

WDRA3D and M_GEN

3D Wave Radiation and Diffraction Analysis
&
Automatic Mesh Generation
User Manual

J.A. Hamilton

Contents

Introduction	3
I 3D Quadratic Boundary Element Method for Linear Diffraction & Radiation Analysis	4
1 Purpose	4
2 Theoretical Background	4
3 Numerical Features	5
4 Overall Program Description	6
5 Input Data Preparation	7
6 Output File	9
7 Program Execution	10
8 Sample input file	10
II Automatic Mesh Generation	13
1 Program m_gen	13
1.1 Program Description	13
1.2 Input File Format	14
1.3 Sample Input File	16
1.4 Program Execution and Memory Requirements	17
2 Secondary Programs	19
2.1 Hemisphere Geometries	19
2.1.1 Program Description	19
2.1.2 Input File Format	19
2.1.3 Sample Input File	20
2.1.4 Program Execution and Memory Requirements	21
2.2 Truncated Cylinder Geometries	22
2.2.1 Program Description	22
2.2.2 Input File Format	22
2.2.3 Sample Input File	23
2.2.4 Program Execution and Memory Requirements	24
2.3 Tension Leg Platform Structures	25
2.3.1 Program Description	25
2.3.2 Input File Format	25

2.3.3	Sample Input File	26
2.3.4	Program Execution and Memory Requirements	27
2.4	Ship Geometries	28
2.4.1	Program Description	28
2.4.2	Input File Format	28
2.4.3	Sample Input File	29
2.4.4	Program Execution and Memory Requirements	31
3	Other strucures	32
3.1	Construction of input files	32
3.2	Useful subroutines	32
3.2.1	Subroutine mirror_x	32
3.2.2	Subroutine mirror_y	33
3.2.3	Subroutine write	33
4	Mesh Verification and Visualization	34
4.1	Program meshver	34
4.1.1	Program Description	34
4.1.2	Program execution	34
4.2	Mesh Plotting	34
4.2.1	Plotting program	34
4.2.2	Program Description	35
4.3	Plotting with Gnuplot	35
III	Post Processing of Results	36
1	Program plot_gnu	36
1.1	Program Description	36
1.2	Program Input	36
2	Plotting with Gnuplot	36
A	Compilation notes on all programs	37
A.1	wdra3d	37
A.2	m_gen	37
A.3	hemisphere	37
A.4	t_cyl	37
A.5	tlp	37
A.6	ship	38
A.7	meshver	38
A.8	plot	38
A.9	plot_gnu	38

Introduction

This report documents the usage of the WDRA3D program for solution of a linearized wave radiation and diffraction problem, and the usage of the automatic mesh generation programs for the creation of the input files required by the WDRA3D program. Specific emphasis is placed on outlining the exact input requirements and output format of the programs.

Part one describes the basic features of the WDRA3D program. The quantities output by the program are defined and the format of the output file outlined. Because the WDRA3D program requires a discretized form of the body to be studied, the input file is prohibitively tedious to create by hand. The `m_gen` program described in part two was developed to automatically create the necessary input file. These programs create an appropriate input file for the WDRA3D program based on specified dimensions of a specific geometry (ie. radius of a hemisphere). Part three describes some utilities for post processing the results of the WDRA3D program. These programs convert the WDRA3D output file into a form more convenient for plotting specific results with existing plotting software.

Part I

3D Quadratic Boundary Element Method for Linear Diffraction & Radiation Analysis

- Title: WDRA3D (Wave Diffraction & Radiation Analysis for 3D Bodies)
- Author: F.P. Chau
- Source Language: FORTRAN 77
- Machine: IBM 6000

1 Purpose

WDRA3D is a computer code which has been developed to analyse the linearized wave diffraction and radiation problem around three dimensional structures. The output contains the velocity potential (local pressure) which is integrated over the body surface to generate the wave exciting force in the diffraction problem and a matrix of hydrodynamic coefficients (added mass and damping) in the radiation problem.

2 Theoretical Background

The mathematical formulation is standard, but is briefly reviewed here in order to provide a basis for discussion of the definition of the coordinate system and the input/output formats.

The problem is solved in the context of linear potential theory such that the flow field can be described by a scalar velocity potential. A righthand space-fixed coordinate system O_{xyz} is chosen such that the origin is located on the undisturbed free surface with the z axis pointing vertically upwards. The incident wave makes an angle β with the positive x -axis. Because the problem is linear, and the flow is harmonic in time with angular frequency ω , the dependence on time may be seperated. The velocity potential may be expressed as

$$\Phi(x, y, z, t) = Re[\phi(x, y, z)e^{-i\omega t}] \quad (1)$$

$\phi(x, y, z)$ may be further decomposed into

$$\phi = \phi_0 + \phi_7 + \sum_{j=1}^6 \xi_j \phi_j \quad (2)$$

where the complex motion amplitude $\xi_j (j = 1, 2, \dots, 6)$ corresponds to surge, sway, heave, roll, pitch, and yaw respectively. ϕ_0 , ϕ_7 , and ϕ_j denote the incident, diffraction, and radiation potentials respectively. The incident potential, ϕ_0 , is defined as follows.

$$\phi_0 = \frac{-igA}{\omega} \frac{\cosh k(z+d)}{\cosh kd} e^{ik(x \cos \beta + y \sin \beta)} \quad (3)$$

Here A is the amplitude of the incident wave, ω is the frequency of the wave, and d is the water depth.

The local pressure acting on the body surface may be obtained by substituting equation 2 into the Bernoulli's equation. The generalized hydrodynamic force F can then be determined by integrating the hydrodynamic pressure over the wetted body surface S_B . The i^{th} component of F can therefore be expressed as

$$F_i = F_i^E + F_i^R \quad (4)$$

where F_i^E denotes the exiting force on a stationary body due to wave diffraction and takes the form

$$F_i^E = -i\omega\rho \iint_{S_B} (\phi_0 + \phi_7) n_i dS \quad (5)$$

Here n_i is the i^{th} component of the generalized normal vector defined as

$$\begin{aligned} n_1 &= n_x \\ n_2 &= n_y \\ n_3 &= n_z \\ n_4 &= zn_x - yn_z \\ n_5 &= xn_z - zn_x \\ n_6 &= yn_x - xn_y \end{aligned} \quad (6)$$

(n_x, n_y, n_z) are the direction cosines of the outward facing normal on the body surface at the point (x, y, z) . This definition also implies that all moment calculations are taken about the axes passing through the fixed origin O_{xyz} . Similarly F_i^R denotes the total hydrodynamic reaction force in the direction i , due to all modes of motion in unit amplitude.

$$F_i^R = -i\omega\rho \sum_{j=1}^6 \left(\iint_{S_B} \phi_j n_i dS \right) \xi_j = \sum_{j=1}^6 f_{ij} \quad (7)$$

It is conveniently decomposed into components in phase with the body acceleration and body velocity associated with each mode of motion j . These are commonly known as the added mass coefficient μ_{ij} and the damping coefficient λ_{ij} respectively since they assume corresponding roles in the equations of motion. μ_{ij} and λ_{ij} may be expressed explicitly as follows.

$$\mu_{ij} = \frac{\rho}{\omega} \iint_{S_b} \text{Im}[\phi_j] n_i dS \quad (8)$$

$$\lambda_{ij} = -\rho \iint_{S_b} \text{Re}[\phi_j] n_i dS \quad (9)$$

3 Numerical Features

To obtain the wave exciting force and the hydrodynamic coefficients using equations 5, 7, 8, and 9, the radiation and diffraction potentials $\phi_j (j = 1, \dots, 7)$ must first be evaluated. The pertinent potential problems have been solved numerically by the integral equation formulation (boundary element method) employing the free surface Green function (see e.g. Eatock Taylor and Chau, 1990 and the cited references). Some features of the present formulation may be noted as the following.

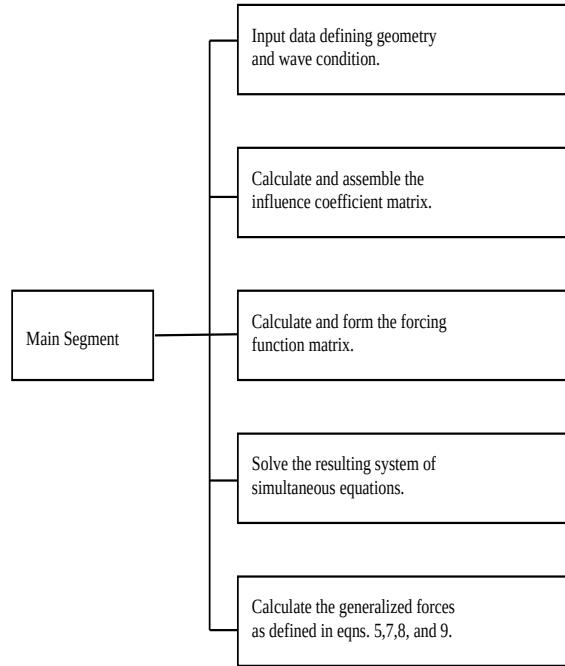


Figure 1: Major program segments

- Based on the idea of isoparametric transformation, 6-node quadratic triangular elements and 8-node quadratic quadrilateral elements are used for modelling both the geometry of the structure and the variations of the potential.
- A spatial coordinate transformation is employed to handle the inherent singularities associated with the integral equation formulation.
- A special routine has been implemented to evaluate the normalized solid angle at all nodal points. This not only reduces the amount of input data but provides a valuable check for possible errors in describing the geometry of the body. (It should be noted that the normalized solid angle over which the fluid is viewed from a smooth surface has the value of $1/2$).
- In order to save computer time and storage, the numerical scheme has been designed to accept two, one, or no vertical planes of symmetry of the body geometry under analysis. The six radiation potentials and the diffraction potentials are separated into symmetric and anti-symmetric modes.

4 Overall Program Description

A modular approach is adopted in the program with separate subroutines being employed to perform the main individual operations. These subroutines are called in sequence by the main segment. Figure 1 highlights the major features of the program.

The first step in the boundary element implementation is the discretization of the wetted body surface into boundary elements, or patches. Each element is completely defined by its nodal points

with respect to the global coordinate system (O_{xyz}). The nodal data which is necessary to represent the geometry of the body must be supplied as input data. In order to reduce the work load and occurrence of human errors in the preparation of the input data, a computer code to automatically generate the element data for the boundary element program presented here has been developed. The output of this mesh generation program is suitable to be fed directly into the boundary element program. This automatic mesh generation scheme is described in Part two of this document.

By discretizing the boundary surfaces into elements, integrals over the body surface are reduced to the summations of integrals over the elements. In order to form system matrices, all the boundary integrals have to be evaluated by successively identifying the field point with each of the nodal points of the element enveloping the body surface. This procedure transforms the problem into a related system of linear algebraic equations. Solution for the diffraction and radiation potentials can be determined by using standard computer routines. From these results, approximations may readily be obtained to the added mass and damping matrices as well as the wave excitation forces by summing up the contribution of each element.

5 Input Data Preparation

The input data that is required for running the program is stored in an external file which may have any name as long as it is a proper UNIX filename. All data is read in a free format mode, (i.e. parameters in each data line can be separated by a space or a comma). An example file is contained in section 8. A description of the input data format is as follows.

Card-1 TITLE

Card-2 MAXRUN IPRINT

Card-3 H AMP BETA ISYM ITYPE

Card-4 NELEM NNODE IPOL

Card-5 IN XYZ(1,IN) XYZ(2,IN) XYZ(3,IN) } NNODE times

Card-6 IE NCN(IE) } NELEM times

Card-7 WK NELEM } MAXRUN times

A description of each variable name used in the input file follows.

- TITLE is the title of the problem or any other identifying information. Maximum length is 40 characters.
- MAXRUN is the total number of frequencies to be analyzed.
- IPRINT controls the option of printing the velocity potential on the body surface. (0 – do not print, 1 – print)
- H is the water depth in meters. If H is negative, infinite water depth analysis is performed.
- AMP is the incident wave amplitude in meters.

- BETA is the incident wave angle in degrees, measured from the positive x axis.
- ISYM defines the number of planes of symmetry in the specification of mesh data that follows.
 - 0 – no planes of symmetry.
 - 1 – one plane of symmetry about $y = 0$. The mesh data should be defined on the half plane $y \geq 0$.
 - 2 – two planes of symmetry about $x = 0$ and $y = 0$. The mesh data should be defined on the $x \geq$ and $y \geq 0$.
- ITYPE is defined so as to facilitate the calculation of the zero frequency Green function.
 - 0 – body is standing on the seabed.
 - 1 – body is freely floating.
- NELEM is the number of elements used to define the mesh.
- NNODE is the number of nodes used to define the mesh.
- IPOL is the number of auxillary polar coordinate systems used, always zero when the mesh generator described in part 2 is used.
- IN represents the node number, varies from 1 to NNODE.
- XYZ(1,IN) is the x-coordinate for node IN.
- XYZ(2,IN) is the y-coordinate for node IN.
- XYZ(3,IN) is the z-coordinate for node IN.
- IE represents the element number, varies from 1 to NELEM.
- NCN(IE) is the number of nodes in element IE. It denotes the type of isoparametric element used for describing the geometry.
 - 6 – six node quadratic triangular element.
 - 8 – eight node quadratic quadrilateral element.
- NCON(k) is the connectivity vector for element IE (i.e. it identifies the global node numbers belonging to element IE). It is defined assuming that the body is viewed from the fluid domain. In the case of a triangular element described by six nodes, NCON(1) may be defined as the node number of any corner node with NCON(2) ... NCON(6) being the numbers of the remaining five nodes in the anticlockwise direction. For quadrilateral elements, NCON(1) may be the node any of the corner nodes, with NCON(2) ... NCON(8) being the remaining seven node numbers in the anticlockwise direction.
- WK is the wave number input for each frequency in the units of meters^{-1} .

6 Output File

The typical output consists of first a printout of all the relevant input data and is self-explanatory. Following this, the results are presented in the following sequence.

1. Normalized solid angle at each nodal point. This will be printed only at the first wave condition of each run.
2. Distribution of the velocity potential on the body surface if IPRINT was set to 1 in the input file. Slightly different output formats result depending on how many planes of symmetry were specified. These are described as follows with a description of the names used to denote output.
 - (a) No planes of symmetry exist.
I K POTENT(K,1,I) } $7 \times \text{NNODE}$ times
 - (b) One plane of symmetry exists.
I K POTENT(K,1,I) POTENT(K,2,I) } $7 \times \text{NNODE}$ times
 - (c) Two planes of symmetry exist.
I K POTENT(K,1,I) POTENT(K,2,I) } $7 \times \text{NNODE}$ times
POTENT(K,3,I) POTENT(K,4,I)
 - I denotes the node number, varies from 1 to NNODE.
 - K identifies the different modes of motion in the specification for the velocity potentials. K equal to (1 ... 6) are associated with the surge, sway, heave, roll, pitch, and yaw. K equal to 0 is associated with the incident wave.
 - POTENT(K,1,I) denotes the velocity potential ϕ_k at node I for which the spatial characteristics are to be supplied.
 - POTENT(K,2,I) denotes the velocity potential ϕ_k at the same spatial location for which node I is mapped symmetrically about the y=0 plane.
 - POTENT(K,3,I) denotes the velocity potential ϕ_k at the same spatial location for which node I is mapped symmetrically about both the x = 0 and the y = 0 plane.
 - POTENT(K,4,I) denotes the velocity potential ϕ_k at the same spatial location for which node I is mapped symmetrically about the x = 0 plane.

3. Generalized forces as defined in equations 5, 7, 8, and 9. In the case of radiation potentials, the output array contains the magnitude of the hydrodynamic force as defined in equation 7 as well as the hydrodynamic coefficients as defined in equations 8, and 9. For diffraction, both the real, imaginary, and magnitude of the excitation force are printed. A description of this output is as follows.

1	μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}	μ_{16}
	λ_{11}	λ_{12}	λ_{13}	λ_{14}	λ_{15}	λ_{16}
	$ f_{11} $	$ f_{12} $	$ f_{13} $	$ f_{14} $	$ f_{15} $	$ f_{16} $
2	μ_{21}	μ_{22}	μ_{23}	μ_{24}	μ_{25}	μ_{26}
	λ_{21}	λ_{22}	λ_{23}	λ_{24}	λ_{25}	λ_{26}
	$ f_{21} $	$ f_{22} $	$ f_{23} $	$ f_{24} $	$ f_{25} $	$ f_{26} $
...	...	...	...	...	...	...
	...	...	...	...	...	...
	...	...	...	...	...	...
6	μ_{61}	μ_{62}	μ_{63}	μ_{64}	μ_{65}	μ_{66}
	λ_{61}	λ_{62}	λ_{63}	λ_{64}	λ_{65}	λ_{66}
	$ f_{61} $	$ f_{62} $	$ f_{63} $	$ f_{64} $	$ f_{65} $	$ f_{66} $
7	$Re[F_1^E]$	$Re[F_2^E]$	$Re[F_3^E]$	$Re[F_4^E]$	$Re[F_5^E]$	$Re[F_6^E]$
	$Im[F_1^E]$	$Im[F_2^E]$	$Im[F_3^E]$	$Im[F_4^E]$	$Im[F_5^E]$	$Im[F_6^E]$
	$ F_1^E $	$ F_2^E $	$ F_3^E $	$ F_4^E $	$ F_5^E $	$ F_6^E $

7 Program Execution

The executable code is called wdra3d and is run by typing the following line:

```
% wdra3d < indata > outdata &
```

The indata file contains all the input data as described in section 5. The outdata file is where the program output is directed. Both indata and outdata must be proper UNIX filenames. The inclusion of a & at the end of the command line causes the program to be run in the background.

The source code consists of many different subroutines that are contained in different files and must be compiled and linked to form the executable code. The subroutine names are outlined in appendix A.

8 Sample input file

A sample input file for wdra3d follows. This input file represents a hemisphere of radius one centered at the origin as indicated by the title line. The second line indicates that two runs at different wave numbers are to be made and that a printout of the velocity potential is not desired. The third line indicates the run conditions; water depth = 2, wave amplitude = 0.1, the incident wave angle = 0 deg, two plane of symmetry employed, and the body is not standing on the bottom. The

fourth line provides information about the mesh, eight elements, and 33 nodes with no auxillary polar coordinate systems. Following this information is the coordinates of the 33 nodes and the connectivity of the eight elements. Finally, the two wave numbers for each run are listed.

One hemisphere center at origin, rad=1.

2	0				
	2.00000000	.10000000	.00000000	2	1
8	33	0			
1		.0000000000	.0000000000	-1.0000000000	
2		.1950903220	.0000000000	-.9807852804	
3		.1379496896	.1379496896	-.9807852804	
4		.0000000000	.1950903220	-.9807852804	
5		.3826834324	.0000000000	-.9238795325	
6		.3535533906	.1464466094	-.9238795325	
7		.2705980501	.2705980501	-.9238795325	
8		.1464466094	.3535533906	-.9238795325	
9		.0000000000	.3826834324	-.9238795325	
10		.5555702330	.0000000000	-.8314696123	
11		.3928474792	.3928474792	-.8314696123	
12		.0000000000	.5555702330	-.8314696123	
13		.7071067812	.0000000000	-.7071067812	
14		.6532814824	.2705980501	-.7071067812	
15		.5000000000	.5000000000	-.7071067812	
16		.2705980501	.6532814824	-.7071067812	
17		.0000000000	.7071067812	-.7071067812	
18		.8314696123	.0000000000	-.5555702330	
19		.5879378012	.5879378012	-.5555702330	
20		.0000000000	.8314696123	-.5555702330	
21		.9238795325	.0000000000	-.3826834324	
22		.8535533906	.3535533906	-.3826834324	
23		.6532814824	.6532814824	-.3826834324	
24		.3535533906	.8535533906	-.3826834324	
25		.0000000000	.9238795325	-.3826834324	
26		.9807852804	.0000000000	-.1950903220	
27		.6935199227	.6935199227	-.1950903220	
28		.0000000000	.9807852804	-.1950903220	
29	1	.0000000000	.0000000000	.0000000000	
30		.9238795325	.3826834324	.0000000000	
31		.7071067812	.7071067812	.0000000000	
32		.3826834324	.9238795325	.0000000000	
33		.0000000000	1.0000000000	.0000000000	
1	6				
1	3	7	6	5	2
2	6				
1	4	9	8	7	3
3	8				

5	6	7	11	15	14	13	10
4	8						
7	8	9	12	17	16	15	11
5	8						
13	14	15	19	23	22	21	18
6	8						
15	16	17	20	25	24	23	19
7	8						
21	22	23	27	31	30	29	26
8	8						
23	24	25	28	33	32	31	27
.20000000							
.40000000							

Part II

Automatic Mesh Generation

The approach taken to automate the creation of the mesh over the wetted surface of a three dimensional body is that of dividing the structure into subregions, or 'parent elements' , and having the computer create a mesh over the geometry described by the parent elements. The primary mesh generation program 'm_gen' described in this document performs this task. However, for a complicated structure the effort required to define the geometry in terms of parent elements may still be considerable. To overcome this, secondary programs have been developed to generate the parent elements that describe a specific geometry (i.e. ship, tlp, etc.) based on more general parameters (length, beam, etc.). The programs described in section two outline these.

By separating the meshing segment from the definition of parent elements, flexibility to describe geometries not covered by the four secondary programs presented here is maintained. In this case it is necessary to either assemble the input file for m_gen by hand or to construct secondary programs for the geometry in question. Guidelines for constructing the input file for the meshing program are contained in section three. In order to verify the correctness of the mesh generated, a verification program as well as a program to graph the mesh have been developed. Section four describes these.

1 Program m_gen

- Title: m_gen (Mesh Generator)
- Author: J.A. Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

1.1 Program Description

The m_gen program generates a mesh over a structure defined by a number of large subregions or 'parent elements'. The program divides the input parent elements into a specified number of divisions and connects the resulting small elements into a coherent mesh. The nature of the final mesh can be controlled somewhat through the selection of parent element types as described below. The final mesh is made entirely of 8 node quadrilaterals and 6 node triangles. The output is written in a format compatible with the wdra3d program.

Four types of parent elements are available. Two of the types are linear elements that are defined by either four or five points and two are quadratic elements that are defined by eight or nine points. The five and nine node elements are included so that some control over the final mesh can be obtained. The center node influences the mesh in the interior area of the parent element. There are no restrictions in the arrangement of the parent elements, two or more elements can have common sides and/or corners. In addition, the orientation of common sides need not be the same. However, if a four or five node element has a common side with a five or nine node element, the intermediate point in the quadratic parent element must lie exactly midway between the corner nodes. Triangular parent elements are created by collapsing one side of any of the parent element

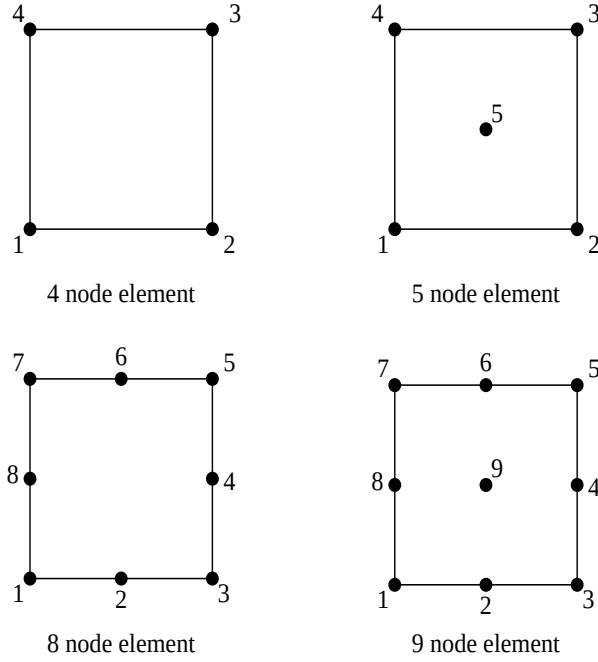


Figure 2: Parent element numbering.

types into one point. When this is done, the parent element is still described by the same number of points but two or three points have the same coordinates. The program places six node triangular elements in the corner created by the collapsed side.

The coordinates of the points defining the parent elements can be expressed in either Cartesian, cylindrical, or spherical coordinates. This overcomes the problem of modeling circular surfaces with a quadratic shape function. Each parent element is defined in its own coordinate system that may be translated from the origin of the global system, making it possible to describe a circular or spherical surface that is not centered at the global origin.

Ordinarily the program divides each parent element into the specified number of divisions in a regular manner. If desired, a parent element can be divided such that the spacing between the nodes across the element reflect a cosine curve.

1.2 Input File Format

The input data for the mesh generation program is read from an input file which is described below. This file can either be assembled by hand or one of the programs that generates the input file based on more general structure parameters may be used.

The data that `m_gen` requires consists of five items for each parent element.

1. Parent element type. (4, 5, 8, or 9 nodes)
2. Parent element coordinate system type. (Cartesian, cylindrical, or spherical)
3. Origin of parent element axis in global coordinates. (x, y, z)

4. Coordinate of each point that defines the parent element.
5. Number of divisions to be made in the parent element. (x and y directions)

In addition the input file must contain the following information that is passed along to the boundary element program.

1. Water depth.
2. Incident wave amplitude.
3. Incident wave angle, measured from the x axis.
4. Definition of structure configuration, free floating or standing on the seabed.
5. Indication of whether or not a printout of the potential distribution is desired.
6. Number of planes of symmetry utilized.
7. Number of runs at different wave numbers to be made.
8. Value of wave number for each run.

The format of the input file is as follows. Each card has a one line title, the required information, and a final blank line. A sample input file is included in section 1.3.

Card-1 Title (up to 40 characters long)

Card-2 Total number of parent elements.

Card-3 Element types (one line for each parent element).

ELTYPE COORDSYS AXIS_X AXIS_Y AXIS_Z

Card-4 Boundary points (groups of 4,5,8, or 9 points seperated by a blank line, each point consists of three coordinates).

Card-5 Divisions (one line for each parent element).

IDIVX IDIVY

Card-6 Number of runs at different wave numbers.

Card-7 Wave numbers for each run (one line for each value).

Card-8 Run conditions.

H AMP BETA ISYM ITYPE IPRINT

A description of the names used above follows.

- ELTYPE denotes the type of parent element used. It may have one of four values:
 - 1 = Four node linear quadrilateral.
 - 2 = Five node quadrilateral.
 - 3 = Eight node quadratic quadrilateral.
 - 4 = Nine node quadratic quadrilateral.

- COORDSYS denotes the parent element coordinate system type and can have one of three values:
1 = Cartesian coordinates.
2 = Cylindrical coordinates.
3 = Spherical coordinates.
- AXIS_X is the x-coordinate of the parent element origin expressed in global coordinates.
- AXIS_Y is the y-coordinate of the parent element origin expressed in global coordinates.
- AXIS_Z is the z-coordinate of the parent element origin expressed in global coordinates.
- IDIVX is the number of divisions in the x-direction of the parent element. If either of these is negative, cosine spacing for the parent element is invoked. A negative IDIVY creates closer spacing at the bottom of the element (on the side with node 1). A negative IDIVX creates closer spacing at the right of the element (on the side with node 3).
- IDIVY is the number of divisions in the y-direction of the parent element.
- H is the water depth in meters. If H is negative, infinite water depth analysis is performed.
- AMP is the incident wave amplitude in meters.
- BETA is the incident wave angle in degrees, measured from the positive x axis.
- ISYM defines the number of planes of symmetry in the specification of mesh data that follows.
 - 0 – no planes of symmetry.
 - 1 – one plane of symmetry about y=0. The mesh data should be defined on the half plane $y \geq 0$.
 - 2 – two planes of symmetry about x=0 and y=0. The mesh data should be defined on the $x \geq$ and $y \geq 0$.
- ITYPE is defined so as to facilitate the calculation of the zero frequency Green function.
 - 0 – body is standing on the seabed.
 - 1 – body is freely floating.
- IPRINT controls the option of printing the velocity potential on the body surface. (0 – do not print, 1 – print)

1.3 Sample Input File

An example of a typical input file to the m_gen program follows. This input file will create two hemispheres of radius = 1 . One is centered at (1,1) and the other at (-1,-1).

Title

Two hemispheres, radius = 1.

Number of superelements

2

Element types

3	3	-1.000000	-1.000000	.000000
3	3	1.000000	1.000000	.000000

Boundary points

1.000000	.000000	180.000000
1.000000	180.000000	180.000000
1.000000	360.000000	180.000000
1.000000	360.000000	135.000000
1.000000	360.000000	90.000000
1.000000	180.000000	90.000000
1.000000	.000000	90.000000
1.000000	.000000	135.000000

1.000000	.000000	180.000000
1.000000	180.000000	180.000000
1.000000	360.000000	180.000000
1.000000	360.000000	135.000000
1.000000	360.000000	90.000000
1.000000	180.000000	90.000000
1.000000	.000000	90.000000
1.000000	.000000	135.000000

Divisions

8	4
8	4

Number of runs

2

Wave numbers for each run

.200000
.400000

Run conditions

2.000000	.100000	.000000	0	1	0	0
----------	---------	---------	---	---	---	---

1.4 Program Execution and Memory Requirements

The `m_gen` program is executed by issuing the following command line:

```
% m_gen < infile > outfile &
```

where `infile` is the input file for the `m_gen` as described above and `outfile` is the name of the file containing the completed mesh, this file is in the correct format to be used as an input file by the `wdra3d` program described in section one.

The `m_gen` program requires 250k of memory and can create structures with up to 500 elements. Details on the compilation of the source code are included in appendix A.

2 Secondary Programs

The four programs developed to create the `m_gen` input file for specific geometries are documented in this section. The output of each program is in the exact format required by the `m_gen` program described above.

2.1 Hemisphere Geometries

- Title: hemisphere
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

2.1.1 Program Description

The hemisphere program generates the appropriate parent element configuration for one or more hemispheres (up to 50) of specified radius and center position. The center of each hemisphere lies on the undisturbed free surface. Each hemisphere consists of one parent element defined in spherical coordinates. The program output is suitable to be used directly as an input file to the mesh generation program `m_gen`.

2.1.2 Input File Format

The hemispheres are described in the input file by their radius and by their origin in the global coordinate system. In addition, the input file must contain the run conditions for the `wdra3d` program. The structure of the input file is outlined below, each card consists of a label line, the required information, and finally a blank line. A sample input file is contained in section 2.1.3.

Card-1 Title (up to 40 characters long).

Card-2 Output file name to contain the mesh description file (up to 30 characters long).

Card-3 Total number of hemispheres (up to 50).

Card-4 Axis of each hemisphere (one line for each hemisphere).

AXIS_X AXIS_Y

Card-5 Radius of each hemisphere (one line for each hemisphere).

Card-6 Number of divisions in the θ direction.

Card-7 Number of divisions in the ϕ direction.

Card-8 Number of axis of symmetry to utilize (0,1,2).

Card-9 Water depth.

Card-10 Wave amplitude.

Card-11 Wave angle.

Card-12 Structure configuration (0=free floating, 1=standing on seabed).

Card-13 Print potential distribution (1=print, 0=no print).

Card-14 Number of runs.

Card-15 Wave number for each run (one line for each run).

2.1.3 Sample Input File

A typical example of an input file for the hemisphere program follows:

TITLE:

Two Hemispheres

NUMBER OF HEMISPHERES:

2

AXIS OF EACH HEMISPHERE (X,Y):

-1.0 -1.0

1.0 1.0

RADIUS OF EACH HEMISPHERE:

1.0

1.0

NUMBER OF DIVISIONS IN THE THETA DIRECTION (per 360 deg.):

8

NUMBER OF DIVISIONS IN THE PHI DIRECTION (per 360 deg.):

4

NUMBER OF AXIS OF SYMMETRY (0,1,2):

0

WATER DEPTH:

-2.0

WAVE AMPLITUDE:

1.0

WAVE ANGLE:

0.0

ITYPE:

1

PRINT POTENTIAL DISTRIBUTION (1=PRINT, 0=NO PRINT):

0

NUMBER OF RUNS:

2

WAVE NUMBER FOR EACH DIFFERENT RUN:

.2

.4

2.1.4 Program Execution and Memory Requirements

The hemisphere program is executed by issuing the following command line:

```
% hemisphere < infile > outfile &
```

where infile is the name of the file containing the input file as described above and outfile is the name of the file that will be used as an input file to the m_gen program. The hemisphere program requires 25k of memory to run and can accomodate up to 50 hemisphere geometries.

2.2 Truncated Cylinder Geometries

- Title: t_cyl
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

2.2.1 Program Description

The t_cyl program generates the appropriate parent element configuration for one or more truncated cylinders of specified radius, length, and center position. The top of the cylinders coincides with the free surface. Each hemisphere consists of two parent elements defined in cylindrical coordinates. The program output file is named 'm_gen.dat' and is suitable to be used directly as an input file to the mesh generation program m_gen.

2.2.2 Input File Format

The truncated cylinders are described by their radius, length, and centerline position in the global coordinate system. In addition, the input file must contain the run conditions for the wdra3d program. The structure of the input file is outlined below, each card consists of a label line, the required information, and finally a blank line. A sample input file is contained in section 2.2.3.

Card-1 Title (up to 40 characters long)

Card-2 Total number of cylinders (up to 50).

Card-3 Axis of each cylinder (one line for each cylinder)
AXIS_X AXIS_Y

Card-4 Radius and length of each cylinder (one line for each cylinder)

Card-5 Number of divisions in the θ , r and z directions.

Card-6 Number of axis of symmetry to utilize (0,1,2).

Card-7 Water depth.

Card-8 Wave amplitude.

Card-9 Wave angle.

Card-10 Structure configuration (0=free floating, 1=standing on seabed).

Card-11 Print potential distribution (1=print, 0=no print).

Card-12 Number of runs.

Card-13 Wave number for each run (one line for each run).

2.2.3 Sample Input File

A typical example of an input file to the t_cyl program follows. This input file generates one cylinder but specifies that two planes of symmetry be used by the wdra3d program.

TITLE:

Four Cyl: H=1, D=1, h=2, L=1.4, Beta=180.

NUMBER OF CYLINDERS:

1

AXIS OF EACH CYLINDER (X,Y):

0.7 0.7

CYLINDER RADIUS AND LENGTH:

0.5 1.0

NUMBER OF DIVISIONS IN THE THETA, R, AND Z DIRECTIONS:

8 4 4

NUMBER OF AXIS OF SYMMETRY (0,1,2):

2

WATER DEPTH:

2.0

WAVE AMPLITUDE:

0.2

WAVE ANGLE:

180.0

ITYPE:

1

PRINT POTENTIAL DISTRIBUTION (1 = PRINT, 0 = NO PRINT):

0

NUMBER OF RUNS:

3

WAVE NUMBER FOR EACH DIFFERENT RUN:

0.1

0.3

0.5

2.2.4 Program Execution and Memory Requirements

The `t_cyl` program is executed by issuing the following command line:

```
% t_cyl < infile > outfile &
```

where `infile` is the name of the file containing the input file as described above and `outfile` is the name of the file that will be used as an input file to the `m_gen` program. The `t_cyl` program requires 38k of memory to run and can accomodate up to 50 truncated cylinders.

2.3 Tension Leg Platform Structures

- Title: tlp
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

2.3.1 Program Description

The tlp program generates the parent element configuration for the geometry of a tension leg platform type structure with four cylinders located at the corners of the structure connected by pontoons. The structure has two planes of symmetry so the user must define only one quadrant. The cylinders are described by their radius and length as well as the position of the centerline in the global coordinate system. Each pontoon is described independently by height and width. In addition, the distance from the bottom of the cylinder to the bottom of the pontoon is required from the user. The degree of final mesh fineness is indicated by the user as well as the number of planes of symmetry that the wdra3d program is to use. However, because the structure has two planes of symmetry it is redundant to use less than two planes of symmetry.

2.3.2 Input File Format

The input file for the tlp program contains the description of the structure in terms outlined above. In addition, the input data must contain the run conditions that are passed along to the boundary element program. The exact format of this input file follows, each card consists of a title line, the required information, and finally a blank line. A sample input file is contained in section 2.3.3.

Card 1: Title (up to 40 characters long)

Card 2: Column radius and length.

RADIUS LENGTH

Card 3: Column axis. (x and y coordinates)

AXIS_X AXIS_Y

Card 4: Dimensions of pontoon 1.

HEIGHT1 WIDTH1 L1 RAD1

Card 5: Dimensions of pontoon 2.

HEIGHT2 WIDTH2 L2 RAD2

Card 6: Division size in the θ , R, and Z directions.

DIV_TH DIV_R DIV_Z DIV_L1 DIV_L2

Card 7: Number of axis of symmetry (0,1,2)

Card 8: Water depth.

Card 9: Wave amplitude.

Card 10: Wave angle.

Card 11: Structure configuration (0=free floating, 1 = standing on seabed).

Card 12: Print potential distribution (1 = print, 0 = no print).

Card 13: Number of runs.

Card 14: Wave number for each run (one line for each run).

A description of the names used above follows.

- RADIUS defines the radius of each column.
- LENGTH specifies the dimension from the bottom of the column to the water surface.
- AXIS_X and AXIS_Y specify the x and y coordinates of the column centerline.
- HEIGHT1 and HEIGHT2 specify the vertical dimension of the two pontoons.
- WIDTH1 and WIDTH2 specify the horizontal dimension of the two pontoons.
- L1 and L2 specify the distance between the bottom of the column and the bottom of the two pontoons, equal to zero if pontoons are flush with the bottom of the columns.
- RAD1 and RAD2 specify the radius of the corner of the pontoons, equal to zero if corner sharp.
- DIV_R specifies the division size in the radial direction.
- DIV_TH specifies the division size in the θ direction. (in degrees)
- DIV_Z specifies the division size in the vertical direction.
- DIV_L1 and DIV_L2 specify the division size along each of the two pontoons.

2.3.3 Sample Input File

A typical input file for the tlp program follows:

TITLE:

Two plane of sym. ISSC TLP - 38 panels.

CYLINDER RADIUS AND LENGTH:

8.435 35.0

CYLINDER AXIS (X AND Y):

43.25 43.25

DIMENSIONS OF PONTOON 1:

10.5 7.5 0.0 0.0

DIMENSIONS OF PONTOON 2:

10.5 7.5 0.0 0.0

DIVISION SIZE IN THE R, THETA (DEG), Z AND L DIRECTIONS:

8.0 80.0 15.0 20.0 20.0

NUMBER OF AXIS OF SYMMETRY (0,1,2):

2

WATER DEPTH:

400.0

WAVE AMPLITUDE:

2.0

WAVE ANGLE:

180.0

ITYPE:

1

PRINT POTENTIAL DISTRIBUTION (1 = PRINT, 0 = NO PRINT):

0

NUMBER OF RUNS:

4

WAVE NUMBER FOR EACH DIFFERENT RUN:

0.0028986

0.0057971

0.0086957

0.0115942

2.3.4 Program Execution and Memory Requirements

The tlp program is executed by issuing the following command line:

```
% tlp < infile > outfile &
```

where infile is the name of the file containing the input file as described above and outfile is the name of the file that will be used as an input file to the m_gen program. The tlp program requires 88k of memory to run.

2.4 Ship Geometries

- Title: ship
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

2.4.1 Program Description

The ship program generates the parent elements to define a ship structure based on the offsets of the ship as well as information about the profile plan. It is possible to define the geometry of a monohull or SWATH type ship. In either case only the offsets defining one half of one hull are required. It is thus necessary that each hull be symmetric about the centerline. It is also necessary that there are the same number of offsets at each station.

2.4.2 Input File Format

The ship hull is described by the offsets at each station, and a description of the ship profile at the bow and stern. It is necessary that there are the same number of offsets at each station. Station zero corresponds to the forward perpendicular. To allow closer station spacing where desired, fractional station numbers are allowed. The offsets are given as pairs of numbers corresponding to the height above the keel and the half breadth of the ship. The bow profile is given as the the height above the keel and the distance of the bow ahead of the fore perpendicular. Similarly, the stern profile is given as the height above the keel and the distance of the stern behind the aft perpendicular. Finally, the wdra3d run information is input along with the type of ship, monohull or SWATH.

The format of the input file is outlined below, each card consists of a title line, the required information, and a blank line. A sample input file is contained in section 2.4.3.

Card 1: Title (up to 40 characters long)

Card 2: Number of stations, and number of offsets at each station. NSTATION NOFFSET

Card 3: Station spacing.

Card 4: Station number of aft perpendicular.

Card 5: Bow profile.

H B } NOFFSET times

Card 6: Offsets for station number N } NSTATION times

H HBREADTH } NOFFSET times

Card 7: Stern profile.

H S } NOFFSET times

Card 8: Number of divisions in length and draft directions.

Card 9: Draft of ship.

Card 10: Number of axis of symmetry (0,1,2)

Card 11: Ship type. (0 – Monohull, 1 – SWATH)

Card 12: Water depth.

Card 13: Wave amplitude.

Card 14: Wave angle.

Card 15: Structure configuration (0=free floating, 1=standing on seabed).

Card 16: Print potential distribution (1=print, 0 = no print).

Card 17: Number of runs.

Card 18: Wave number for each run (one line for each run).

A description of the names used above follows.

- NSTATION specifies the total number of stations at which offsets are given. item NOFFSET specifies the number of offsets at each station. item H specifies the height at which the offset or profile value is specified. item N specifies the station number. item B specifies the bow profile values. item HBREADTH specifies the half breadth at a particular height at a particular station. item S specifies the stern profile values.

2.4.3 Sample Input File

A typical input file for the ship program follows:

TITLE:

Ship

OF STATIONS, # OF OFFSETS AT EACH STATION:

4 5

STATION SPACING

10.0

STATION NUMBER OF AFT PERPENDICULAR

2.

BOW PROFILE

0.3 10.

1.0 23.

2.0 10.

3.0 14.

4.0 16.

STATION # -0.5

0.2 0.0

1.0 1.4

2.0 2.15

3.0 2.65

4.0 2.7

STATION # 0.0

0.0 0.0

1.0 1.5

2.0 2.25

3.0 2.75

4.0 2.8

STATION # 1.0

0.0 0.0

1.0 1.5

2.0 2.25

3.0 2.3

4.0 2.5

STATION # 2.0

0.0 0.0

1.0 1.3

2.0 1.7

3.0 1.0

4.0 1.2

AFT PROFILE

0.0 10.

1.0 10.

2.0 10.

3.0 10.

4.0 10.

NUMBER OF DIVISIONS IN THE LENGTH AND DRAFT DIRECTIONS:

4 4

DRAFT OF SHIP:

4.0

NUMBER OF AXIS OF SYMMETRY (0,1):

1

SHIP TYPE (1=MONOHULL, 2=SWATH)

1

WATER DEPTH:

2.0

WAVE AMPLITUDE:

1.0

WAVE ANGLE:

0.0

ITYPE:

1

PRINT POTENTIAL DISTRIBUTION (1 = PRINT, 0 = NO PRINT):

0

NUMBER OF RUNS:

3

WAVE NUMBER FOR EACH DIFFERENT RUN:

.5228

1.0

2.0

2.4.4 Program Execution and Memory Requirements

The ship program is executed by issuing the following command line:

```
% ship < infile > outfile &
```

where infile is the name of the file containing the input file as described above and outfile is the name of the file that will be used as an input file to the m_gen program. The ship program requires 190k of memory to run and can accomodate up to 40 stations with 40 offsets at each station.

3 Other strucures

If a structure of a type not included by the above programs is to be used, it is neccessary to either construct the `m_gen` input file by hand or to write additional secondary programs to generate the parent element configuration desired. This section provides guidelines for connecting parent elements together into the desired structure as well as describing some existing subroutines that can be used within FORTRAN programs written to create a `m_gen` input file.

3.1 Construction of input files

When constructing a `m_gen` input file by hand or when writing a secondary program to perform this task, a few features of the `m_gen` program should be kept in mind. The normal vector of the parent elements should always be directed out of the structure. The normal vector of the parent elements is defined such that the normal vector of the elements shown in figure 2 is pointed out of the page. Different types of parent elements may share sides and/or corners with no restrictions except in the case of a parent element with four or five nodes shareing a side with a parent element with eight or nine nodes. In this case, the node in the center of the edge must lie exactly halfway between the corners of the parent element. Finally, when a parent element defined in cylindrical or spherical coordinates adjoins an element defined in Cartessian coordinates, it is best that the Cartessian parent element has a higher number than the cylindrical or spherical parent element.

3.2 Useful subroutines

The following three subroutines have been written to aid in writing programs to create parent element configurations. The first two are useful when it is neccessary to generate a mesh that is symmetric about the x or y axis. The third subroutine writes the `m_gen.dat` file in the correct format given all the neccessary information.

3.2.1 Subroutine `mirror_x`

This subroutine mirrors the existing parent elements about the plane $x = 0$. Therefore, it is neccessary that parent elements are defined on only one side of this plane before calling the routine. The arguments of the subroutine are the following (in order):

- bpts: Array defining the points making up each parent element.
- axis: Array defining the global coordinates of the origin of each parent element coordinate system.
- idiv: Array defining the number of divisions to make in each parent element.
- n_coord: Array defining the parent element type and the coordinate system type of each parent element.
- ntse: Total number of parent elements defined, before mirror operation.
- max: Exact value of first dimension of `bpts`, `axis`, `idiv`, and `n_coord` in calling program.

3.2.2 Subroutine mirror_y

This subroutine mirrors the existing parent elements about the plane $y = 0$. Therefore, it is necessary that parent elements are defined on only one side of this plane before calling the routine. The arguments of the subroutine are the following:

- bpts: Array defining the points making up each parent element.
- axis: Array defining the global coordinates of the origin of each parent element coordinate system.
- idiv: Array defining the number of divisions to make in each parent element.
- n_coord: Array defining the parent element type and the coordinate system type of each parent element.
- ntse: Total number of parent elements defined, before mirror operation.
- max: Exact value of first dimension of bpts, axis, idiv, and n_coord in calling program.

3.2.3 Subroutine write

This subroutine writes out the m_gen.dat file in the exact form required by the m_gen program. The arguments of the subroutine are (in order):

- title: Character string of length 40 containing title information.
- bpts: Array defining the points making up each parent element.
- axis: Array defining the global coordinates of the origin of each parent element coordinate system.
- idiv: Array defining the number of divisions to make in each parent element.
- n_coord: Array defining the parent element type and the coordinate system type of each parent element.
- ntse: Total number of parent elements defined, before mirror operation.
- max: Exact value of first dimension of bpts, axis, idiv, and n_coord in calling program.

4 Mesh Verification and Visualization

Two programs have been written to aid in establishing the correctness of the final mesh. The first of these is called 'meshver' and examines the mesh point by point and checks for errors. The second puts the mesh in a form that allows plotting using the 'gnuplot' plotting software.

4.1 Program meshver

- Title: meshver
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

4.1.1 Program Description

This program performs two tasks. First, the coordinates of each node are compared such that the condition of two nodes with the same coordinates is located. In a correct mesh this will never be the case and the program returns the message that zero points have common coordinates. If the mesh is flawed and two or more points do have common coordinates, the program lists these incidences. Second, the program checks each element and ensures that each node number in the element exists and is unique within that element.

4.1.2 Program execution

The program is run by typing 'meshver'; the filename containing the mesh description (generated by m_gen) is to be entered by the user and will be prompted for.

4.2 Mesh Plotting

The gnuplot software requires that the mesh data be in a slightly different format than the wdra3d program requires. The 'plot' program described below puts the mesh data in the appropriate format. Also, notes on using gnuplot to view the graph are included.

4.2.1 Plotting program

- Title: plot
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

4.2.2 Program Description

This program prompts the user for the filename of the mesh data as output from the `m_gen` program. It is necessary to enclose the filename in quotes. The program outputs the data in a file named `'plot_xyz.dat'`. This file consists simply of each element defined by a set of nine points, with a blank line separating each element. The mesh is plotted by simply connecting each set of nine points with straight lines.

4.3 Plotting with Gnuplot

A gnuplot load file has been created to perform the graphing, therefore, only the following command line is necessary to create the plot.

```
% gnuplot plot.gnu
```

Part III

Post Processing of Results

The output of the wdra3d program is not in a form that is easily graphed. To overcome this a process of converting the raw output to graphical format has been developed. This involves two steps, rewriting the output in a form in which the data can be easily graphed by existing software, and the subsequent plotting of the data. Program plot described below scales the result as indicated in the input file, performs a spline interpolation between the computed points, and rewrites the data in a form that can be easily graphed. The gnuplot software is then used to make the graphs.

1 Program plot_gnu

- Title: plot_gnu
- Author: J. Andrew Hamilton
- Source Language: FORTRAN 77
- Machine: IBM 6000

1.1 Program Description

Program plot nondimensionalizes the results as indicated by the user in the input file, performs a spline interpolation between the computed points, and rewrites the data in a form that can be easily graphed. The program creates graphs of added mass and damping for the nonredundant and nonzero cases.

1.2 Program Input

It is necessary for the user to enter the filename containing the wdra3d output, information about the nondimensionalization of the values, and the number of runs at different wave numbers made. The program prompts the user for this information. The various results are divided by the values entered as nondimensionalization constants.

2 Plotting with Gnuplot

After the plot program is run, a number of files will exist that contain the nondimensionalized wdra3d results and the cubic spline curves. A macro file for the Gnuplot program has been created to aid in the plotting of the data. This file is named plot.gnu; to create the graphs, type 'gnuplot plot.gnu'.

A Compilation notes on all programs

Most of the programs discussed in this documentation consist of many subroutines that are contained in different files. Compilation of these is required to form the executable files. This appendix outlines the subroutines and filenames that make up each program. To aid in the compilation process the 'make' program is utilized. Makefiles for each program have been created and to compile a program it is necessary to type only:

```
% make -f filename
```

where 'filename' is the name of the makefile for the specific program. These makefile names are indicated in the following sections.

A.1 wdra3d

The filenames containing the source codes that make up the wdra3d program are as follows.

- wdra3d.f, meshda.f, bodyfd.f, intbod0p.f, intbod1p.f, intbod2p.f, dinpo2x.f, dinpo2y.f, dinpo2z.f, inpot.f, sing0p.f, sing1p.f, sing2p.f, tripol.f, assemb0p.f, assemb1p.f, assemb2p.f, dforce0p.f, dforce1p.f, fingreen.f, zerofq.f, solve.f

The makefile that will compile and link these separate files is named Makefile.

A.2 m_gen

The filenames containing the source codes that make up the m_gen program are as follows.

- m_gen.dat, modify.f, mesh.f, exists.f, n.f, ln.f.

The makefile that will compile and link these separate files is named make_mgen.

A.3 hemisphere

The filenames containing the source codes that make up the hemisphere program are as follows.

- hemisphere.f, write.f

The makefile that will compile and link these separate files is named make_hemisphere.

A.4 t_cyl

The filenames containing the source codes that make up the t_cyl program are as follows.

- t_cyl.f, write.f

The makefile that will compile and link these separate files is named make_tcyl.

A.5 tlp

The filenames containing the source codes that make up the tlp program are as follows.

- tlp.f, column.f, pon.f vect.f, mirror_x.f, mirror_y.f, write.f

The makefile that will compile and link these separate files is named make_tlp.

A.6 ship

The filenames containing the source codes that make up the ship program are as follows.

- ship.f, spline.f, mirror_y.f, write.f

The makefile that will compile and link these separate files is named make_ship.

A.7 meshver

The filenames containing the source code that make up the meshver program are as follows.

- meshver.f

The make program is not needed because this program is only one segment. To compile and link, use the command 'xlf -o meshver meshver.f'.

A.8 plot

The filenames containing the source code that make up the plot program are as follows.

- plot.f

The make program is not needed because this program is only one segment. To compile and link, use the command 'xlf -o plot plot.f'.

A.9 plot_gnu

The filenames containing the source codes that make up the plot_gnu program are as follows.

- plot_gnu.f, spline.f

The makefile that will compile and link these separate files is named make_plotgnu.

References

- [1] Chau, F.P., The second order velocity potential for diffraction of waves by fixed offshore structures, *PhD Thesis* Univerisity of London, (1989).
- [2] Eatock Taylor, R. and Chau F.P., Wave Diffraction Theory – Some developments in Linear and Non-Linear Theory, 1991 Offshore Mechanics and Offshore Engineering Conference, Norway **1A** pp 19-28.
- [3] Zienkiewicz & Phillips, An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Coordinates, *International Journal for Numerical Methods in Engineering* **3** (1971) 519–528.