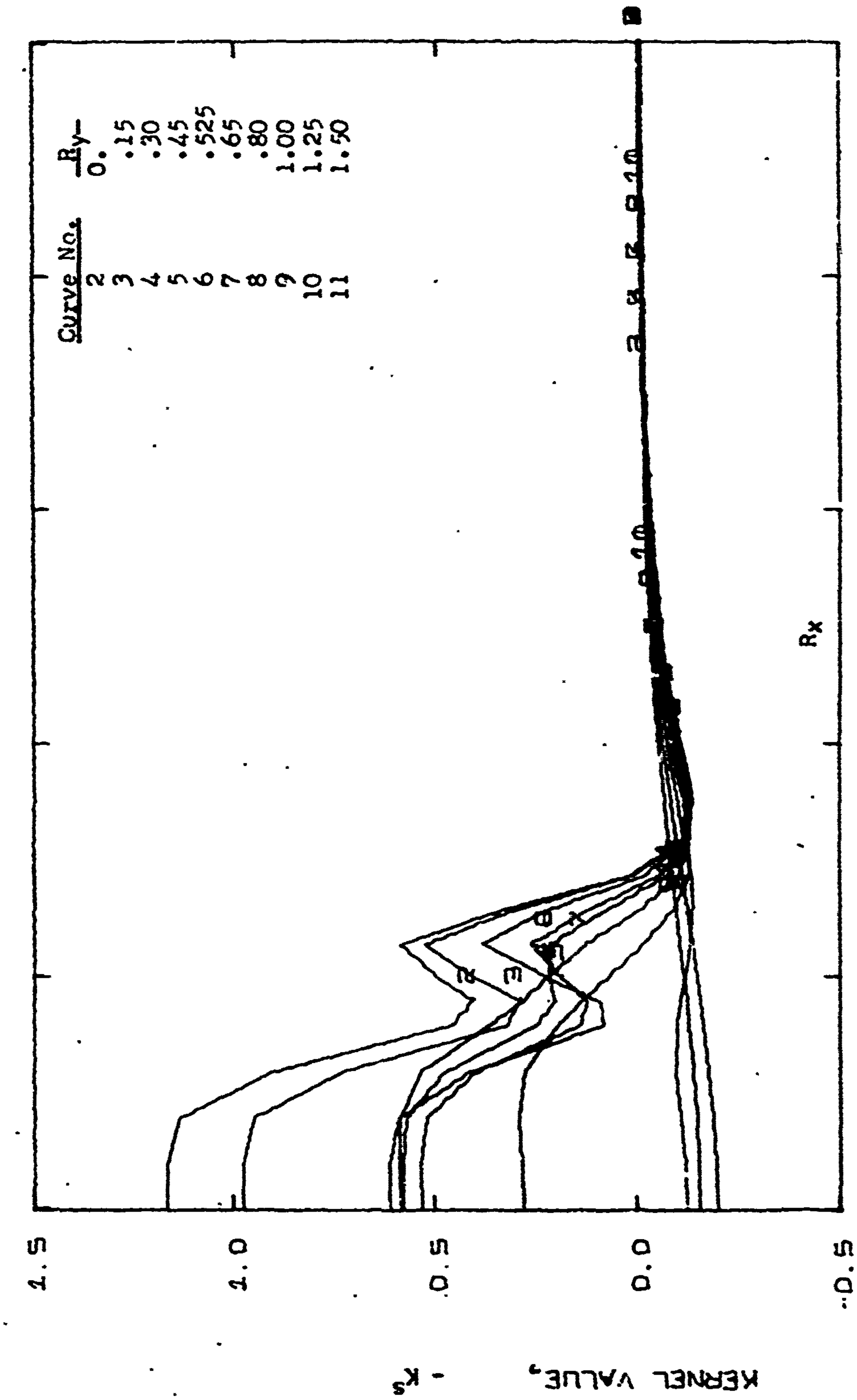


Figure B.4 KERNEL-TABLE FOR TIME = 1.50000

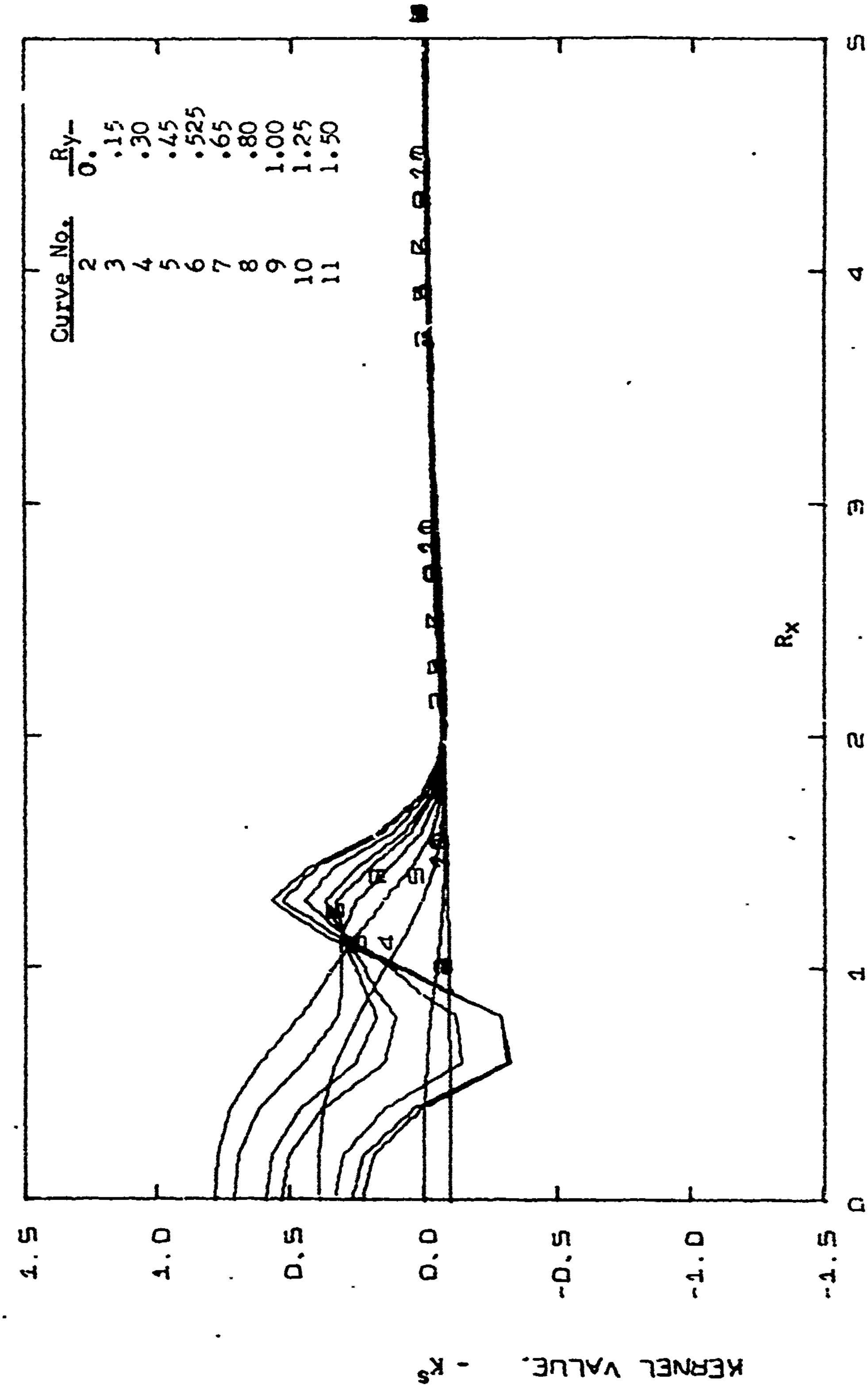


Figure B.5 KERNEL-TABLE FOR TIME = 2.00000

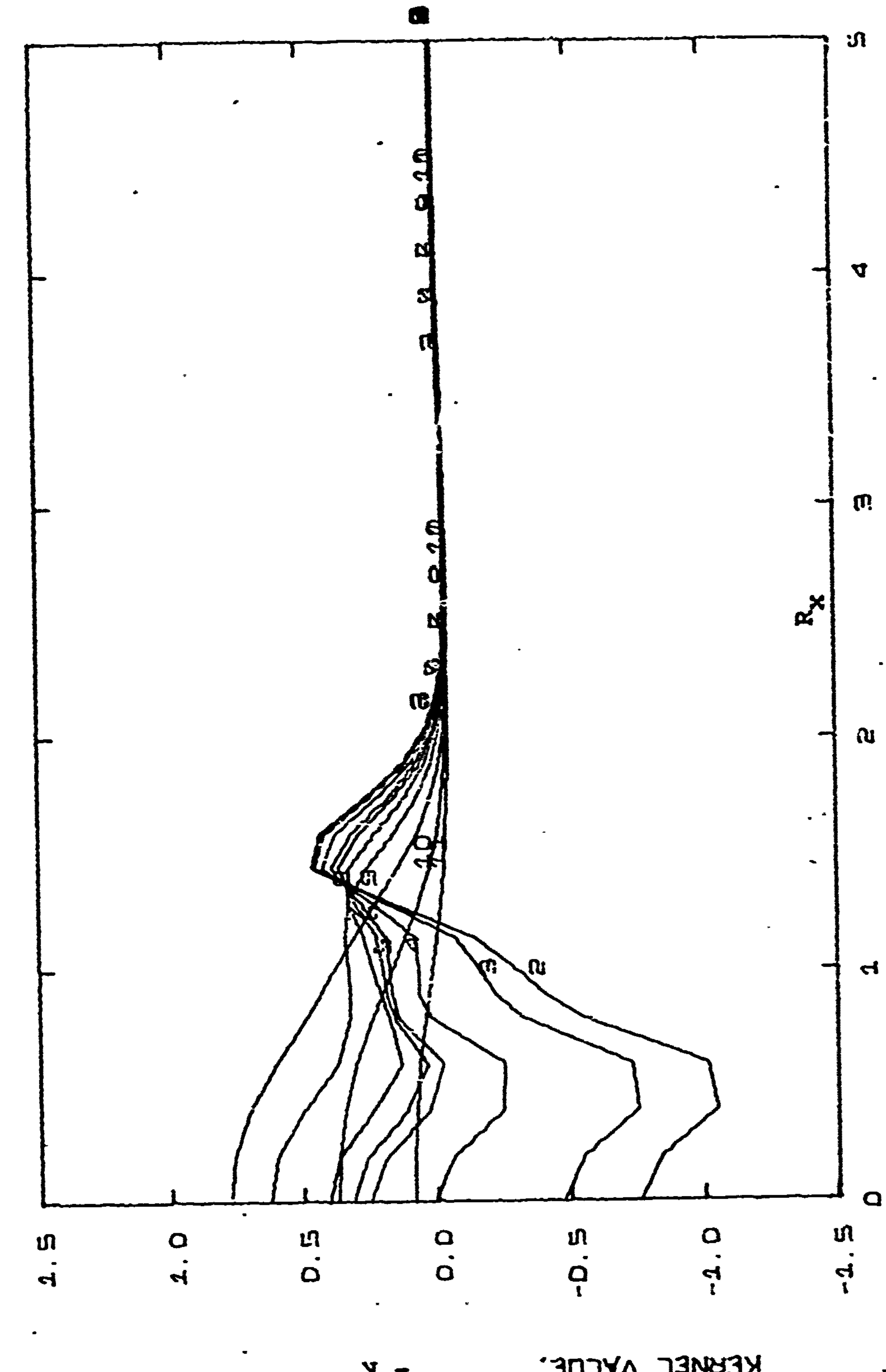


Figure B.6. KERNEL-TABLE FOR TIME = 2.50000

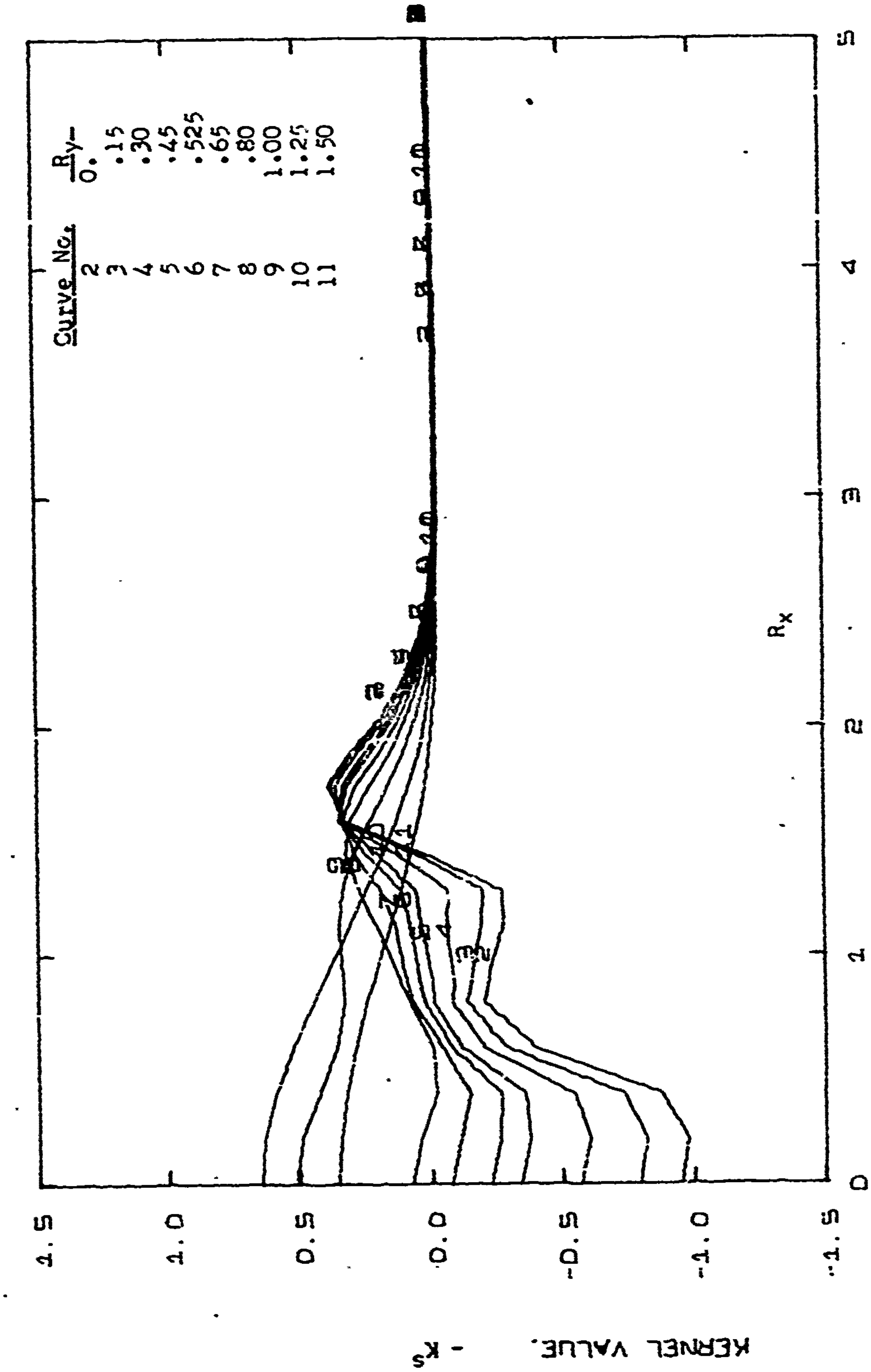


Figure B.7. KERNEL-TABLE FOR TIME = 3.00000

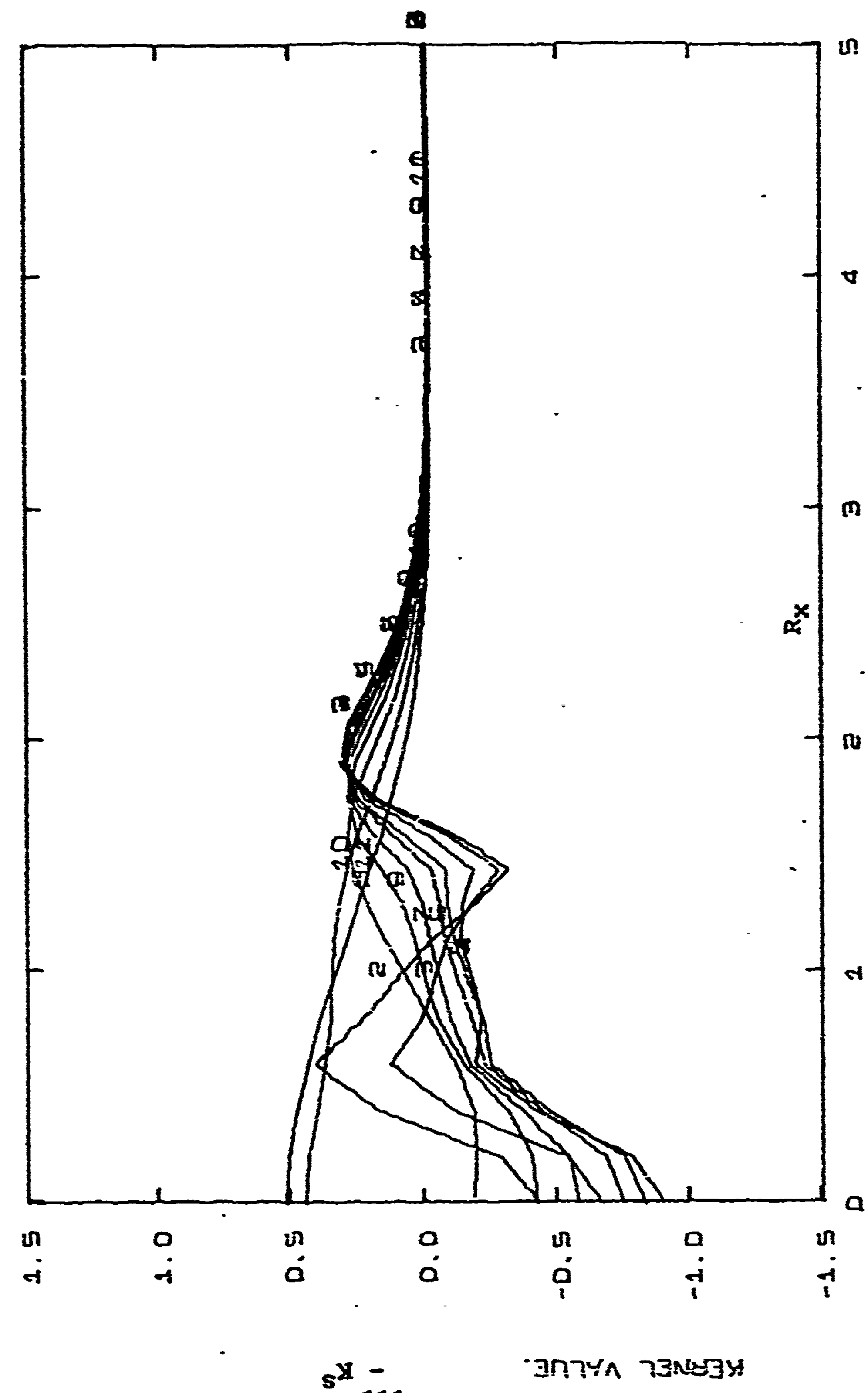


Figure B.8. KERNEL-TABLE FOR TIME = 3.5000

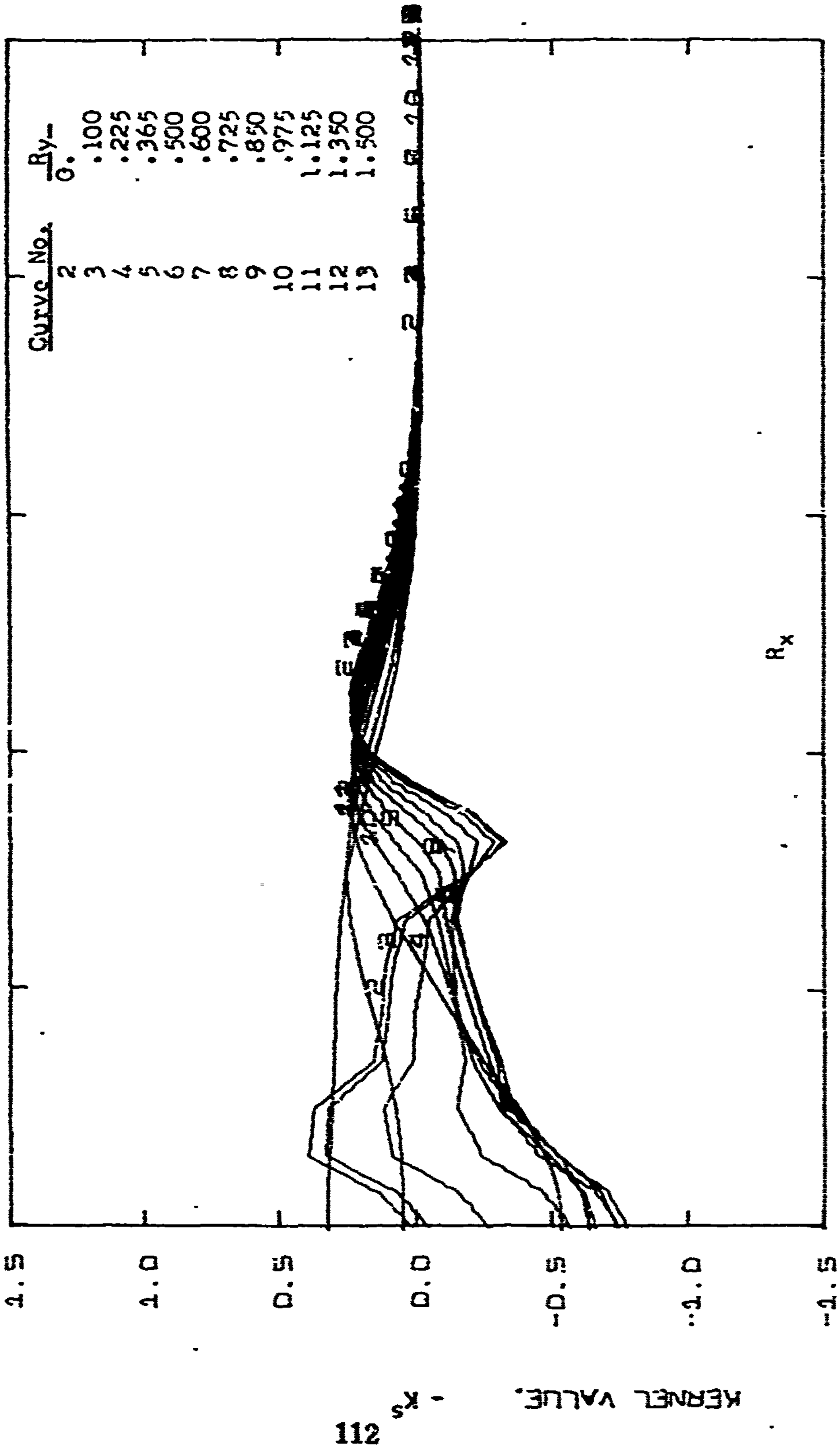


Figure B.9. KERNEL-TABLE FOR TIME = 4.00000

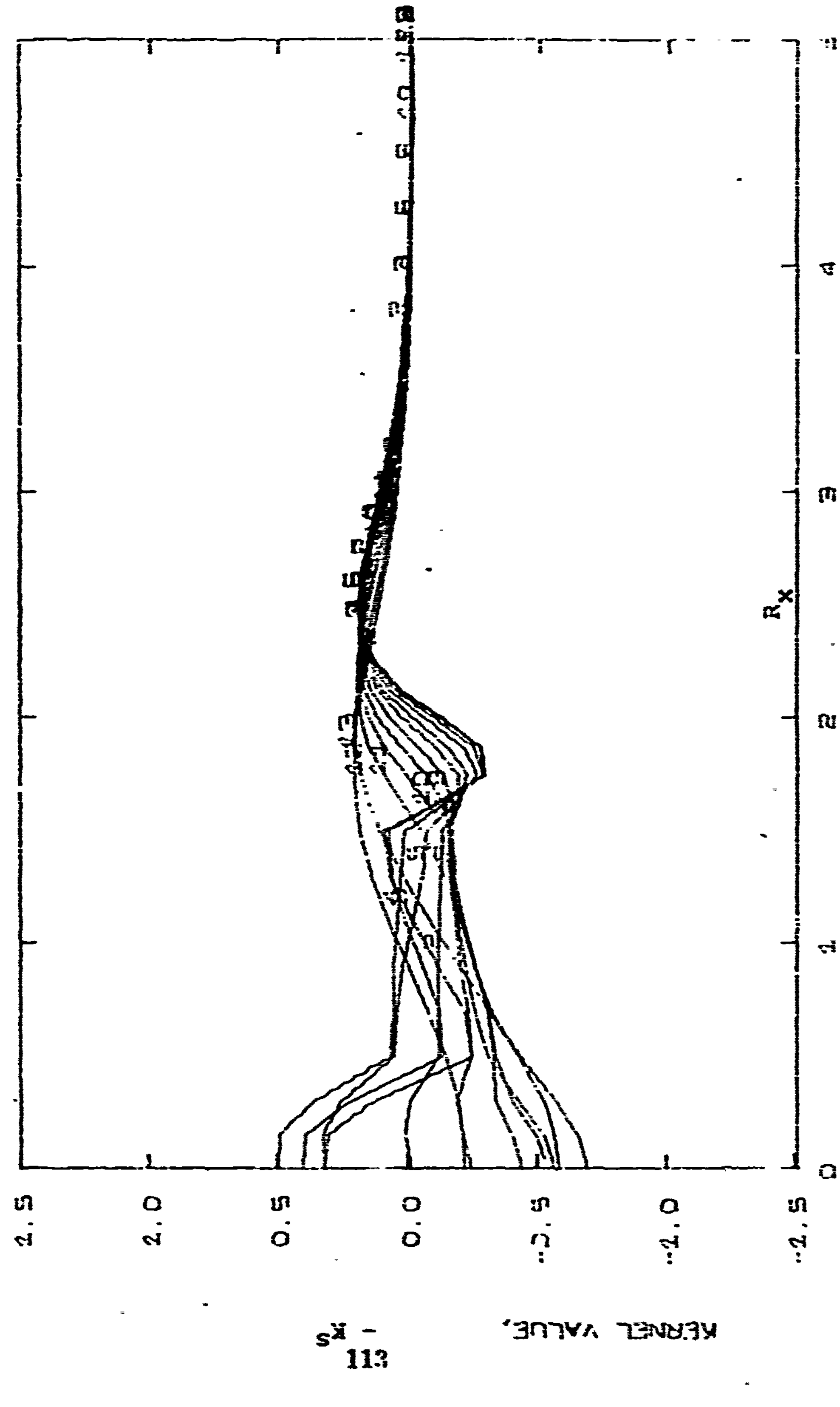


Figure B.10. KERNEL-TABLE FOR TIME = 4.50000

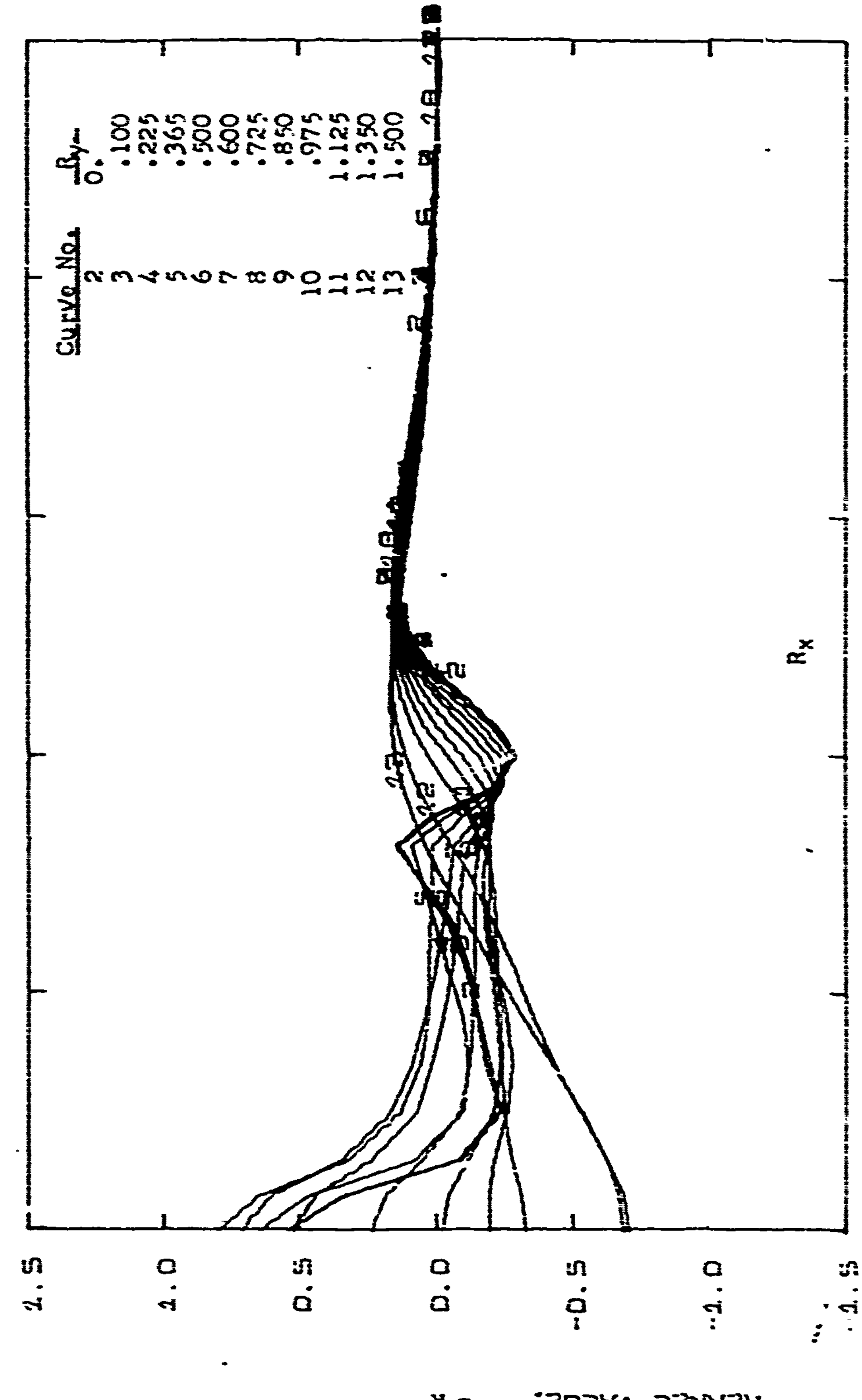


Figure B.11. KERNEL TABLE FOR TIME = 5.00000

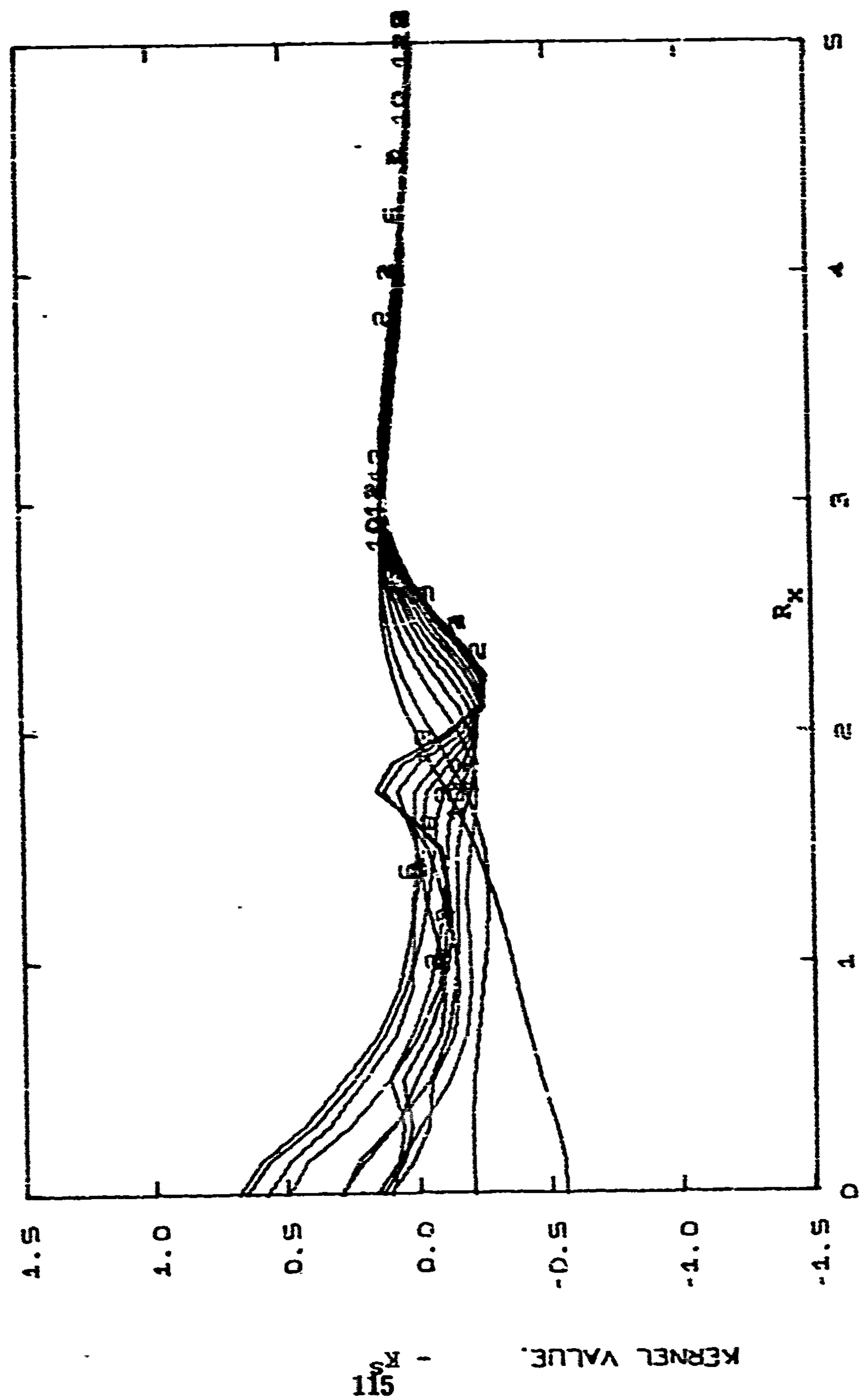


Figure B.12. KERNEL-TABLE FOR TIME = 5.50000

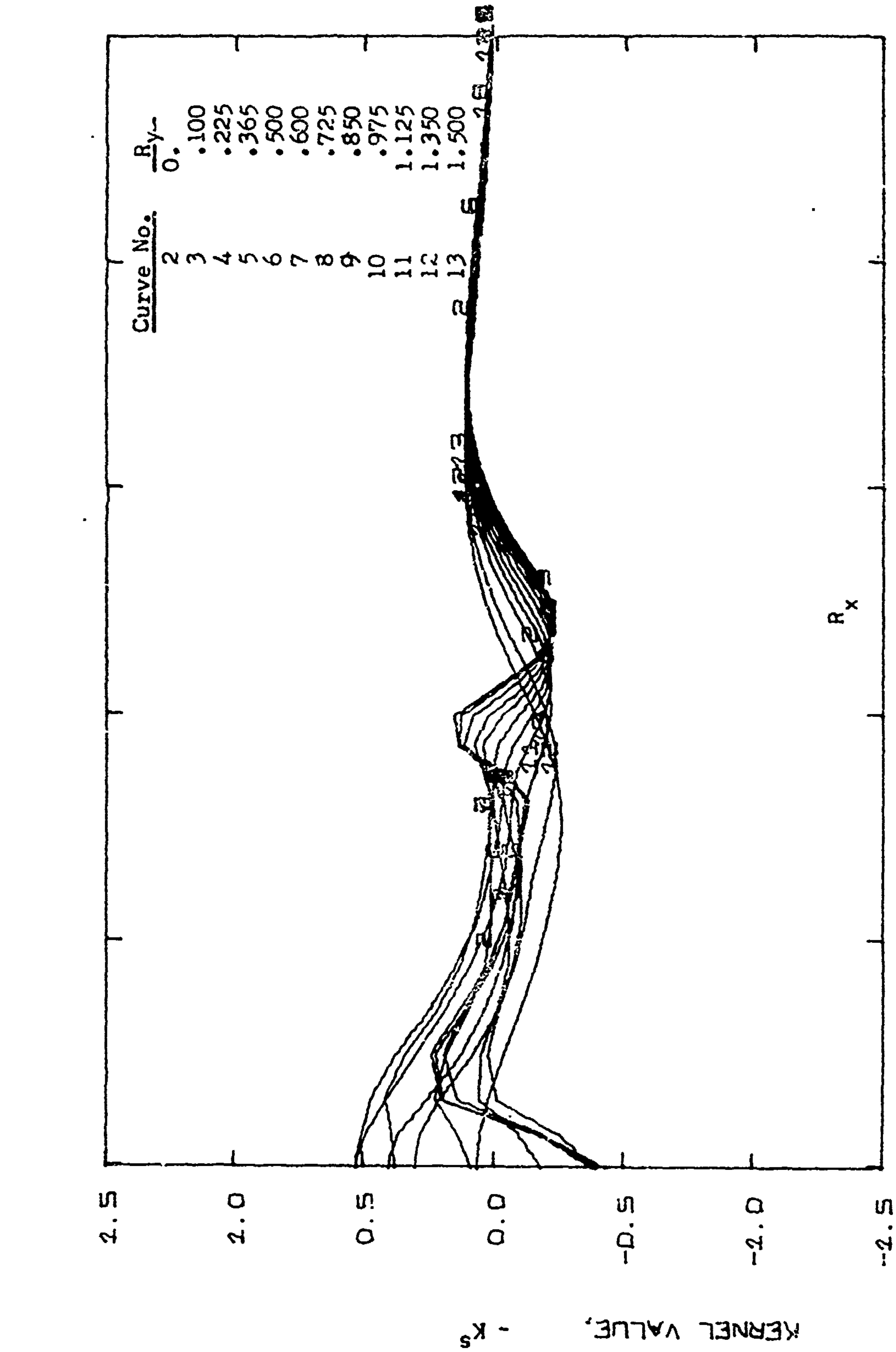


Figure B.13. KERNEL-TABLE FOR TIME = 6.00000

THE THETA-INTEGRAL ROUTINE PACKAGE, Q 24. Documentation is provided here for a package of subroutines that calculates the theta-integral, $Q(k, R, \psi)$. The mathematical basis behind the evaluation of this integral has already been given in the second section of this appendix. Consider equations (136) and (145) and we define:

$$Q24(k, R_x, R_y, \alpha') = \sum_{n=1,3,5\ldots}^{\infty} C_n(k') \left[\sum_{i=1}^{2} \sum_{j=1}^{2} J_{2n}(kR_{ij}) \sin(2n\,\psi_{ij})\right], \tag{153}$$

where

$$C_n(k') = 4 \int_0^{\pi/2} \frac{\sin 2n\theta \quad d\theta}{sh(k'\cos\theta)\, sh(k'\sin\theta)}, \quad k' \neq 0 \tag{154}$$

$$R_{ij} = \left\{[R_x-(-)^i]^2 + [R_y-(-)^j\mu]^2\right\}^{1/2} \tag{155}$$

$$\psi_{ij} = \tan^{-1}\left[\frac{R_y-(-)^j\mu}{R_x-(-)^i}\right] \qquad \left.\begin{matrix} i = 1,2 \\ j = 1,2 \end{matrix}\right\} \tag{156}$$

$$k' = \alpha' k_1 \tag{157}$$

and $\alpha' = \pi/2\alpha$. Equations (153 - 157) have already been given earlier. Here then are repeated for the sake of completeness. Note that Q24 when multiplied by $k\gamma\cdot\sin\gamma\, t$, as can be seen from (136), is the integrand of the k-integral. The physical parameters that need to be specified before (153) can be calculated, are α', the decay factor, and μ, the beam to length ratio. In the function subprogram Q24, the argument list consists of the value k, and the value of α'; the quantities R_{ij} and ψ_{ij} are transferred through a common block. This and other information are detailed in the following paragraphs:

a. Limitations of Parameter values.

$$k' = [.001,\ 125.]$$

$$kR_{ij} = [0.,\ 750.0]$$

Absolute accuracy is approximately 10^{-6}.

b. Contents of Package and Brief Description.

Routine Q24 (AK, ALPP). This routine takes input from the common block/SAVE/(see user-instructions section) and from the argument list. It serves as the calling program for routines GCNK4 and BESSEL as well as sums the function name, Q24.

Routine PREP. This short routine is to be called by the calling program that uses Q24. It should be called once before Q24 is used. Its purpose is simply to load the spline-coefficients data into a common block /COEFG/ so that it can be later used by GCNK4.

Routine BESSEL.. An efficient routine converted from the MATH LIB version. It generates a series of Bessel functions for a given argument.

Routine SESSEL. While BESSEL uses double-precision arithmetic, routine SESSEL which is identical to BESSEL, uses single precision arithmetic.

Routine NBESSL. A function subprogram called by Q24 for estimating the number of terms required in the Bessel series. Equation (141) is used in this case.

Routine GCNK4. This routine utilizes the input data in common block /COEFG/ to obtain values of the coefficients $C_n(k')$ by the method of interpolation.

One Data Deck with a header card, containing values of spline coefficients which will be loaded when routine PREP is called.

c. Instructions for Sample Usage.

C Declarative and other statements in calling program.

```
REAL *8 PI, Q24
COMMON/CNSTAN/PI,P12,MACCY, FACTOR
COMMON/SAVE/R(4),SI(4),QF(4),NORDR(4)
DIMENSION P(4,2)
PI = 3.14159265359
P12 = .5*PI
MACCY = 6
FACTOR = 2
CALL ERRSET (208,999,-1,1)
CALL PREP
```

C Generate R_{ij} and ψ_{ij} values for a given (R_x, R_y) value (ST=Beam/

C length ratio). P(L,M) below are the cartesian components of R_{ij}, ψ_{ij}:

```
P(1,1) = RX + 1.
P(1,2) = RY + ST
P(2,1) = P(1,1)
P(2,2) = RY - ST
P(3,1) = RX - 1.
P(3,2) = P(1,2)
P(4,1) = P(3,1)
P(4,2) = P(2,2)
```

```
C

      DO 120M = 1,4
      R(M) = SQRT (P(M1,)**2 + P(M,2)**2)
  120SI(M) = ATAN2(P(M,2),P(M,1) )

C Define AK and ALPP before calling, and then

      QVALUE = Q24 (AK,ALPP)
```

Note: (1) R_{ij} and ψ_{ij} are transmitted through the COMMON block/SAVE/.

(2) QF(4) and NORDR(4) are values, upon exit, corresponding to $(-)^{i+j}Q(kR_{ij}, \psi_{ij}, k')$ and the number of terms used in the series, respectively.

d. <u>Computation Time.</u> On an IMB 370 machine, each call consumes approximately .01 sec for a value of kR = 5 and may go up to .05 sec for the value of kR = 150.

It is worthwhile to point out that another example of the application of Q24 can be seen in the program TABKS described in the fourth section of this appendix. In that case, the main program calls PREP and sets up the quantities R_{ij} and ψ_{ij}. The function Q24 is then called by the routine WNING.

Listing of the Q24 package is given in the following pages.

```
      DOUBLE PRECISION FUNCTION Q24(TAK,ALPHA)
C  ALPHA HERE IS ALPHA-PRIME IN PROG. REPORT.
C
C  CALCULATE THE THETA-INTEGRAL USING THE BESSEL SERIES EXPANSION
C AND SUMMING THE FOUR SERIES, EACH CORRESPONDING TO A REF. POINT
C  AT THE CORNER OF THE CRAFT.
C  INPUT VALUES OF THE RADIAL DISTANCE, R, AND ANGLE, SI, COMES
C  THRU THE COMMON BLOCK /SAVE/
C   Q24 = SUM I, (I=1,4) OF(-1.)**I * ( SUM N, (N=ODD) OF
C  -1.*J2N(AK*R(I)) * SIN(2N*SI(I)) * CN(AK*ALPHA) )
C  WHERE CN(K) IS OBTAINED BY CALLING THE INTERPOLATION
C  ROUTINE *GCNK4*
C
      REAL*8 TSI, TS, TC, TST, S4, C4, SUMQ, AKR, UNITY(4)
      REAL*8 Y(2000), PI, ASIMPS, AFILON
      DIMENSION DUMMY(400)
      COMMON /SCRTCH/ Y, ND, NF, ASIMPS, AFILON, ACUBNT, ATNH, JJ
      REAL*8 JB(1031), SIGNA
      EQUIVALENCE (JB(1),Y(1)), (DUMMY(1),Y(1032))
      COMMON /CNSTAN/ PI, PI2, MACCY, FACTOR
      COMMON /RESULT/ CK(500)
      COMMON /SAVE/ RR(4), PSI(4), QF(4), NORDR(4)
      DATA UNITY(1), UNITY(2), UNITY(3), UNITY(4) /4.D0,-4.D0,-4.D0,
     1 4.D0/
C
C  FIND MAX. VALUE OF RR IN RR ARRAY.
      AK = TAK
      IF ( AK*ALPHA .LT. .001 ) AK = .001001/ALPHA
      IMKR=1
      AA= RR(1)
      DO 10  I=2,4
      IF ( RR(I) .LE. AA) GO TO 10
      IMKR=I
      AA=RR(I)
   10 CONTINUE
      AKR = AA*AK
      PK = AK*ALPHA
C
C  CHECK THE MAX. VALUE OF AKR.
      IF ( AKR .LT. 750. ) GO TO 11
      WRITE (6, 1001) AK,AA, AKR
 1001 FORMAT ( //, 5X, ' AK=', F10.4,'RMAX=', F10.4, ' AKR=', F10.4, /,
     1 '   AKR TOO LARGE FOR BESSEL FUNCTION ARGUMENT.')
      STOP
C
C  USE MAX. AKR VALUE AS INPUT TO NBESSL TO DETERMINE THE ORDER
C  OF BESSEL FUNCTIONS REQUIRED (HENCE, THE NO. OF TERMS) IN THE
C  SERIES.  THIS WILL THEN BE USED IN THE GENERATION OF THE
C  COEFFICIENT ARRAY CK.
C
C NBSL IS RETURNED TO BE THE ABSOLUTE INDEX OF SERIES, IT IS EVEN.
C  ABSOLUTE INDEX IS THE ORDER OF BESSEL FUNCTION BEYOND WHICH TERMS
C  ARE NEGLIGIBLE.
   11 NORDR(IMKR) = NBESSL(AKR, MACCY)
      NTRMS = (NORDR(IMKR)+2)/4 + 2
      CALL GCNK4 (1, NTRMS, PK)
```

```
C
C  NOTE.. THE COEFFICIENTS *CK* ARE GOOD FOR ALL 4 KR VALUES.
C
      Q24=0.D0
C
C  L-INDEX OF THE LOOP BELOW IS AASSOCIATED WITH EACH CORNER OF CRAFT.
      DO 20 L=1,4
      IF ( ABS (RR(L)) .GT. 1.E-10 ) GO TO 21
      SUMQ=0.
      GO TO 35
   21 AKR=AK*RR(L)
      SI = PSI(L)
      IF ( L .NE. IMKR)    NORDR(L) = NBESSL(AKR, MACCY)
C
C  GENERATE VALUES OF BESSEL FUNCTIONS IN SERIES.
      IF ( AKR -240.) 45, 46, 46
   45 NMAX = 400
      SKR = AKR
      CALL SESSEL(SKR,NMAX,JB,-1.,IERROR,1.E-50,DUMMY)
      GO TO 47
C
C  SINGLE PRECISION BESSEL FUNCTION FOR ARGUMENT LESS THAN 250.
   46 NMAX = 1031
      CALL BESSEL (AKR, NMAX, JB, -1.D0, IERROR, 1.E-50)
C
C  SUM BESSEL SERIES
C  J(N) IS STORED IN JB(N+1)
   47 NN = NORDR(L)/2
C
      TSI = 2.D0*SI
      TS = DSIN (TSI)
      TC = DCOS( TSI)
      S4 = 2.D0*TS*TC
      C4 =  1.D0 - 2.D0*(TS**2)
      SUMQ = JB(3) * CK  (1)*TS
      J=1
C
C  LOOP INDEX IN SUMMING OF SERIES IS ODD IN VALUE, STEP=2
      DO 30  N=3,  NN, 2
C      CALCULATE SIN(2N*SI)
      TST = TS
      TS = TS*C4 + TC*S4
      TC = TC*C4 - TST*S4
      J = J+1
   30 SUMQ = SUMQ + JB(2*N+1)*TS*CK(J)
C
   35 SUMQ = SUMQ*UNITY(L)
      QF(L) = SUMQ
      Q24 = Q24 + SUMQ
   20 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PREP
C
C  READS IN SPLINE COEFFICIENTS FROM DATA CARDS.
      COMMON /COEFG/ NPK, NPKM1, NCCAL, PKSP(100),CSP(4,100,10)
C
      READ (5,1001) NPK, NCCAL
      NPKM1 = NPK-1
      READ(5,1002)  (PKSP(K), K=1, NPK)
      READ (5,1003) (((CSP(J,K,N), J=1,4), K=1, NPKM1), N=1, NCCAL)
 1001 FORMAT (16I5)
 1002 FORMAT (6X, 6E11.4)
 1003 FORMAT (6X, 4E16.7)
      RETURN
      END
```

```
      FUNCTION NBESSL ( AKR, M)
      REAL*8 AKR
C
C  THIS FUNCTION DETERMINES THE NO. OF TERMS REQUIRED IN THE BESSL SERIES FOR
C GIVEN ACCURACY OF 10**(-M), AND OF ARGUMENT AKR.
C  MUST SOLVE FOR N SUCH THAT ( ALOG10(N) > ALOG10(F) +M/N
C
      IF ( AKR .GT. 1.E-5 ) GO TO 10
      NBESSL=10
      RETURN
   10 F = 2.7182*AKR*.5
      ALF = ALOG10(F)
      AM2 = M
C
      NS= 1.36*AKR
      IF ( NS.LT. 10) NS=10
      DO 20 N=NS, 1000, 4
      NBESSL = N
      IF( ALOG10(FLOAT(N))*(1.+.5/N) .GT. (ALF+AM2/N)) GO TO 30
   20 CONTINUE
      WRITE (6, 1001)
 1001 FORMAT (/, '  WATCH - OUT !, REQUIRES NBESSL >1000, BUT SET=1000')
   30 CONTINUE
      NS= NBESSL/2
      IF ( NS*2 .NE. NBESSL) NBESSL=NBESSL-1
      RETURN
      END
```

```
      SUBROUTINE GCNK4 (N1, NTRMS,PPK)
C  THIS SUBROUTINE GENERATES THE ARRAY CNK(PK) AND STORES IT IN CK(N),
C  HERE N ACTUALLY TAKES ON THE VALUE OF N=1,3,5,7 ... ETC, EVEN THE
C  STORAGE IS IN SEQUENCE.
C  VALUES OF CK(J), J=N1, NTRM, ARE CALCULATED FROM THE
C  EXISTING SPLINE COEFFICIENTS STORED IN THE ARRY CSP.
      REAL*8 PI
      COMMON /CNSTAN/PI, PI2, MACCY, FACTOR
      COMMON /COEFG/ NPK, NPKM1, NCCAL, PKSP(100),CSP(4,100,10)
      COMMON /RESULT/ CK(500)
C
      PK = PPK
      IF ( PK .LT. PKSP(1)) GO TO 100
      IF ( PK .LE. PKSP(NPK)) GO TO 200
  100 CONTINUE
C 100 WRITE (6,1001) PK, PKSP(NPK), PKSP (1)
 1001 FORMAT (//, '  WARNING REQUESTED VALUES OF PK IS OUT OF THE RANGE
     1OF PK AVAILABLE, ', /, '  EXPRAPOLATION IS USED,  PK=', E10.5,
     2 'PKSP(NPK)=',E10.5,'PKSP(1)=', E10.5 )
C
C  SEARCH FOR LOCATION OF PK IN PKSP ARRAY.
  200 DO 10  J=1, NPKM1
      JJ = J
      IF ( PK .GT. PKSP(J+1)) GO TO 10
      GO TO 20
   10 CONTINUE
C
   20 NN = NTRMS
      IF ( NN .GT. NCCAL)  NN = NCCAL
      H1 = PK-PKSP(JJ)
      H2 = H1**2
      H3 = H2*H1
      PKSHP = PK*SINH(PK)
C
      DO 30  N=N1, NN
C  INTERPOLATE CNK FROM TABLE FOR RANGE OF N SPECIFIED.
      CCC    = CSP(1,JJ,N)*H3 + CSP(2,JJ,N)*H2 + CSP(3,JJ,N)*H1 +
     1 CSP(4,JJ,N)
   30 CK(N) = EXP(CCC)/ PKSHP
C
C
      IF ( NTRMS .LT. NCCAL ) RETURN
      NN = NCCAL+1
      DO 40  N=NN, NTRMS
   40 CK(N) = PI2*DTANH(N*PI/PPK) / PKSHP
      RETURN
      END
```

```
      SUBROUTINE BESSEL(X,N,ARRAY,SIGNA,IERROR, ALOW)
C
C        PURPOSE
C              TO COMPUTE ALL ORDERS OF THE BESSEL FUNCTIONS J(M,X) OR
C              E**(-X)*I(M,X) FOR A GIVEN ARGUMENT X.
C
C        DESCRIPTION OF PARAMETERS
C              X = THE REAL*8 ARGUMENT
C              N = THE INTEGER*4 MAXIMUM DIMENSION OF ARRAY IN WHICH THE
C                  RESULTS ARE TO BE PLACED (ON INPUT)
C              N = THE INTEGER*4 INDEX OF THE LAST ELEMENT OF ARRAY WHICH
C                  CONTAINS A NON-ZERO RESULT. (ON OUTPUT)
C              ARRAY = THE REAL*8 ONE-DIMENSIONAL ARRAY OF LENGTH N IN
C                      WHICH THE RESULTS ARE TO BE PLACED.  AFTER
C                      EXECUTION OF THIS SUBPROGRAM. THE MTH ELEMENT OF
C                      ARRAY CONTAINS J(M-1,X) OR E**(-X)*I(M-1,X).
C               SIGNA = THE REAL*8 INDICATION SPECIFYING WHICH KIND OF
C                       BESSEL FUNCTIONS ARE TO BE COMPUTED.  SET SIGNA
C                       = -1 FOR J(M,X) OR SIGNA = 1 FOR E**(-X)*I(M,X).
C              IERROR= AN ERROR INDICATOR RETURNED
C                IERROR = 0 MEANS NO ERROR
C                IERROR = 1 MEANS ARGUMENT TOO SMALL, ABS(X) .LE. 10**-38
C                IERROR = 2 MEANS ARGUMENT TOO LARGE, ABS(X) .GT. 250
C                FOR J(M,X) OR ABS(X) .GT. 350 FOR E**(-X)*I(M,X)
C               ALOW = REAL*4 VARIABLE FOR ERROR CONTROL.  FOR BEST
C                      ACCURACY, USE 1.E-78.
C
C        DESCRIPTION OF PROGRAM
C              WHEN ABS(X) .LT. 0.1, THE BESSEL FUNCTIONS ARE COMPUTED
C              USING THEIR POWER SERIES EXPANSIONS.(SEE THE HANDBOOK OF
C              MATHEMATICAL FUNCTIONS.)
C              WHEN ABS(X) .GE. 0.1, THE BESSEL FUNCTIONS ARE COMPUTED
C              USING A RECURSION RELATION METHOD.  (SEE "GENERATION OF
C              BESSEL FUNCTIONS ON HIGH SPEED COMPUTERS")
C
      REAL*8 ARRAY(1), ZUM, SIGNA, X
      DATA            F2/1.359141/,BLOW/1.E-38/
      DATA IFLG/0/, NMAX/1030/,BREAK/.1/
      IF(IFLG.NE.0) GO TO 98
      DLOW=ALOG(1.E14*ALOW)
      IFLG=1
   98 IERROR=0
      NM1 =N-1
      SIGNT=1.
      IF(X.LT.0) SIGNT=-SIGNT
      X=DABS(X)
      XX = X
      DO 99 I=1,N
   99 ARRAY(I)=0.D0
      IF(XX.GE.BREAK) GO TO 200
C
C  SERIES METHOD - USED FOR X SMALLER THAN .1
C
      IF ( XX .GT.BLOW) GO TO 101
      ARRAY(1)=1.D0
      N=1
```

```
      IERROR=1
      RETURN
 101  CLOW=10.*ALOW/X
      EXPX=1.
      IF(SIGNA .GT.0.D0)EXPX=EXP(XX)
      I=1
      FMULT=X*X*SIGNA/4.
      E=1.
      GO TO 103
 102  E=E*(X/2.)/FLOAT(I-1)
 103  SUM=1.
      ANPLUS=FLOAT(I)
      AN=1.
      TERM=SUM
 104  SUME=SUM
      TERM=TERM*FMULT/(ANPLUS*AN)
      SUM=SUM+TERM
      ANPLUS=ANPLUS+1.
      AN=AN+1.
      IF(ABS(SUME-SUM) .GT. 1.E-70) GO TO 104
      ARRAY(I)=E*SUM/EXPX
      I=I+1
      IF(I .LE. N .AND.E .GT. CLOW) GO TO 102
      N=I-1
      GO TO 205
 150  IERROR=2
      N=0
      RETURN
C
C  RECURSION RELATION MEHTOD - USED FOR X LARGER THAN .1
C
  200 CLXE = ALOG(XX*F2)
      IF (SIGNA .LT. 0.D0) GO TO 207
      IF (XX.GT. 750.) GO TO 150
      CLOW=DLOW+X
      I=IFIX(XX)
      GO TO 201
 207  IF (XX.GT. 750.) GO TO 150
      CLOW=DLOW
      I = IFIX(1.38*XX)
 201  I=I+3
      AN=FLOAT(I)
      IF ((-ALOG(AN)*(.5+AN)+AN*CLXE) .GT. CLOW .AND.(I.LT.NM1))GOTO 201
      IF(I .GT.NMAX) GO TO 150
      N=I
CC    SS=-ALOG(AN)*(.5+AN) + AN*CLXE
C     WRITE(6, 1001)  SS,CLOW, CLXE
C1001 FORMAT ( 1H0, '  CHECK- ',3E15.6)
      ARRAY(I)=1.D-64
      DO 202 I=2,N
      J=N-I+1
 202  ARRAY(J)=ARRAY(J+2)*SIGNA+2.D0*DFLOAT(J)*ARRAY(J+1)/X
      ZUM=ARRAY(1)
      J=1
      IF(SIGNA .LT.0.D0)J=2
      JLOW=J+1
      DO 203 I=JLOW,N,J
  203 ZUM= ZUM+2.D0*ARRAY(I)
      DO 204 I=1,N
 204  ARRAY(I)=ARRAY(I)/ZUM
```

```
205   IF(SIGNT .GT.0) RETURN
      DO 206 I=2,N,2
206   ARRAY(I)=-ARRAY(I)
      RETURN
      END
```

```
      SUBROUTINE SESSEL(X,N,ARRAY,SIGNA,IERROR, ALOW, BRRAY)
C
C  ROUTINE SIMILAR TO *BESSEL*, EXCEPT..
C  SINGLE PRECISION ARITHMETIC IS USED.
C   *BRRAY* IS USED IN COMPUTATIONS.  *ARRAY* RETURNS THE RESULTS.
C   NOTE.  SIGNA IS REAL*4,  X IS ALSO REAL*4, WHILE *ARRAY* IS REAL*8
C
      REAL*8 ARRAY(1)
      DIMENSION BRRAY(1)
      DATA              F2/1.359141/,BLOW/1.E-38/
      DATA IFLG/0/, NMAX/ 400/,BREAK/.1/
      IF(IFLG.NE.0) GO TO 98
      DLOW=ALOG(1.E14*ALOW)
      IFLG=1
 98   IERROR=0
      NM1 =N-1
      SIGNT=1.
      IF(X.LT.0) SIGNT=-SIGNT
      X= ABS(X)
      XX = X
      DO 99 I=1,N
   99 BRRAY(I)=0.
      IF(XX.GE.BREAK) GO TO 200
C
C  SERIES METHOD - USED FOR X SMALLER THAN .1
C
      IF ( XX .GT.BLOW) GO TO 101
      ARRAY(1)=1.D0
      N=1
      IERROR=1
      RETURN
 101  CLOW=10.*ALOW/X
      EXPX=1.
      IF(SIGNA .GT.0.  )EXPX=EXP(XX)
      I=1
      FMULT=X*X*SIGNA/4.
      E=1.
      GO TO 103
 102  E=E*(X/2.)/FLOAT(I-1)
 103  SUM=1.
      ANPLUS=FLOAT(I)
      AN=1.
      TERM=SUM
 104  SUME=SUM
      TERM=TERM*FMULT/(ANPLUS*AN)
      SUM=SUM+TERM
      ANPLUS=ANPLUS+1.
      AN=AN+1.
      IF(ABS(SUME-SUM) .GT. 1.E-70) GO TO 104
      ARRAY(I)=E*SUM/EXPX
      I=I+1
      IF(I .LE. N .AND.E .GT. CLOW) GO TO 102
      N=I-1
      GO TO 205
 150  IERROR=2
      WRITE (6,1001)  X,N
```

```
 1001 FORMAT (  '  STOPPED IN SESSEL.  X'. F10.4, ' NREOD ='.I2.
     1  '  .GT. NMAX' )
      N=0
      STOP
C
C  RECURSION RELATION MEHTOD - USED FOR X LARGER THAN .1
C
  200 CLXE = ALOG(XX*F2)
      IF (SIGNA .LT. 0.   ) GO TO 207
      IF (XX.GT. 350.) GO TO 150
      CLOW=DLOW+X
      I=IFIX(XX)
      GO TO 201
 207  IF (XX.GT. 250.) GO TO 150
      CLOW=DLOW
      I = IFIX(1.28*XX)
 201  I=I+3
      AN=FLOAT(I)
      IF ((-ALOG(AN)*(.5+AN)+AN*CLXE) .GT. CLOW .AND.(I.LT.NM1))GOTO 201
      IF(I .GT.NMAX) GO TO 150
      N=I
      BRRAY(I) = 1.E-64
      BRRAY(I+1) = 0.
      DO 202 I=2.N
      J=N-I+1
 202  BRRAY(J)=BRRAY(J+2)*SIGNA+  2.* FLOAT(J)*BRRAY(J+1)/X
      ZUM = BRRAY(1)
      J=1
      IF(SIGNA .LT.0.   )J=2
      JLOW=J+1
      DO 203 I=JLOW.N.J
  203 ZUM= ZUM+2.   *BRRAY(I)
      DO 204 I=1.N
 204  ARRAY(I)=BRRAY(I)/ZUM
 205  IF(SIGNT .GT.0) RETURN
      DO 206 I=2.N.2
 206  ARRAY(I)=-BRRAY(I)
      RETURN
      END
```