
TIMIT

A panel-method program for transient wave-body interactions.

VERSION 4.0: For zero and forward speed analysis of a single body with any number of waterlines, arbitrary wave heading, generalized modes, and infinite or finite depth.

F. T. Korsmeyer H. B. Bingham J. N. Newman

Research Laboratory *of* Electronics
Massachusetts Institute of Technology

May 1999

©1995-1999 Massachusetts Institute of Technology. All rights reserved.

If you have questions or comments concerning this manual or T̄MIT please email them to:
`timit@chf.mit.edu`.

The text of this manual is printed in the Times Roman typeface. Words, symbols, and arabic numerals, which would appear on your terminal screen as commands or lines in files, are printed in `typewriter` typeface.

The following terms used in this manual are trademarks: MS-DOS is a trademark of Microsoft Corporation; PostScript is a trademark of Adobe Systems, Inc.; UNICOS is a trademark of Cray Research Corporation; UNIX is a trademark of AT&T; WAMIT is a trademark of the Massachusetts Institute of Technology.

The T̄MIT code is a direct result of research by: Tom Korsmeyer, Harry Bingham, Nick Newman, Marcos Ferreira, Thomas Farstad, Don Danmeier, Anne-Katrine Bratland, and Ali Beskok.

Many others have contributed through their research as members of the Computational Hydrodynamics Facility and the Research Laboratory *of* Electronics at MIT.

The development of T̄MIT has been supported by the Office of Naval Research, the Joint Industry Project “Wave Effects on Offshore Structures”, the Consortium for Numerical Analysis of Wave Effects on Offshore Structures, and the Naval Ship Warfare Center.

Contents

1	Introduction	3
2	Program Overview	5
3	Getting Started	6
3.1	Portability and Coding Conventions	6
3.2	The T̄MIT Distribution	7
3.3	How to Install T̄MIT	9
3.3.1	Unpacking the archive	9
3.3.2	Editing the makefile	9
3.3.3	Creating the binary data in VTGN98.BNY	10
3.4	Verifying Your T̄MIT Distribution	10
3.5	Creating and Running the T̄MIT Executables	11
3.5.1	Creating T̄MIT executables	11
3.5.2	Runing <code>potent</code> and <code>respon</code>	11
4	Input to T̄MIT	13
4.1	File-names List File, FLF	13
4.2	The Potential Control File, PCF	14
4.3	The Geometric Data File, GDF	16
4.4	The Basis Flow File, BFF	19
4.5	The Response Control File, RCF	20
4.6	The Body Property File, BPF	23
4.7	The Incident Wave File, IWF	25
4.8	The eXtended Geometry File, XGF	27
4.9	The Generalized Mode File, <code>defmod.f</code>	28
5	Output from T̄MIT	29
5.1	Hydrostatic Quantities	29
5.1.1	The format of the hydrostatic output	30
5.2	Time Domain Hydrodynamic Quantities	30
5.2.1	Format of the time domain hydrodynamic output	31
5.3	Frequency Domain Hydrodynamic Quantities	31
5.3.1	Format of the frequency domain output	33
6	Strategies for Using T̄MIT	34
6.1	Memory Allocation in T̄MIT	34
6.2	Optimizing T̄MIT	34
6.3	Efficiency	34
6.4	Effective GDF's	36
6.5	Time-step Size	37
6.6	Separating the Radiation and Diffraction Problems	38
6.7	Inclusion of Nonlinear Damping	38
6.8	Generalized Modes	39

6.9	Negative Forward Speed	39
6.10	Very Low Froude Number	40
6.11	Error Checking	40
7	Mathematical Formulation	42
7.1	The Inertial Terms	43
7.2	The Hydrostatic Terms	43
7.3	The Hydrodynamic Pressure	44
7.4	The Potential Formulation	46
7.5	Source Only Integral Equations	46
7.6	Finite Depth	47
7.7	Solving for the Radiation Potentials	48
7.7.1	Decomposition of the radiation source strengths	49
7.8	Solving for the Scattered Potential	50
7.9	First Order “Steady” Forces	51
7.10	Transient Integrated Quantities	51
7.11	Following Seas When $U > 0$	54
7.12	The Haskind Relations	55
7.13	Transient Local Quantities	57
7.14	Asymptotic Continuation	58
7.15	Time Harmonic Integrated Quantities	59
7.16	Time Harmonic Local Quantities	60
7.17	Second Order Steady Forces	60
7.18	Exact Equations of Motion and Nonlinear Hydrostatics	61
7.18.1	Exact equations of motion	62
7.18.2	Nonlinear hydrostatics	63
7.19	Generalized Modes	63
7.20	Numerical Solution	63
A	Description of the Code Structure	67
A.1	A Task Oriented Outline	67
A.2	A Subroutine Oriented Outline	69

1 Introduction

The suite of programs called T̄MIT performs a linear seakeeping analysis for bodies with or without forward speed. There are two main computational modules of T̄MIT called **potent** (say: poten-tee') and **respon**. They solve the transient radiation and diffraction problems and provide: the steady force and moment, impulse-response functions, time histories of body response in a prescribed sea of arbitrary frequency content, frequency-domain coefficients, response-amplitude operators and the second-order mean force and moment. Results are provided in terms of both global and local quantities. Other programs referred to as *utilities* are provided to aid in pre-processing the input and post-processing the output.

The general hydrodynamic problem of a body in motion near a free surface of a fluid is simplified to be a linearized problem of finding the velocity potential (or the source strength distribution) on the body surface under the assumption that the fluid is ideal and in irrotational motion. The fluid depth may be infinite or finite (with some limitation) and the body, which may be surface-piercing or submerged, may be undergoing steady translation in addition to small motions about its mean position.

The determination of useful hydrodynamic quantities proceeds by solving a set of canonical problems with impulsive forcing. These canonical problems will be referred to as the *radiation problem* and the *diffraction problem*. Solving the radiation problem means finding the velocity potential (or source distribution) forced by an impulsive body motion in an otherwise undisturbed streaming flow. Solving the diffraction problem means finding the velocity potential (or source distribution) forced by an impulsive incident wave impinging on the body fixed to its mean position. Steady calm water resistance, sinkage force, and trim moment are obtained from the steady state limit of a particular radiation problem. These initial-boundary-value problems are transformed into integral equations, for either the potential or the source distribution, by using Green's theorem and the transient free-surface Green function. In fact, for each of the potential or source formulations, a single integral equation is solved for all of these problems, the distinctions between them leading only to differing boundary conditions appearing in the right-hand sides. The integral equation is discretized by a *panel method*. This approach requires the discretization of the wetted surface of the body into an ensemble of quadrilaterals (or triangles) referred to as panels. Collocation at the panel centroids leads to a linear system that must be factored initially, and then solved for new right-hand sides at each time step. The solution is the value of the velocity potential or the source strength on each panel at each time step for each radiation mode, or the scattering of the incident wave.

A second step in the computation is the integration of the potentials into the hydrodynamic quantities: impulse-response functions. Impulse-response functions from radiation problems are the time-domain counterpart of added-mass and damping coefficients in the frequency domain. Impulse-response functions from the diffraction problem are the time-domain counterpart of exciting force coefficients in the frequency domain.

These impulse-response functions may be Fourier transformed into the frequency domain or they may be used in simulations of general body motions in the time domain.

The following analyses may be conducted in finite or infinite water depth with TIMIT Version 4.0:

- Computation of hydrostatic coefficients.
- Computation of the transient first-order calm water resistance, sinkage force, and trim moment from the large-time limit of a certain impulsive problem.
- Computation of impulse-response functions for the radiation and diffraction forces.
- Linear time-domain simulations in (pseudo-)random seas from a single heading angle with the following output:
 - Body motions in six degrees of freedom.
 - Fluid velocity on user-specified body panels.
 - Fluid pressure on user-specified body panels.
- Time-domain simulations with nonlinear hydrostatics in (pseudo-)random seas from a single heading angle with the following output:
 - Body motions in six degrees of freedom.
 - Fluid velocity on user-specified body panels.
 - Fluid pressure on user-specified body panels.
- Transformations to the frequency domain with the following output:
 - Hydrodynamic coefficients.
 - Response-amplitude operators for body motions.
 - Response-amplitude operators for the velocity on each panel.
 - Response-amplitude operators for the pressure on each panel.
 - The mean second-order force and moment on the body.
- Generalized or user-defined modal analysis to produce:
 - Impulse-response functions.
 - Response-amplitude operators.

2 Program Overview

There are two main modules in the TIMIT suite: **potent**, which solves the canonical problems for the velocity potentials and the source strengths, and uses the source strengths to compute the gradients of the potential; and **respon**, which uses these potentials and source strengths to evaluate the hydrodynamic quantities, perform simulations, *etc*, in both the time and frequency domains.

potent solves equation (24) as the potential formulation, and (28) as the source formulation. These are both integral equations involving only the submerged body surface (see §7). The input required by this module is the wetted body surface geometry in the *Geometric Data File* or GDF, and a set of parameters that determines which problems will be solved, and how they will be solved, in the *Potential Control File* or PCF. If m -terms corresponding to other basis flows than that for the Neumann-Kelvin linearization are desired then these must be provided in the *Basis Flow File* or BFF. See §4 for the specific contents and formatting of these files.

The output from **potent** is the binary file TIMIT.P2R. This file contains the input to **potent** and the solutions of the canonical hydrodynamic problems for further processing by **respon**. The output from **potent** is typically the value of the potential, the source strength, and the gradient of the potential, on each panel at each time step for each problem solved. See §5 for the format of this file.

respon uses the body geometry and the hydrodynamic quantities computed by **potent** to evaluate impulse-response functions, frequency-domain hydrodynamic coefficients, and simulated global and local quantities like motions and fluid velocity or pressure (see §7). Therefore, input required by this module is a set of parameters that determines which quantities to compute in the *Response Control File* or RCF. If the equation of motion is solved then **respon** also requires information about the body's distribution of mass and external damping or stiffness (if present) in the *Body Property File* or BPF. If a transient simulation of the body motions in waves is requested, then a time history of the incident wave elevation at the origin of the body-fixed coordinates must be provided in the *Incident Wave File* or IWF. If these simulations are to be undertaken using nonlinear hydrostatics, then a complete geometric description of the body must be provided in the *eXtended Geometric File* or XGF. See §4 for the specific contents and formatting of these files.

The output from **respon** is a set of ASCII text files containing the computed quantities. The hydrostatic coefficients are in one file, while a separate file will be created to contain each of the requested hydrodynamic quantities. Frequency-domain quantities are output in files which are formatted in the same manner as WAMIT[®] output files. See §5 for the definitions of the computed quantities and the file naming convention.

The **potent** module performs the dominant computational burden of the linear, transient, hydrodynamic analysis. The discretized integral equations require the solution of a linear system with multiple right-hand sides (a new right-hand side at each time step). In typical problems it is the generation of the right-hand sides by convolution that consumes almost all of the computational effort.

3 Getting Started

3.1 Portability and Coding Conventions

- A UNIX environment is assumed. Although a UNIX or UNIX-like operating system is assumed, we make every effort to accommodate others such as MS-DOS and VMS. The distribution of T̄MIT is typically in the form of a `tar` file and is most easily made available via `ftp`. We can provide other configurations if you require them.
- A makefile is provided. This file controls how modules are made and contains instructions for verifying your distribution. You will have to edit a few lines in this makefile so that it conforms to your particular environment. If you do not work in a UNIX-like environment, the makefile may be used as a guide to writing your own MS-DOS macros or VMS command files to accomplish the same things. See §3.2, 3.4, and 3.5.
- The code has been written in FORTRAN. T̄MIT was originally written in FORTRAN-77, but version 4.0 contains FORTRAN-90/95 programming, so it must be compiled with a FORTRAN-90/95 compiler.
- There are reserved file names. The following file names are reserved: `EVALPV.DAT` is a data file for use in the evaluation of the transient free-surface Green function; `VGTN98.BNY` is a data file for use in the evaluation of the transient free-surface Green function when the depth is finite; `TIMIT.P2R` is the binary file that contains the results of running the `potent` module; and `FNAMES.TD` is the file that stores the names of the input files for both the `potent` and `respon` modules.
- Variables are initialized to zero as needed. It is not assumed that your compiler zeroes any scalars or arrays of any type. The code is in single precision (4 byte words) with the exception of a few variables in the subroutines used to compute the incident potential in the file `iphi.f`. There are some complex variables in these routines, as well as in the section of the response computation module which handles frequency-domain quantities.
- Vectorization is assumed. The code has been written with vectorizing compilers in mind. For example, the code vectorizes well when compiled aggressively under the Cray compiling system. However high performance will not be achieved on the Cray, or other machines which use eight byte words as the default, if `DOUBLE PRECISION` declarations and double precision functions are not demoted to single precision.
- Runs are time-stamped. In the file `dattim.f`, the routine `DATTIM` calls the FORTRAN-90/95 date and time routine. This routine uses 4 digits to report the year (see the next item).
- Y2K is not an issue. T̄MIT does not use the year returned by date functions for any other purpose besides time-stamping, so it does not matter whether two or four digits are used for its specification.
- Linear system solutions are computed by routines in the LAPACK libraries. These are the fastest and most robust dense matrix factoring routines readily available. [The factoring is done once for many right-hand sides, so iterative solvers are not appropriate for T̄MIT .]

Modules of TIMIT and the makefile must be edited so that TIMIT is linked to the LAPACK routines that are available on your system. If your system does not have the LAPACK libraries they may be obtained without charge over the Internet from netlib. To learn about netlib and how to obtain codes, send email to Oak Ridge National Laboratory. The address is: `netlib@ornl.gov`. Use “send index” as the subject, with a null message body. You will receive the index of netlib codes and instructions on how to obtain them.

- Output data is usually written in the subroutines in which it is computed. We do not (with some exceptions) use separate subroutines for the computation and the output of quantities. A quantity is computed in a subroutine and then a file is opened and the quantity is written to that file. We do not return from subroutines without closing open files except in the case of VTGN98.BNY the binary data file used in computing the finite depth Green function.
- System errors on I/O are not trapped. If there is an I/O error, the error message you see will be generated by your system, not TIMIT. We think that the system error messages are most useful.
- Old output files are not protected from over-writing. The `potent` module writes the output from the solution of the hydrodynamic problems in a file called `TIMIT.P2R`. This file is opened with `STATUS='UNKNOWN'` so that if the file exists from a previous run it will be over-written *and the previous data will be lost*. After running `potent` it is the user's responsibility to rename the `TIMIT.P2R` file.
- Error messages are handled where errors occur. When errors are trapped, messages are provided to the user via the statement `PRINT *`, which always appears in the code where the error is trapped. See §6.11.

3.2 The TIMIT Distribution

All necessary components of TIMIT are distributed in the form of a compressed tar file which will have a name like, `timit-1.0-22nov94.tar.Z`, where the version number and release date are incremented from time to time.

This file must be uncompressed and untarred. The commands for this are as follows:

```
your-prompt> uncompress timit-1.0-22nov94.tar.Z
```

```
your-prompt> tar xf timit-1.0-22nov94.tar
```

The `tar` extraction from the directory `./` will create a directory tree with the top directory `./timit`. Under this directory will be the directories `./timit/doc`, `./timit/verify`, `./timit/util`, and `./timit/src`. The makefile in the distribution assume that the files remain in this tree, so files should not be moved after the distribution is unpacked else you will have to edit the makefile to correct the paths.

The TIMIT makefile is not provided in a form which you can directly use because your system is likely to have compiler names, switches, and library names we cannot anticipate. The lines of the makefile you will need to edit are described in §3.3.

The directory `timit` contains the makefile and the subdirectories of documentation and code.

The subdirectory `timit/verify` contains sample input and output so that you may confirm that your executables are performing as they should. This output consists of impulse-response functions for a Wigley hull discretized by 32 panels. It is for verifying the code only; *we do not imply that this discretization is adequate for hydrodynamic analysis*. The input and output files found in `timit/verify` have names that conform to the typical (but not in general, mandatory) T̄MIT file naming conventions, so see §4 and 5 for descriptions of these files. There is also a file in the subdirectory `timit/verify` that is used to verify that your T̄MIT installation has been successful. This is `vertim.f`, a FORTRAN program that performs part of the verification process. See §3.4 for instructions for verifying the installation.

The subdirectory `timit/doc` contains this documentation in the form of PostScript files. `UMan.ps` is the PostScript file with formatting for 8.5 × 11 inch paper and `UManA4.ps` is the PostScript file with formatting for A4 paper.

The subdirectory `timit/src` contains the source files for the T̄MIT modules. The following files will be found in `timit/src`:

`tdpot.f` : The main line and routines unique to the `potent` module.

`tdres.f` : The main line and routines unique to the `respon` module.

`mdfp.f` : Routines used to compute frequency-domain quantities in the `respon` module.

`dattim.f` : Routine used for applying time and date stamping to output.

`EVALPV.DAT` : Include file containing data necessary for the calculation of the transient free-surface Green function.

`geom.f` : Routines used to compute all panel specific quantities and hydrostatic coefficients.

`iphi.f` : Routines used to compute the incident potential and its derivatives.

`pret.d.f` : Preprocessor utility which determines memory allocation for `potent` and `respon`.

`rpan.f` : Routines used to compute Rankine Green function coefficients.

`tdio.f` : Routines which are common to both the `potent` and `respon` modules.

`vtgn98.f` : Routines used to compute the time-domain free surface Green function for infinite and finite depth.

`vtgn98bn.f` : This is a program that you will use to create your own binary data for the routines in `vtgn98.f` to use to compute the time-domain free surface Green function if the depth is not infinite.

The subdirectory `timit/util` contains the source files for T̄MIT utilities. These are programs which we find useful in our own hydrodynamic analyses and we anticipate that you will also. While we believe these codes provide correct results, strictly speaking we do not support them. The following files will be found in `timit/util`:

`if-file-reader.f` : Utility used to read frequency-domain output files created by T̄MIT or WAMIT[®] and reorder the data for plotting.

`genft.f` : Utility used to take Fourier transforms.

`dblbody.tar` : Utilities used to create double-body basis flow BFF's and GDF's.

`wigley.f` : Utility used to create Wigley hull form GDF's.

`sphgen.f` : Utility used to create spheroid GDF's.

`cylgen.f` : Utility used to create cylinder GDF's.

3.3 How to Install T̄MIT

This section describes how to install T̄MIT on your system. Details are provided for each of the installation steps, which are the following:

1. Unpacking the archive.
2. Editing the makefile `Makefile` for the compilation settings.
3. Creating the binary data in `VTGN98.BNY` for finite depth computations.

3.3.1 Unpacking the archive

All necessary components of T̄MIT are distributed in the form of a compressed tar file which will have a name like, `timit-1.0-22nov94.tar.Z`, where the version number and release date are incremented from time to time. This file must be uncompressed and untarred. The commands for this are as follows:

```
your-prompt> uncompress timit-1.0-22nov94.tar.Z
your-prompt> tar xf timit-1.0-22nov94.tar
```

The `tar` extraction will create a directory tree under the directory in which you perform the extraction. The root of this tree is `timit`. Under this directory will be the directories `timit/doc`, `timit/verify`, `timit/util`, and `timit/src`. (See §3.2.)

3.3.2 Editing the makefile

The makefile `timit/Makefile` must be edited to suit your computing environment. The lines you will need to edit are:

```
LINLIB = linlib
CNAME = cname
CFLAGS = cflags
LFLAGS = lflags
```

LINLIB : Change `linlib` to the switch that will link your LAPACK library to the T̄MIT objects. For instance on DEC Alpha workstations under DEC UNIX this would be `-ldxml`.

CNAME : Change `cname` to the name of your FORTRAN 90/95 compiler. We assume that invoking a single compiler will handle both FORTRAN-77 and FORTRAN-90/95. If you do not have such a compiler, you may need to change other compiling instructions in the makefile.

CFLAGS : Change `cflags` to the compiler flags you wish to invoke. We suggest `-c`, plus whatever flag invokes a high level of optimization (`-O` for many compilers).

LFLAGS : Change `lflags` to the linker flags you wish to invoke.

If you have moved files from the default directory tree, you will also wish to edit the paths to the various directories where we assume certain files to live.

SRCDIR : This variable should be set to the directory path to the source files (`./src` by default).

OBJDIR : This variable should be set to the directory path to the object files (`./src` by default).

INSTALLDIR : This variable should be set to the directory path to the executables (`$HOME/bin` by default).

3.3.3 Creating the binary data in `VTGN98.BNY`

The Green function evaluation routine requires a large data set for computations in finite depth. These data are read from `VTGN98.BNY`. Before you can make finite depth calculations you will have to create this data with the utility `/timit/src/vtgn98bn.f`. This program will ask you for the maximum nondimensional time you wish to compute to and then will proceed to compute the data. This can be quite time consuming, but needs to be done only once or occasionally.

The nondimensional time is defined by:

$$\tilde{T} = t\sqrt{\frac{g}{h}}$$

Compute the maximum value of \tilde{T} you anticipate needing and then run `vtgn98bn` as follows:

```
your-prompt> cd src
your-prompt> f77 -o vtgn98bn vtgn98bn.f      (your command may differ)
your-prompt> ./vtgn98bn
your-prompt> mv VTGN95.BNY ../
your-prompt> cd ..
```

For $\tilde{T} = 100.$, the runtime to create `VTGN95.BNY` on a 433MHz workstation is about 78 hours and the size of the file is 4.4Mb. The file created in this way (`VTGN98.BNY`) must be resident in the directory in which you perform finite-depth calculations unless you edit the `OPEN` statement in `vtgn98.f` to provide the complete path to this file.

Upon completion of these tasks you should have the new file `Makefile` in `timit`, which is the correct makefile for your environment, and a new data file `timit/VTGN98.BNY`, which contains the data for finite depth computations. To use `VTGN98.BNY` with a `TIMIT` run to make finite depth calculations, move this file to the directory in which you are running `TIMIT`.

3.4 Verifying Your `TIMIT` Distribution

Verification of your distribution is an important first step in using `TIMIT`. Not only will this confirm that you have correctly installed the distribution, it will also provide your first hands-on experience with the modules `potent` and `respon`. The distribution provides sample input and output in the directory `timit/verify` as indicated in §3.2. Running the `TIMIT` executables

using the input in this directory should produce output which is consistent with that found in the output files in this directory. By consistent we mean that the output will differ within a small tolerance. The tolerance in the verification check is set at 0.5E-05. The computations are sufficiently robust that different machines should generate output that passes this test. When failures are detected, the failing data are reported to `stdout` so that the severity of the failure may be evaluated.

You should undertake the verification procedure as soon as you have installed T̄MIT . The verification procedure is invoked by the target `verification` in the makefile. The command to verify the installation of T̄MIT is issued in the directory `timit` and is:

```
your-prompt> make verification
```

When this make target is invoked, you will see the salutations from `potent` and `respon` as they execute in succession. After T̄MIT runs, the FORTRAN utility (`./verify/vertim.f`) will be run to analyze the data. This utility will inform you of the success or failure of the verification.

3.5 Creating and Running the T̄MIT Executables

This section describes how to create and run the T̄MIT executables. Details are provided for each of the steps, which are the following:

1. Using `make` to create executables `potent` and `respon`.
2. Running `potent` and `respon`.

3.5.1 Creating T̄MIT executables

Compilation of T̄MIT is accomplished by using the command `make` from the directory `timit`. The makefile which `make` will read must have been edited for your environment during installation. (See §3.3.)

To make one or more T̄MIT modules the commands are as follows:

```
your-prompt> make target
```

where *target* may be one of the following:

`potent` : compiles and links the `potent` module only.

`respon` : compiles and links the `respon` module only.

`timit` : both `potent` and `respon` are compiled and linked.

There are no make targets provided for utilities. These codes are provided for your convenience; they are not supported T̄MIT programs.

3.5.2 Running `potent` and `respon`

To run the `potent` module, the command is `potent`. To run the `respon` module the command is `respon`. There are no command-line arguments. Both of these modules output informative and documentary information to `stdout`. For background jobs it is convenient to redirect this

output to a journal file. To run `potent` with output to `stdout` redirected to be appended to `timit.jou`:

```
your-prompt> potent >> timit.jou &
```

To run `potent` with output to `stdout` directed both to `stdout` and to be written to `timit.jou`:

```
your-prompt> potent | tee timit.jou &
```

The appropriate commands when running `respon` are similar.

4 Input to T_IMIT

Three input files are required to run the modules `potent` and `respon`. One of these, the *Geometric Data File*, referred to as the GDF, describes the geometry of the body under analysis. This geometry is required for both modules, but the GDF is read by the `potent` module only because that module passes the geometric information (in the file `TIMIT.P2R`) to the `respon` module along with the solutions to the hydrodynamic problems. Each module also has its own control file which contains the parameters that determine the type of analysis to be conducted. These are the *Potential Control File*, referred to as the PCF, for `potent` and the *Response Control File*, referred to as the RCF, for `respon`. The `potent` module has one optional input file the *Basis Flow File*, referred to as the BFF, which is used to input the gradients of the steady flow field and the m -terms on each body panel. This file is discussed in §4.2. Three optional input files are associated with solving the equations of motion in the `respon` module. When the equations of motion are solved (in either the time or the frequency domain) the *Body Property File*, referred to as the BPF, is used to input information about the body's distribution of mass. When a simulation is to be performed, the *Incident Wave File*, referred to as the IWF, is used to input the time history of the incident wave elevation (or three time histories in following seas, $U > 0$) at the origin of the body-fixed coordinates. If the exact hydrostatic and inertia forces are to be included in the simulation then the *eXtended Geometry File*, referred to as the XGF, is used to input the complete body geometry up to the sheer line. These three files are discussed in §4.5. Any of the optional input files may be omitted when they are not needed.

The input files may have any names which are legal in your operating system provided that they are not longer than twenty characters. A master file, the *File-names List File*, referred to as the FLF, with a reserved name: `FNAMES.TD`, is used to specify the names of the primary input files, while the optional input file names are specified in the primary input files.

All input is read with free format, so data items must be separated by at least one blank space. Line breaks are necessary where shown, and unnecessary where noted. Additional line breaks should have no effect. Each of these files is described in detail in the following sections.

4.1 File-names List File, FLF

The FLF specifies the input files described in §4.2 and 4.5. The FLF has the reserved name `FNAMES.TD`. T_IMIT *will not run without this file*. (This reserved name may be redefined by the user by editing the `DATA` statements and error messages in the source files.) The FLF `FNAMES.TD` contains a list of the primary input file names which specify the parameters to run `potent` (PCF), and those to run `respon` (RCF). For example, the contents of `FNAMES.TD` might be:

```
froude3.pcf
froude3.rcf
```

The extensions used in the example are a useful choice to help manage files, but they are not required. The order of the file names: PCF, RCF; *is* required. Additional input files, whether required or optional, are specified in the PCF and RCF.

4.2 The Potential Control File, PCF

The PCF is used to place the body coordinates in the earth-fixed system, to define the steady forward speed, and to determine which problems are to be solved by the `potent` module. The data in the PCF should be input as follows (where we have used the variable names internal to TIMIT to identify quantities):

```
HEAD
IPOT ISOR IVEL IGREEN U HBOT
NBODY
FILEN(1)
XBODY(1) XBODY(2) XBODY(3) XBODY(4)
IRAD IDIF
MODE(1) MODE(2) MODE(3) MODE(4) MODE(5) MODE(6) GENMOD
DELT MAXTR MINTD MAXTD
NBETA
BETA(1) BETA(2) ... BETA(NBETA)
INTERM
FILEN(7)
```

The definition of each quantity in the PCF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

IPOT determines whether the potential formulation is solved.

IPOT = 0: The potential formulation is not solved.

IPOT = 1: The potentials are computed using the potential formulation.

ISOR determines whether the potentials are computed from the source formulation.

ISOR = 0: The potentials are not computed using the source formulation.

ISOR = 1: The potentials are computed using the source formulation. **Note:** If **IPOT** is set to 1, then the potentials are computed from the potential formulation, regardless of the value of **ISOR**.

IVEL determines whether the three components of fluid velocity on each body panel are calculated from the source strengths. **Note:** When $U > 0$, the x -component of velocity is always computed.

IVEL = 0: The velocities are not calculated.

IVEL = 1: The velocities are calculated.

IGREEN controls whether the Green function coefficients are stored for re-use or recomputed. The trade-off of disk space versus processing time is discussed in §6.

IGREEN = 0: Calculate the Green function coefficients once and store them in scratch files to be re-used throughout the run.

IGREEN = 1: Recalculate all of the Green function coefficients at every time step.

U is the steady forward speed of the body in units which are consistent with **ULEN**.

HBOT is the water depth in units which are consistent with **ULEN**.

HBOT < 0: The depth is infinite.

HBOT > 0: The depth is finite as specified.

NBODY is the number of bodies to be analyzed. **Note:** Only **NBODY = 1** is supported.

FILEN(1) is the name of the GDF for the body to be analyzed.

XBODY(I) determines the location of the body-fixed coordinate system with respect to the earth-fixed coordinate system. (See §4.3)

XBODY(I), I=1,2,3 are the x_0 , y_0 , and z_0 coordinates in the earth-fixed system of the origin of the body-fixed coordinates.

XBODY(4) is the angle from the positive x_0 -axis in the earth-fixed system to the positive x -axis in the body-fixed system, with the positive sense defined by the right-hand rule.

IRAD, IDIF are indices used to specify whether the radiation and diffraction problems are to be solved. The following options are available:

IRAD = 1: Solve the complete radiation problem.

IRAD = 0: Solve the radiation problem for those modes which are specified by non-zero values in the array **MODE** (see below).

IRAD = -1: Do not solve the radiation problem.

IDIF = 1: Solve the diffraction problem.

IDIF = 0: Solve only those parts of the diffraction problem which are necessary to obtain the exciting forces in the modes which are specified by non-zero values in the array **MODE** (see below). **WARNING!** Setting **IDIF = 0** economizes the problem for the computation of global quantities because only the symmetric or anti-symmetric parts of the problem are solved as required for the forces. This means that there is insufficient information to compute local quantities.

IDIF = -1: Do not solve the diffraction problem.

MODE(I) is a six-element array of indices, where **I=1,2,3** correspond to the surge, sway and heave translational modes in the body-fixed (x, y, z) axes, and **I=4,5,6** to the roll, pitch and yaw rotational modes about the same axes, respectively. Each of these six indices should be set equal to 0 or 1, depending on whether the corresponding radiation mode(s) and diffraction component(s) are required. (See the options **IRAD=0** and **IDIF=0** above.)

The MODE array specifies the modes for which body motions and exciting forces are to be evaluated. If the body has symmetry planes, the complete diffraction problem is decomposed into symmetric/antisymmetric components in a manner which permits the most efficient solution. Only those components of the diffraction potential required to evaluate the exciting force for the specified modes are evaluated. For example, if ISX=1, IDIF=0, MODE(1)=1, and the remaining elements of MODE are set equal to zero, then the only component of the diffraction potential which is solved is that part which is antisymmetric in x .

GENMOD is the number of generalized modes specified by the subroutine DEFINE for which analysis is desired. **Note:** If generalized modes are calculated then ULEN should be set to 1. Otherwise it will be difficult to interpret the output as TIMIT has no way to correctly apply the power of ULEN used in normalizing user defined modes.

DELT, MAXTR, MINTD, MAXTD are the the time step size, the maximum time for the radiation problem, and the minimum and maximum times for the diffraction problem; all in *seconds*. The same time step size is used for both the radiation and the diffraction problems.

NBETA is the number of incident wave headings.

BETA(i) i=1, NBETA are the incident wave heading angles in degrees measured from the positive x_0 -axis of the earth-fixed coordinate system. $\beta = 180^\circ$, corresponds to head seas, and $\beta = 0^\circ$ to following seas.

IMTERM is a flag to indicate whether to read and use the basis flow velocities, second derivatives, and m -terms listed in the BFF. This flag is optional and may be left out of the PCF with the same result as setting it to zero.

IMTERM = 0: Use the undisturbed free-stream velocities and the Neumann-Kelvin m -terms.

IMTERM = 1: Read the basis flow velocities, second derivatives, and the m -terms at every panel centroid from the BFF listed on the subsequent line.

FILEN(7): The name of the optional BFF from which to read the basis flow velocities and m -terms. This can be any 20 character file name, or it can be left out if IMTERM = 0 or IMTERM is missing.

4.3 The Geometric Data File, GDF

The GDF contains the geometric description of the body as an ensemble of surface panels which may be quadrilaterals or triangles. The GDF transmits this representation to the program through: indicators of scale and units, a pair of symmetry indices, the total number of panels, and the Cartesian coordinates of the vertices of each panel. TIMIT uses two right-handed Cartesian coordinate systems to define the problem. The *earth-fixed coordinate system* (x_0, y_0, z_0) defines the free-surface by the $z_0 = 0$ plane, and the direction of the body's steady forward motion as the positive sense of the x_0 -axis. The *body-fixed* or *body* coordinate system (x, y, z) is that in which the panels are defined. The body coordinate system must have the

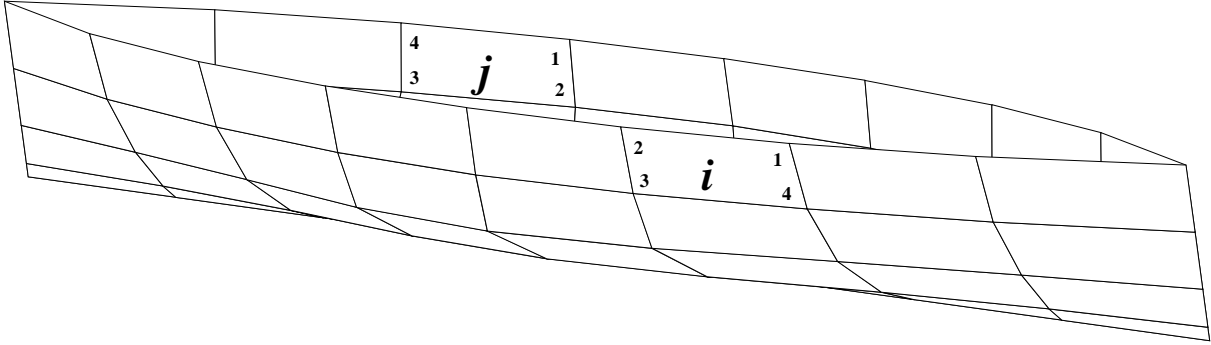


Figure 1: Discretization of a Wigley hull showing the convention for panel vertex numbering. We are viewing panel i from *inside* the fluid domain, so the vertex ordering appears *anti-clockwise*. We are viewing panel j from *outside* the fluid domain, so the vertex ordering appears *clockwise*.

z -axis vertical (positive upward) and $z = 0$ must define a plane which is parallel to the free surface. This coordinate system, which is implied by the input, is the coordinate system in which the output from both **potent** and **respon** is defined. The position and orientation of the body coordinate system in the earth-fixed coordinate system is defined in the PCF.

Only the wetted surface of the body should be paneled, and then only half or a quarter of it, if there exist one or two planes of geometric symmetry respectively. If planes of symmetry are exploited, they can only be the planes $x = 0$ or $y = 0$. With forward speed, only port-starboard ($y = 0$ plane) symmetry may be exploited. Exploitation of geometric symmetry is not required, but since the some of the computational cost is proportional to the square of the number of panels, it is unwise not to.

When the body has forward speed, T̄MIT must identify the rows of panels adjacent to the waterlines. Some discretization error is allowed for in the identification of these waterline panels; however, the top edge of these panels must lie within 0.5% of the body's maximum draft

of the $z = 0$ plane.

The input vertices of a panel do not need to be co-planar. T̄MIT internally defines planar panels by projection of the input vertices onto the plane defined by the four midpoints of the four sides of the input panel. The extent to which a panel is not planar is referred to as *skew*.

The data in the GDF should be written as shown below, where we have used the variable names internal to T̄MIT to identify quantities:

```

HEAD
ULEN GRAV
ISX ISY
NP
X(1,1,1) X(2,1,1) X(3,1,1) X(1,2,1) X(2,2,1) ... X(3,4,1)
X(1,1,2) X(2,1,2) X(3,1,2) X(1,2,2) X(2,2,2) ... X(3,4,2)
.
.
.
X(1,1,NP) X(2,1,NP) X(3,1,NP) X(1,2,NP) X(2,2,NP) ... X(3,4,NP)

```

The definition of each quantity in the GDF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

ULEN is the dimensional length characterizing the body. This parameter corresponds to the quantity L used in §5 to non-dimensionalize the quantities output from T̄MIT. ULEN may be in any system of units, as long as all other input uses the same units. ULEN must be a positive number and greater than 10^{-5} .

GRAV is the acceleration of gravity in the same units as ULEN. *The units of time are always seconds.* If lengths are in meters or feet, GRAV should be 9.807, or 32.174, respectively.

ISX, ISY are the geometry symmetry indices which must take the integer values 0 or 1.

ISX = 1: The $x=0$ plane is a geometric plane of symmetry.

ISX = 0: The $x=0$ plane is not a geometric plane of symmetry.

ISY = 1: The $y=0$ plane is a geometric plane of symmetry.

ISY = 0: The $y=0$ plane is not a geometric plane of symmetry.

NP is the number of panels used to discretize the body or section of the body if symmetry is exploited.

X(i,j,n) is the i^{th} coordinate of the j^{th} vertex of the n^{th} panel. All forces and moments will be computed with respect to the coordinate system defined by these inputs. The connectivity of the panels is unimportant, but the four vertices of each panel must be listed consecutively so as to traverse the panel boundary in an *anti-clockwise* direction when viewed from the fluid domain as shown in Figure 1. Triangular panels should

be input as quadrilaterals with any two vertices coincident. Line breaks in the vertex coordinate list are arbitrary.

4.4 The Basis Flow File, BFF

The BFF allows the specification of steady basis flow velocities $\nabla\bar{\Phi}$, the m -terms m_k , $k = 1, 2, \dots, 6$, which to be used in the radiation body boundary condition, and in the evaluation of fluid pressures, and the derivatives of the basis flow velocities $\nabla\nabla\bar{\Phi}$ which are used to compute c_{jk}^0 (See §7.) These data are required only if the Neumann-Kelvin basis flow is not desired, and so this file is optional. The quantities in this file must be evaluated at the centroid of every panel used to discretize the body. The order in which the basis flow quantities are input in the BFF must correspond exactly to the order in which the panels are input in the GDF. **Note:** When this option is used, the body boundary condition and the fluid pressures will be linearized about the user-defined $U\bar{\Phi}$ while the free-surface condition is still linearized about $-Ux$.

The format of this file is as follows (where we have used the variable names internal to TIMIT to identify quantities):

```

HEAD
NP
(BASEVL(J,1),J=1,3) (MTERM(J,1),J=1,6) ((GRDVL0(J,K,1),J=1,K),K=1,3)
(BASEVL(J,2),J=1,3) (MTERM(J,2),J=1,6) ((GRDVL0(J,K,2),J=1,K),K=1,3)
.
.
.
(BASEVL(J,NP),J=1,3) (MTERM(J,NP),J=1,6) ((GRDVL0(J,K,NP),J=1,K),K=1,3)

```

The definition of each quantity in the BFF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

NP is the number of panels used to discretize the body and must be identical to the number appearing in the GDF.

BASEVL(J,I) are the x , y , and z -components of $\nabla\bar{\Phi}$ ($J=1,2,3$ respectively) at the centroid of panel I. The units must be consistent with those of the gravitational constant.

MTERM(J,I) are the values of m_j due to the user's defined $\bar{\Phi}$ at the centroid of panel I. The units must be consistent with those of the gravitational constant.

GRDVL0(J,K,I) are the second derivatives of the basis flow potential, $\bar{\Phi}_{jk}$ (with $J,K=1,2,3$ corresponding to x -, y -, and z - derivatives respectively) at the centroid of panel I. The units must be consistent with those of the gravitational constant.

4.5 The Response Control File, RCF

The Response Control File is used to determine what quantities are to be computed by the `respon` module. The data in the RCF should be input as follows (where we have used the variable names internal to TIMIT to identify quantities):

```
HEAD
ITOPTN(1) ITOPTN(2) ... ITOPTN(9)
NBETAH (BETAH(I), I=1,NBETAH)
NPL (PNLLST(I),I=1,NPL)
IFOPTN(I) IFOPTN(2) ... IFOPTN(9)
DLTFRQ MAXFRQ
FILEN(4)
FILEN(5)
FILEN(6)
```

The definition of each quantity in the RCF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

ITOPTN is an array of nine option indices, each of which must be set equal to 0 or 1, or in some cases, 2. These indices determine which quantities are to be evaluated. The i^{th} option is selected by setting $ITOPTN(i) > 0$, the i^{th} option is rejected by setting $ITOPTN(i) = 0$.

ITOPTN(1): Compute the time history of the calm water, first-order force and moment acting on the hull when it is impulsively accelerated to forward speed U . (The large time mean values of these transient quantities provide the steady resistance, sinkage force, and trim moment acting on the hull.) There is no flag for solving this problem in the PCF; it is always solved when $U > 0$.

ITOPTN(2): Determines how the radiation impulse-response functions are computed. The frequency dependent coefficients and the motions all depend upon the impulse-response functions, and the default method of computing the impulse-response functions is by direct pressure integration. So, for example, if $ITOPTN(2)=0$ and $IFOPTN(1)=1$ then the added-mass coefficients will be computed by Fourier transform of the impulse-response functions computed by direct pressure integration, but the impulse-response functions will not be output.

ITOPTN(2)=1: Compute the radiation impulse-response functions from the radiation potentials by a direct integration of the pressure over the body.

ITOPTN(2)=2: Compute the radiation impulse-response functions from the radiation potentials using Tuck's theorem.

ITOPTN(3): Determine the exciting force impulse-response functions from the solution of the radiation problem via the Haskind relations. This will produce impulse-response functions for incident wave headings $BETAH(i)$, $i = 1, NBETAH$. This

option may only be used for zero-speed analysis, as the forward-speed Haskind relations require reverse-flow radiation potentials not computed by `potent`. The Haskind relations are not exact for the forward-speed problem, anyway. If exciting force coefficients from the Haskind relations are desired (`IFOPTN(2)=1`), then `ITOPTN(3)` must be set to 1.

ITOPTN(4): Determines how the exciting force impulse-response functions are computed. The frequency dependent coefficients and the motions all depend upon the impulse-response functions, and the default method of computing the impulse-response functions is by direct pressure integration. So, for example, if `ITOPTN(4)=0` and `IFOPTN(3)=1` then the exciting force coefficients will be computed by Fourier transform of the impulse-response functions computed by direct pressure integration, but the impulse-response functions will not be output.

ITOPTN(4)=1: Determine the exciting force impulse-response functions from the solution of the diffraction problem by a direct integration of the pressure over the body. This will produce impulse-response functions for incident wave headings `BETA(i)` which are specified in the PCF. This is because the wave headings are required when the scattered potential problem is solved in `potent`.

ITOPTN(4)=2: Compute the exciting force impulse-response functions from the diffraction potential using Tuck's theorem.

ITOPTN(5): Perform a simulation of body motions in a (pseudo-)random seaway. The computed impulse-response functions and hydrostatic coefficients; the information about the distribution of the body's mass in the BPF; and a time history of an incident wave elevation in the IWF are used in a solution of the equations of motion. The solution provides the body displacement and velocity in six degrees of freedom. While hydrodynamic quantities are computed in `TIMIT` for an array of wave heading angles, the simulation is carried out at a single wave angle which is specified in the IWF. There must be hydrodynamic quantities at this wave angle. If there are, they will be found and used. If not, the simulation will abort with an error message.

ITOPTN(5)=1: Use the linearized hydrostatic coefficient matrix, linearized hydrodynamic quantities, and linearized equations of motion to determine the body motion.

ITOPTN(5)=2: Use the nonlinear hydrostatic coefficient matrix, linearized hydrodynamic quantities, and exact equations of motion to determine the body motion.

ITOPTN(6): If a simulation is run (`ITOPTN(5)>0`), then compute and output the fluid pressure on a selection of body panels. These panels are specified by `PNULLST` described below.

ITOPTN(7): If a simulation is run (`ITOPTN(5)>0`), then compute and output the fluid velocity on a selection of body panels. These panels are specified by `PNULLST` described below.

ITOPTN(8): Reserved for future use.

ITOPTN(9): Reserved for future use.

NBETAH is the number of incident wave headings to be used in determining exciting force impulse-response functions from the Haskind relations ($U = 0$ only). If impulse-response functions from the Haskind relations are not desired (ITOPTN(3)=0), then this line must not be present in the RCF.

BETAH(i) $i=1$, NBETAH are the incident wave heading angles in degrees (measured from the positive x_0 -axis of the earth-fixed coordinates) to be used in determining exciting force impulse-response functions from the Haskind relations. $\beta = 180^\circ$, corresponds to head seas, and $\beta = 0^\circ$ to following seas. If ITOPTN(3)=0 then this line must not be present.

NPL is the number of panels selected for the computation and output of fluid pressure and/or velocity. If ITOPTN(6)=0 and ITOPTN(7)=0 then this line must not be present. Versions 3.* and 4.* of TIMIT support $NPL < 9$, although it is straightforward to edit the code to increase this limit. Resetting NPLT in `tdres.f` is sufficient.

PNLLST(i) $i=1$, NPL are the indices of the panels selected for computation and output of fluid pressure and/or velocity.

IFOPTN is an array of nine option indices, each of which must be set equal to 0 or 1, or 2 or 3 in some cases. These indices determine which frequency dependent quantities are to be evaluated from the Fourier transforms of the canonical transient problems. The i^{th} option is selected by setting $IFOPTN(i) > 0$, the i^{th} option is rejected by setting $IFOPTN(i) = 0$. All of the following quantities are output at MAXFRQ/DLTFRQ frequencies running from $\omega_0 = 0$ to $\omega_0 = \text{MAXFRQ}$. (The four implemented options are identical to the IOPTN index which is used in the .FRC file for WAMIT/tm)

IFOPTN(1): Compute the added mass and damping coefficients as the Fourier transform of the radiation impulse-response functions.

IFOPTN(2): Compute the exciting force coefficients by taking the Fourier transform of the exciting force impulse-response functions which have been computed via the Haskind relations. This option may only be used for zero-speed analysis, as the forward-speed Haskind relations require radiation potentials not computed by `potent`.

IFOPTN(3): Determine the exciting force coefficients from the Fourier transform of the exciting force impulse-response functions computed from the diffraction potential.

IFOPTN(4): Compute the response-amplitude operator (RAO) for the body in all of the requested modes of motion. (The RAO is the ratio of the amplitude of the body's response in a particular mode of motion to the amplitude of the incident wave elevation.)

IFOPTN(4) = 0: Do not compute the RAO's.

IFOPTN(4) = 1: Compute the RAO's using the exciting forces found by the Haskind relations.

IFOPTN(4) = 2: Compute the RAO's using the exciting forces found from the diffraction potentials.

IFOPTN(5): Compute fluid pressure and/or velocity RAO's from the Fourier transforms of the pressure and velocity impulse-response functions and the RAO's for body motion.

IFOPTN(5) = 1: Determine the fluid pressure on each panel.

IFOPTN(5) = 2: Determine the fluid velocity on each panel.

IFOPTN(5) = 3: Determine the fluid pressure and the fluid velocity on each panel.

IFOPTN(6): Not supported.

IFOPTN(7): Not supported.

IFOPTN(8): Not supported.

IFOPTN(9): Compute the second-order steady force (and moment) acting on the body by integration of the fluid pressure.

DLTFRQ is the absolute frequency step size $\Delta\omega_0$ in units consistent with **ULEN** and **GRAV** at which the frequency domain quantities are desired. If no frequency domain options are selected then this line must not be present.

MAXFRQ is the highest absolute frequency in units consistent with **ULEN** and **GRAV** at which the frequency domain quantities are desired. **Note:** When $U > 0$ the frequency dependent coefficients (and the RAO's) computed by **TIMIT** are singular at the critical frequency of oscillation corresponding to $\tau = 1/4$ (see §7.14). When one of the desired frequencies coincides with the critical frequency (to machine precision) that frequency is shifted down by a non-dimensional frequency of 1.E-05 in order to produce finite results. All frequency domain output is evaluated over the range $0 < \omega_0 \leq \text{MAXFRQ}$ at intervals of **DLTFRQ**, except for the added-mass and damping coefficients for which **DLTFRQ** and **MAXFRQ** are interpreted as encounter frequency. If no frequency domain options are selected then this line must not be present.

FILEN(4) is the name (up to 20 characters) of the BPF. If either a time domain simulation or RAO computations are requested and the radiation problem has been solved, then the body mass distribution must be provided. The BPF is described in §4.6.

FILEN(5) is the name (up to 20 characters) of the IWF. If a time domain simulation is requested then the incident wave elevation must be provided. The IWF is described in §4.7.

FILEN(6) is the name (up to 20 characters) of the XGF. If a time-domain simulation with nonlinear hydrostatics and exact equations of motion is requested then the complete body geometry must be provided. The XGF is described in §4.8.

4.6 The Body Property File, BPF

The BPF is used to input the body's center of gravity, radii of gyration, and optionally, mass, damping and stiffness properties to the **respon** module. These data are required if the equations of motion are to be solved (i.e. when **ITOPTN(5) = 1** or **IFOPTN(4) = 1**) or if there are external

constraints or generalized modes, so this file is optional. The data in the BPF should be input as follows (where we have used the variable names internal to TIMIT to identify quantities):

```

HEAD
ICG CG(1) CG(2) CG(3)
R(1,1) R(1,2) R(1,3)
R(2,1) R(2,2) R(2,3)
R(3,1) R(3,2) R(3,3)
IMASS
EXMASS(1,1) EXMASS(1,2) ... EXMASS(1,MAXD)
EXMASS(2,1) EXMASS(2,2) ... EXMASS(2,MAXD)
.
.
.
EXMASS(MAXD,1) EXMASS(MAXD,2) ... EXMASS(MAXD,MAXD)
IDAMP
EXDAMP(1,1) EXDAMP(1,2) ... EXDAMP(1,MAXD)
EXDAMP(2,1) EXDAMP(2,2) ... EXDAMP(2,MAXD)
.
.
.
EXDAMP(MAXD,1) EXDAMP(MAXD,2) ... EXDAMP(MAXD,MAXD)
ISTIFF
EXSTIFF(1,1) EXSTIFF(1,2) ... EXSTIFF(1,MAXD)
EXSTIFF(2,1) EXSTIFF(2,2) ... EXSTIFF(2,MAXD)
.
.
.
EXSTIFF(MAXD,1) EXSTIFF(MAXD,2) ... EXSTIFF(MAXD,MAXD)

```

The definition of each quantity in the BPF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

ICG is a flag which indicates whether the horizontal position of the center of gravity of the body should be set equal to the horizontal position of the center of buoyancy or read from input.

ICG=0: The computed (in *potent*) horizontal position of the center of buoyancy **CB(1)** and **CB(2)** will be used as the horizontal location of the center of gravity.

ICG=1: The input values of **CG(1)** and **CG(2)** will be used as the horizontal location of the center of gravity.

CG(I) is the i-th component of the position vector of the center of gravity in body coordinates. **CG(3)** is always used to provide the gravitational contribution to the stiffness matrix. **CG(1)** and **CG(2)** will be used if **ICG = 1** (see **ICG** above). **CG** is dimensional and must be consistent with **ULEN**.

R(I,J), **I=1,2,3**, **J=1,2,3** is the three by three matrix of the body's radii of gyration in the body-fixed coordinate system. This quantity is dimensional and must be consistent with **ULEN**.

IMASS indicates whether the elements of a user specified mass matrix follow.

IMASS=0: No mass matrix elements follow.

IMASS=1: A **MAXD** by **MAXD** array of mass matrix elements follows, with **MAXD** equal to the total number of degrees of freedom.

EXMASS(I,J) is the j-th element in the i-th row of the external mass matrix with dimension **MAXD** by **MAXD**. For a single rigid body analysis **MAXD = 6**, while for an analysis with, say, 8 generalized modes, **MAXD = 14**. The elements of this mass matrix are *substituted* for the mass matrix based on the data above and defined in §7.1.

IDAMP indicates whether the elements of a user specified mass matrix follow.

IDAMP=0: No damping matrix elements follow. **Note:** The use of an external damping matrix is not supported so **IDAMP** must be zero.

IDAMP=1: Not supported.

EXDAMP(I,J) **Note:** The use of an external damping matrix is not supported.

ISTIFF indicates whether the elements of a user specified stiffness matrix follow.

ISTIFF=0: No stiffness matrix elements follow.

ISTIFF=1: A **MAXD** by **MAXD** array of stiffness matrix elements follows with **MAXD** equal to the total number of degrees of freedom.

EXSTIFF(I,J) is the j-th element in the i-th row of the external stiffness matrix with dimension **MAXD** by **MAXD**. For a single rigid body analysis **MAXD = 6**, while for an analysis with, say, 8 generalized modes, **MAXD = 14**. The elements of this stiffness matrix are *added to* the hydrostatic, gravitational, and hydrodynamic stiffness matrix **C** defined in §7.2.

If any of **IMASS**, **IDAMP**, or **ISTIFF**, are nonzero, then they all must be present although any can be zero and consequently not followed by any matrix elements. Alternatively the file may be terminated after the ninth radius of gyration value.

4.7 The Incident Wave File, IWF

The IWF is used to input to the **respon** module a time history of the wave elevation (in units consistent with **ULEN**) at the origin of the body-fixed coordinate system. These data are required only if the equations of motion are to be solved in the time domain (i.e. when

ITOPTN(5)>0) so this file is optional. When the body has a non-zero forward speed and the waves are incident from abaft the beam, ($-90^\circ < \beta < 90^\circ$) then the incident wave elevation must be split into three pieces as discussed in §7.11. **Note:** The elevation must be specified at the origin of the body-fixed coordinate system, which is a moving coordinate system relative to the earth-fixed frame in a forward speed problem. The data in the IWF should be input as follows (where we have used the variable names internal to TIMIT to identify quantities):

```

HEAD
U BETA DLTSIM
ZETA(0)
ZETA(1)
.
.
.
ZETA(MAXSIM)

```

The definition of each quantity in the IWF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

U is the steady forward speed of the body.

BETA is the heading angle in degrees of the incident wave system with respect to the positive x_0 -axis of the earth-fixed coordinates.

DLTSIM is the time step size in seconds of the incident wave record.

ZETA(i), i=0,1,...,MAXSIM is the incident wave elevation at the origin of the body-fixed coordinate system for each of MAXSIM+1 time steps. When $U > 0$ and $-90^\circ < \beta < 90^\circ$, the incident wave must be decomposed into three pieces $ZETA_1(t)$, $ZETA_2(t)$, and $ZETA_3(t)$ as discussed in §7.11.

When $U > 0$ and $-90^\circ < \beta < 90^\circ$, the three incident wave elevations should be input three to a line.

```

header
U BETA DLTSIM
ZETA1(0) ZETA2(0) ZETA3(0)
ZETA1(1) ZETA2(1) ZETA3(1)
.
.
.
ZETA1(MAXSIM) ZETA2(MAXSIM) ZETA3(MAXSIM)

```

4.8 The eXtended Geometry File, XGF

The XGF is used to input the complete surface description of the body over which the hydrostatic pressure may be integrated. These data are required only for simulations with nonlinear hydrostatic forcing and exact equations of motion (*i.e.* ITOPTN(5) = 2), so this file is optional. The XGF panels are displaced by finite translations and rotations during a simulation to find the *instantaneous* position of the body. For this reason, the XGF must include panels up the the sheer line of the body. The Cartesian coordinates of the XGF panel vertices imply an XGF-body coordinate system. This body frame need not coincide with the coordinate system implied by the GDF panels. The z -axis, however, must be vertical (positive upward) and $z = 0$ must define a plane which is parallel to the free surface. The position and orientation of the XGF body coordinate system relative to the earth-fixed system is defined in the XGF itself.

The data in the XGF should be input as follows (where we have used the variable names internal to TIMIT to identify quantities):

```

HEAD
ULENS GRAVS
XBODYS(1) XBODYS(2) XBODYS(3) XBODYS(4)
ISXS ISYS
NPS
XS(1,1,1) XS(2,1,1) XS(3,1,1) XS(1,2,1) XS(2,2,1) ... XS(3,4,1)
XS(1,1,2) XS(2,1,2) XS(3,1,2) XS(1,2,2) XS(2,2,2) ... XS(3,4,2)
.
.
.
XS(1,1,NPS) XS(2,1,NPS) XS(3,1,NPS) XS(1,2,NPS) XS(2,2,NPS) ... XS(3,4,NPS)

```

The definition of each quantity in the XGF is as follows:

HEAD denotes a one-line character header dimensioned CHARACTER*72. This line is available for the user to insert a brief description of the file.

ULENS is the dimensional length characterizing the XGF body. ULENS may be in any system of units, as long as all other XGF input uses the same units. ULENS must be a positive number and greater than 10^{-5} .

GRAVS is the acceleration of gravity in the same units as ULENS. *The units of time are always seconds.* If lengths are in meters or feet, GRAVS should be 9.807, or 32.174, respectively.

XBODYS(I), I=1,2,3 are the x_0 , y_0 , and z_0 coordinates in the earth-fixed system of the origin of the XGF body-fixed coordinates.

XBODYS(4) is the angle from the positive x_0 -axis in the earth-fixed system to the positive x -axis in the XGF's body-fixed system, with the positive sense defined by the right-hand rule.

ISXS, ISYS are the geometry symmetry indices which must take the integer values 0 or 1.

ISXS = 1: The $x=0$ plane is a geometric plane of symmetry.

ISXS = 0: The $x=0$ plane is not a geometric plane of symmetry.

ISYS = 1: The $y=0$ plane is a geometric plane of symmetry.

ISYS = 0: The $y=0$ plane is not a geometric plane of symmetry.

NPS is the number of panels used to discretize the body or section of the body if symmetry is exploited.

XS(i,j,n) is the i^{th} coordinate of the j^{th} vertex of the n^{th} panel. The connectivity of the panels is unimportant, but the four vertices of each panel must be listed consecutively so as to traverse the panel boundary in an *anti-clockwise* direction when viewed from the fluid domain as shown in Figure 1. Triangular panels should be input as quadrilaterals with any two vertices coincident. Line breaks in the vertex coordinate list are arbitrary.

4.9 The Generalized Mode File, `defmod.f`

The file `defmod.f` is not an input file. `defmod.f` is a FORTRAN source file giving the user control of the definition of generalized modes. This source file contains the subroutine **DEFINE**, in which the user may code the definitions of quantities for performing generalized modal analysis of (for example) bending of structures, multiple bodies, or multiple rigid bodies connected by compliant couplings. In the file `defmod.f` provided with the distribution, there are modes defined for the longitudinal bending of a structure. These are the so-called “dry modes” of a beam. If the input variable **GENMOD** (see §4.2) is greater than 0, TIMIT will use the information in `defmod.f`. See §6.8 for more on this topic.

5 Output from T̄MIT

The quantities evaluated by T̄MIT are output in non-dimensional form in terms of the appropriate combinations of water density ρ , the acceleration due to gravity g , body velocity U , body length L and, in the case of frequency-domain quantities, the amplitude of the incident wave $|\mathcal{A}|$. The definition of the hydrostatic quantities, and their non-dimensionalization is provided in §5.1; while the non-dimensionalization of the hydrodynamic quantities appears in §5.2 and 5.3.

A complete definition of the hydrodynamic quantities can be found in §7; however, the context which illuminates the role of these quantities is the following. We typically solve the equation of motion for a body in a seaway in the form of equation (1), forced by the right-hand side defined by (3). The impulse-response functions are composed of the coefficients and kernels which appear in these equations. If we take the incident wave to be time-harmonic at the frequency of encounter ω then, in the limit as $t \rightarrow \infty$, the motions of the body also become time harmonic (see equation (81)) and the impulse-response functions can be related to a complementary set of frequency-response functions through Fourier transforms. These, then, appear in the frequency-domain equation of motion (82).

The output from `potent` is the binary file `TIMIT.P2R` (a reserved name) containing the potential values, and/or velocities on each panel at each time step for each problem solved. This file is read by the `respon` module. The `respon` module performs the appropriate calculations and writes the ASCII-text output files of hydrostatic data and hydrodynamic quantities.

5.1 Hydrostatic Quantities

All hydrostatic data can be expressed as surface integrals over the body wetted surface S_B , by virtue of Gauss' divergence theorem as is shown in §7.2. The non-dimensionalization of these quantities is as follows:

The body volume is

$$\bar{\mathcal{V}} = \frac{\mathcal{V}}{L^3}.$$

The surface area is

$$\bar{S} = \frac{S}{L^2}.$$

The coordinates of the center of buoyancy are

$$\bar{x}_b = \frac{x_b}{L}, \quad \bar{y}_b = \frac{y_b}{L}, \quad \bar{z}_b = \frac{z_b}{L}.$$

The matrix of hydrostatic coefficients is

$$\bar{C}_{ij} = \frac{C_{ij}}{\rho g L^n},$$

where $n = 2$ for $i = j = 3$; $n = 3$ for $i = 3, j = 4, 5$; and $n = 4$ for $i = 4, 5, j = 4, 5, 6$.

5.1.1 The format of the hydrostatic output

The hydrostatic data is written to the file `static.dat`. This file begins with four lines of information that identify the T̄MIT run by the run date- and time-stamp and the headers from the GDF, PCF, and RCF. The hydrostatic output includes computed hydrostatic quantities and input mass distribution quantities. All of the output is non-dimensional and headers appear in the file to identify the data. The hydrostatic data following the four lines of headers is organized as follows:

$$\begin{aligned}
&(\bar{C}(1, k), k = 1, 2, \dots, 6) \\
&(\bar{C}(2, k), k = 1, 2, \dots, 6) \\
&\quad \cdot \quad \cdot \quad \cdot \\
&\quad \cdot \quad \cdot \quad \cdot \\
&\quad \cdot \quad \cdot \quad \cdot \\
&(\bar{C}(6, k), k = 1, 2, \dots, 6) \\
&\bar{V} \quad \bar{S} \\
&\bar{x}_b \quad \bar{y}_b \quad \bar{z}_b \\
&\bar{x}_g \quad \bar{y}_g \quad \bar{z}_g \\
&(\bar{r}_{1j}, j = 1, 3) \\
&(\bar{r}_{2j}, j = 1, 3) \\
&(\bar{r}_{3j}, j = 1, 3)
\end{aligned}$$

5.2 Time Domain Hydrodynamic Quantities

The time-domain quantities computed by T̄MIT are defined completely in §7. These quantities are non-dimensionalized as follows (with an over-bar indicating the non-dimensional quantity).

Time is

$$\bar{t} = t\sqrt{g/L}.$$

The non-dimensional time-histories of the first-order force in the x -direction (negative of the resistance), force in the z -direction (suction force), and the moment about the y -axis (trim moment, bow-down = positive) acting on the body are

$$\bar{F}_j = \frac{F_j}{\rho U^2 L^n}$$

where $n = 2$ for $j = 1, 3$ and $n = 3$ for $j = 5$.

The non-dimensional body motions are

$$\bar{x}_k = \frac{x_k}{L/L^n},$$

where $n = 0$ for $k = 1, 2, 3$ and $n = 1$ for $k = 4, 5, 6$.

The non-dimensional impulse-response functions are

$$\bar{K}_{jk} = \frac{K_{jk}}{\rho L^n (g/L)}, \quad \bar{K}_{jD} = \frac{K_{jD}}{\rho L^m (g/L)^{3/2}}, \quad \bar{a}_{jk} = \frac{a_{jk}}{\rho L^n}, \quad \bar{b}_{jk} = \frac{b_{jk}}{\rho L^n \sqrt{g/L}}, \quad \bar{c}_{jk} = \frac{c_{jk}}{\rho L^n g/L}.$$

where $n = 3$ for $j, k = 1, 2, 3$; $n = 4$ for $j = 1, 2, 3$ $k = 4, 5, 6$ or $j = 4, 5, 6$ $k = 1, 2, 3$; $n = 5$ for $j, k = 4, 5, 6$; $m = 3$ for $j = 1, 2, 3$; $m = 4$ for $j = 4, 5, 6$.

The local fluid pressure and velocity on the body surface computed during a simulation are non-dimensionalized as follows.

$$\bar{p} = \frac{p}{\rho g L}, \quad \bar{\nabla} \phi = \frac{\nabla \phi}{\sqrt{g L}}.$$

Note that the input simulation wave elevation must be non-dimensionalized

$$\bar{\zeta}(t) = \frac{\zeta(t)}{L}.$$

5.2.1 Format of the time domain hydrodynamic output

Each time-domain hydrodynamic output file begins with four lines of information that identify the TĪMIT run by the run date- and time-stamp and the headers from the GDF, PCF, and RCF. For example, these lines might appear:

```
TiMIT job on: 21-Nov-95 at: 16:03:54
GDF header: 32 panel Wigley hull with y symmetry.
PCF header: Pot. form., Fn=0.3, beta=0,180 -10<0<10, trlim=10, dt=.2
RCF header: Irf's, frf's, simulation, rao's
```

These four lines common to all of the files are followed by a line specific to the file identifying its contents with quantities such as modes, wave heading angle, body speed, *etc.* Table 1 identifies the time-domain hydrodynamic output in each file resulting from the selection of the options ITOPTN.

5.3 Frequency Domain Hydrodynamic Quantities

The frequency-domain hydrodynamic quantities computed by TĪMIT and defined more completely in §7, are non-dimensionalized and output in the same way they are in WAMIT[®].

In the cases of local quantities and mean drift forces (IFOPTN = 5-9) the output is for the complete solution, both radiation and diffraction, for a given frequency. The same quantities can be evaluated separately for either the radiation or diffraction problems by the appropriate choice of options. (See §6.6.)

Frequency and period are normalized by $\sqrt{g/L}$ so that

$$\bar{\omega} = \frac{\omega}{\sqrt{g/L}}, \quad \bar{T}_0 = \frac{2\pi}{\bar{\omega}_0},$$

where the encounter frequency ω is related to the absolute frequency by

$$\omega = \omega_0 - U k \cos \beta$$

and the wavenumber $k = \omega_0^2/g$.

The frequency-domain hydrodynamic coefficients are defined

$$\bar{A}_{jk} = \frac{A_{jk}}{\rho L^n} - \frac{\bar{c}_{jk}}{\bar{\omega}^2}, \quad \bar{B}_{jk} = \frac{B_{jk}}{\rho L^n \omega}, \quad \bar{X}_j = \frac{X_j}{\rho g A L^m},$$

where $n = 3$ for $j, k = 1, 2, 3$; $n = 4$ for $j = 1, 2, 3$ $k = 4, 5, 6$ or $j = 4, 5, 6$ $k = 1, 2, 3$; $n = 5$ for $j, k = 4, 5, 6$; $m = 2$ for $j = 1, 2, 3$; $m = 3$ for $j = 4, 5, 6$; and $A = |\mathcal{A}|$. Note that the

Option	Output File Name	Output Quantities			
1	itoptn.1	\bar{t}	$\bar{F}_x(t)$	$\bar{F}_z(t)$	$\bar{M}_y(t)$
2	irf.ij	a_{jk}	b_{jk}	c_{jk}	
		\bar{t}	$\bar{K}_{ij}(\bar{t})$		
3	irf.jH. β	\bar{t}	$\bar{K}_{jD}(\beta, \bar{t})$	$\bar{K}_{jS}(\beta, \bar{t})$	
4	irf.jD. β (.m)	\bar{t}	$\bar{K}_{jD}(\beta, \bar{t})$	$\bar{K}_{jS}(\beta, \bar{t})$	
5 or 6	itoptn.pnl	k	\vec{x}	s	$n_1\dots$
5	itoptn.5	\bar{t}	$\bar{x}_1(\bar{t})$	$\bar{x}_2(\bar{t})$	$\bar{x}_3(\bar{t})\dots$
6	itoptn.6	\bar{t}	$\bar{p}(\vec{x}_1, \bar{t})$	$\bar{p}(\vec{x}_2, \bar{t})$	$\bar{p}(\vec{x}_3, \bar{t})\dots$
7	itoptn.7x	\bar{t}	$\bar{\phi}_x(\vec{x}_1, \bar{t})$	$\bar{\phi}_x(\vec{x}_2, \bar{t})$	$\bar{\phi}_x(\vec{x}_3, \bar{t})\dots$

Table 1: Names and content of the files written by the **respon** module for time-dependent quantities. Under ‘‘Option’’ are the ITOPTN values. Under ‘‘Output File Name,’’ the characters following ‘‘irf’’ for options 2 through 4 are variables as follows: i and j are the mode indices (with the constants H for Haskind and D for diffraction), β is the incident wave heading angle in degrees, and m is the index for the three components of the diffraction impulse-response function if $U > 0$ and the waves are incident from abaft the beam. Options 6 and 7 provide local quantities at the centroids of specified panels denoted \vec{x}_n . Useful data for these panels are provided in `itoptn.pnl` for which k is the panel index, \vec{x} is the three Cartesian coordinates of its centroid, s is its area, and $n_1\dots$ is its generalized normal. Selection of option 7 results in three output files (one for each velocity component), only the one for the x -component is shown, the other two are analogous.

non-dimensional definition of the added-mass coefficient includes the hydrodynamic restoring force coefficient c_{jk} .

The response amplitude operators (RAO) for motion, fluid pressure, and fluid velocity are given by

$$\bar{\xi}_k = \frac{\xi_k}{A/L^n}, \quad \bar{p} = \frac{p}{\rho g A}, \quad \nabla \bar{\phi} = \frac{\nabla \phi}{i g A / (\omega_0 L)},$$

where $n = 0$ for $k = 1, 2, 3$ and $n = 1$ for $k = 4, 5, 6$, and $A = |\mathcal{A}|$. These quantities can be evaluated separately for the diffraction or radiation problems by following the procedure described in §6.6.

The definition of the non-dimensional mean drift force and moment in unidirectional waves is

$$\bar{F}_i = \frac{F_i}{\rho g A^2 L^k}$$

where $k = 1$ for the forces ($i = 1, 2, 3$), and $k = 2$ for the moments ($i = 4, 5, 6$). These parameters can be evaluated separately for the diffraction or radiation problems by following the procedure described in §6.6.

5.3.1 Format of the frequency domain output

Each frequency-domain hydrodynamic output file begins with four lines of information that identify the TIMIT run by the run date- and time-stamp and the headers from the GDF, PCF, and RCF. For example, these lines might appear:

TIMIT job on: 21-Nov-95 at: 16:03:54

GDF header: 32 panel Wigley hull with y symmetry.

PCF header: Pot. form., Fn=0.3, beta=0,180 -10<0<10, trlim=10, dt=.2

RCF header: Irf's, frf's, simulation, rao's

Table 2 identifies the frequency-domain hydrodynamic output in each file resulting from the selection of the options IFOPTN.

Option	Output File Name	Output Quantities								
1	ifoptn.1	T	i	j	\bar{A}_{ij}	\bar{B}_{ij}				
2	ifoptn.2	T_0	β	i	$ \bar{X}_i $	$\arg \bar{X}_i$	$\Re \bar{X}_i$	$\Im \bar{X}_i$		
3	ifoptn.3	T_0	β	i	$ \bar{X}_i $	$\arg \bar{X}_i$	$\Re \bar{X}_i$	$\Im \bar{X}_i$		
4	ifoptn.4	T_0	β	i	$ \bar{\xi}_i $	$\arg \bar{\xi}_i$	$\Re \bar{\xi}_i$	$\Im \bar{\xi}_i$		
5	ifoptn.pnl	m	k	x	y	z	s	n_1	$n_2 \dots$	
5	ifoptn.frq	ω_0	T_0	k	λ	β	ω	T		
5	ifoptn.5p	T_0	β	m	n	$ \bar{p} $	$\arg \bar{p}$	$\Re \bar{p}$	$\Im \bar{p}$	
5	ifoptn.5vx	T_0	β	m	n	$ \bar{\phi}_x $	$\arg \bar{\phi}_x$	$\Re \bar{\phi}_x$	$\Im \bar{\phi}_x$	
9	ifoptn.9	T_0	β	β	i	$ \bar{F}_i $	$\arg \bar{F}_i$	$\Re \bar{F}_i$	$\Im \bar{F}_i$	

Table 2: Names and content of the files written by the **respon** module for frequency-dependent quantities. Under “Option” are the IFOPTN values. Under “Output Quantities” T_0 is the absolute wave period in seconds, β is the incident wave heading angle in degrees, i and j are mode indices, m is the geometric reflection index, n is the panel index, k is the wave number, λ is the wave length, ω is the encounter frequency, T is the encounter period, and the rest of the quantities are the hydrodynamic variables define in this and §7. Phases are relative to a wave crest at the origin of the global coordinate system. Selection of option 5 for velocities (IFOPTN(5) = 2 or 3) results in three output files (one for each velocity component), only the one for the x -component is shown, the other two are analogous. For the supplementary panel file **ifoptn.pnl**, k is the panel index, x , *etc*, are the three Cartesian coordinates of its centroid, s is its area, and $n_1 \dots$ is its generalized normal. Selection of option 9 for mean force and moment results in \bar{F}_i on six sequential lines, with the moment components defined with respect to the body coordinate system at rest. When $\text{IRAD} \neq -1$, an additional three components of the moment are provided which are defined with respect to the body-fixed coordinate system. These two sets of moment components are distinguished in the output file by the sign of the mode index. For $i = 4, 5, 6$, the moment components are in the body coordinate system at rest. For $i = -4, -5, -6$, the moment components are in the body-fixed coordinate system.

6 Strategies for Using T_IMIT

6.1 Memory Allocation in T_IMIT

The point of exploiting some aspects of FORTRAN-90/95 in T_IMIT was to implement dynamic memory allocation. Users of version 4.* are no longer faced with recompiling for each run, or setting generous dimensions that will be sufficient for most cases.

The amount of memory required, in words, for a particular run of **potent** may be estimated from:

$$130 \text{ NPT} + 60 \text{ NTT NPT} + 20 \text{ NPT}^2$$

while the amount of disk space needed, in words, when **IGREEN** is set equal to zero (i.e. when the Green function coefficients are stored for re-use) for a particular run of **potent** may be estimated from:

$$2 \text{ MAX}(2 \text{ IPOT}, 2 \text{ ISOR}, 3 \text{ IVEL}) (\text{NPT} + \text{NW}) \text{ NPT NTT}$$

where **NPT** is the number of panels input in the GDF, **NW** is the number of panels along the waterline, **NTT** is the number of time steps; **IPOT**, **ISOR**, and **IVEL** are defined in § 4; and we have assumed one plane of symmetry and one wave heading. Note that in a typical UNIX environment scratch files will be opened in the user's temporary directory (usually given by the environment variable **\$TMP** or **\$TMPDIR**) so the temporary-directory environment variable must be set to point to a directory on a partition with sufficient free space. If the option to compute the Green function values as they are needed is chosen, then the processing time will increase by approximately a factor of four on a typical workstation.

The cpu time required will depend on the particular machine, but the scaling of the time for any number of panels and time steps is:

$$\text{CPU time} \propto \text{NPT}^2 \text{NTT}^2.$$

Note that the time required for LU decomposition scales like NPT^3 , but for typical numbers of panels and time steps the cpu effort is dominated by the convolution effort which scales as above.

6.2 Optimizing T_IMIT

The file **Makefile**, edited during the installation should contain commands for optimized executables if you provide the compiler switches for optimization. Note however that the rules provided in **Makefile** are for compilation on a file-by-file basis (the **-c** switch is used). On many compilers, higher optimization levels may be achieved if all files are compiled together. This suggests that the specific procedure which allows for the highest level of optimization with your compiler should be added to the makefile and invoked for large T_IMIT runs.

6.3 Efficiency

This section is included to provide some guidance to the user on choosing run parameters which are appropriate for the desired analysis. For any complex analysis the **potent** module should be compiled at the highest level of optimization available with your compiler. This is not, in

general, achievable with the makefile provided with the distribution. You must provide the compilation rules for optimization which are appropriate for your environment.

The **potent** module requires essentially all of the computational effort needed by T̄MIT to complete any given analysis. This makes it a good idea to plan your **potent** run very carefully, after which numerous runs of the **respon** module (to perform different simulations for example) can be executed using relatively little computational effort.

The computational cost (in CPU time) of running the **potent** module is dominated by the total number of Green function coefficients needed for the job, and therefore scales with the number of unknowns squared times the number of time steps squared. Thus the CPU time required by a particular machine for engineering applications can be estimated by scaling up from a more coarsely discretized run.

The cost of computing the potentials is the same whether it is done via the potential formulation or via the source formulation, but computing the three components of fluid velocity is approximately twice the effort involved in computing the potentials. On the other hand, adding modes to the radiation problem and/or wave heading angles to the diffraction problem requires additional right hand sides at each time step, but the same number of total Green function coefficients, and so is generally insignificant to the total cost. It is therefore a good idea to solve the complete radiation problem and the diffraction problem for a number of wave heading angles, all at one time. T̄MIT also provides the option of solving the radiation problem for a different number of time steps as one uses to solve the diffraction problem. We do not encourage the user to take advantage of this option, since it simply decreases the number of right hand sides after one problem has finished, but does not effect the total number of Green function coefficients. We suggested instead that the user determine which problem needs the most time steps, and set up both problems with that number of time steps.

The most delicate aspect of setting up the **potent** run is in choosing the time step size and the time range of the solution. The accuracy of the solution is a function of discretization, both in time and space, and the only way to be completely confident in your results is to uniformly refine both the spatial and the temporal discretizations until convergence is reached. Some general guidelines are as follows. (Note that all numeric guidelines are with respect to non-dimensional time, $\bar{t} = \sqrt{\frac{g}{L}} t$.)

- When $U = 0$, the minimum acceptable time range will be determined by the diffraction problem. Acceptable accuracy in the integrated quantities can usually be obtained using a non-dimensional time range of $-5 \leq \bar{t} \leq 5$, however local quantities (especially velocities) may require as much as $-15 \leq \bar{t} \leq 15$ because of very slowly traveling short waves in the solution.
- When $U > 0$, the minimum acceptable time range will usually be determined by the radiation problem. The asymptotics of the radiation problem are dominated by a slowly decaying oscillation at the critical frequency corresponding to $\tau = \bar{\omega}_c F_n = 1/4$ where the body Froude number is $F_n = U/\sqrt{gL}$. In order to make use of the quantities computed by **potent** the record must be long enough to allow an estimate of this oscillation in the solution. Generally this means that $\bar{t}_{max} \approx 7 + 8\pi * F_n$ (or $-\bar{t}_{max}/2 \leq \bar{t} \leq \bar{t}_{max}/2$ for the diffraction problem.) If integrated quantities are the only concern this is a conservative limit, and using $\bar{t}_{max} \approx 7 + 4\pi * F_n$ will provide sufficient accuracy in most cases.

- When the time step size is made too large errors will dominate the solution and it will become unstable in time. This situation is usually detected internally by T̄MIT and the run is halted with an error message. In this case, the run should be repeated using a finer temporal discretization.
- A finer spatial discretization usually requires a finer temporal discretization as well. Most bodies show convergence with time discretization at a non-dimensional step size in the range of $\Delta\bar{t} = 0.01 \rightarrow 0.1$.
- We expect to see convergence for ship-like forms using somewhere from 250 to 400 unknowns on one half of the hull.

While executing, T̄MIT writes a new TIMIT.P2R file every ten time steps of each formulation. This feature can be used to check the progress of a job, and also assures that partial results can be salvaged from a crashed job. In a typical UNIX environment, the `respon` module can be run at any time after starting the `potent` module. Note however that jobs are completed sequentially in `potent` (*i.e.* potential formulation, source formulation, potentials from source strengths, velocities from source strengths) so for example, during a job with IPOT=1 and IVEL=1, no information about velocities will be available until the potentials have been computed for the entire time range.

6.4 Effective GDF's

It is important to discretize the geometry of a body in a way which will produce the most accurate hydrodynamic results for a given amount of computational effort. While we expect results to converge as the number of panels increases, the specific way that the panels are arranged on the body surface can affect this process. There are several guidelines we can suggest which will promote efficient computations.

An important consideration for determining the discretization is the interplay between the frequency content of the computation and the representative length of a panel. If the panels have a representative length h , then we cannot expect to resolve waves of length less than $2h$ in principle, and perhaps twice that in practice. On the other hand, since the Green function is never evaluated at a depth less than the submergence of the waterline panel centroids, the computation is exponentially filtered with respect to wavenumber. This implies that panels should not have a large horizontal dimension and a small vertical dimension. This is particularly true for forward speed problems.

Here are several ideas for producing effective discretizations:

- Panel aspect ratios should be approximately 1, where the aspect ratio is the horizontal dimension divided by the vertical dimension. Aspect ratios of less than 1 may be useful, although for ships, this can lead to a poor approximation of the geometry for coarse discretizations.
- Panels which lie in proximate, roughly parallel, planes, (such as those on the surfaces of a keel), should have a characteristic dimension which is less than the separation distance of the planes.

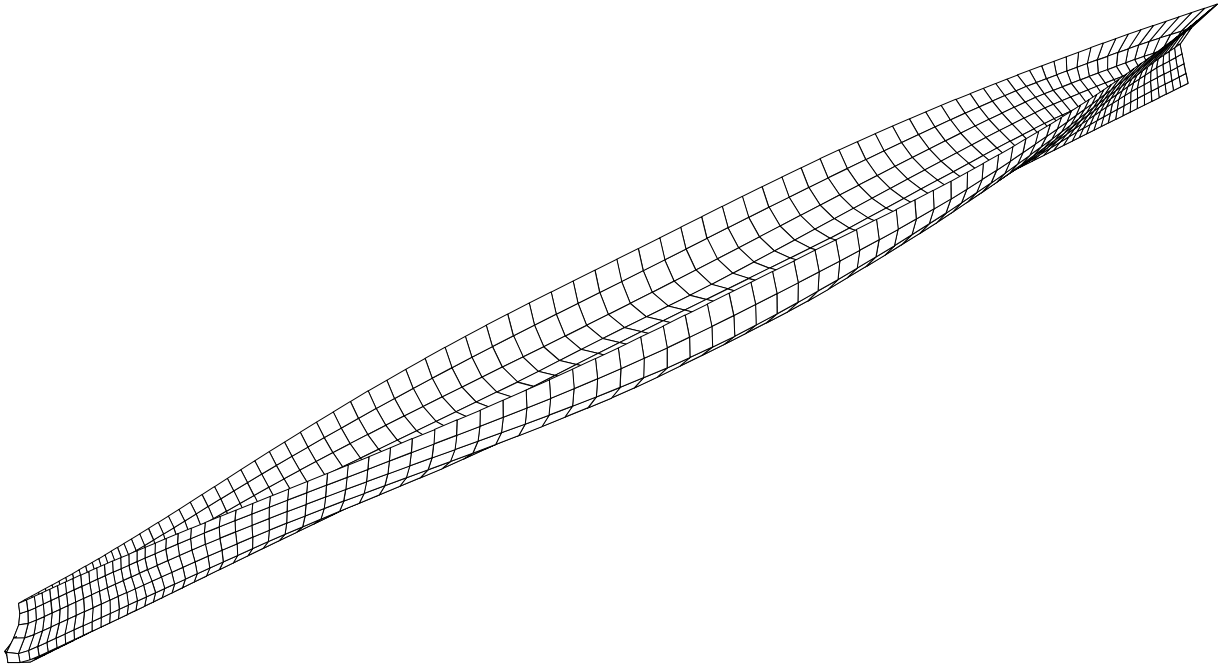


Figure 2: Discretization of the SL-7 with 1032 panels (516 unknowns). (Produced by the panel generation program `xdish`.)

- Panels along the waterline of a body should be as close to rectangular as possible. This may be difficult for ships with overhanging sterns or bow flare.
- Adjacent panels should not be of disparate size.
- Cosine spacing should be used approaching the corners of a body and panels on either side of the corner should be of comparable size.

Figure 2 shows a discretization of the SL-7 hull which produces good results for the number of panels employed. Note that all of the suggestions above, with the exception of the suggestion concerning corners, are followed in this discretization.

6.5 Time-step Size

For any given spatial discretization there is a time-step size which is adequately small to produce results which are converged with respect to temporal discretization. Ideally one would like to make computations with any spatial discretization at this temporal discretization and demonstrate convergence as a function of the number of panels. However it is not easy *a priori* to know the proper time step size. Table 3 gives the results from our experience of running `TIMIT`.

Problem	Number of Panels	Time-Step Size
$F_n = 0.0$ hemisphere	64	0.2
$F_n = 0.0$ hemisphere	144	0.15
$F_n = 0.0$ hemisphere	256	0.1
$F_n = 0.3$ Wigley hull	288	0.1
$F_n = 0.3$ Wigley hull	512	0.05

Table 3: Suggested time-step sizes for various analyses. Number of panels refers to the number for the entire body.

6.6 Separating the Radiation and Diffraction Problems

The frequency-domain output quantities in IFOPTN = 5-9 can be evaluated for the radiation problem alone or the diffraction problem alone by appropriate selections of the indices IRAD and IDIF in the PCF, and the option indices IFOPTN in the RCF. Table 6.6 summarizes the most useful possibilities.

In the first case, where there are no incident waves, the radiation problem is defined by the superposition of all modes specified in the PCF, each having unit amplitude and zero phase. Thus the nondimensional complex amplitudes $\bar{\xi}_j$ of the specified mode(s) of motion are set equal to 1. For example, if MODE(3)=1 the value of A which applies in the nondimensional definition of the pressure is obtained by the relation $A = \xi_3$, where ξ_3 is the dimensional complex amplitude of the forced heave oscillation.

There is no corresponding separation of radiation and diffraction in the computation of time-domain quantities in TIMIT. In other words, a simulation requires both the diffraction problem solution (to compute the excitation on the right-hand side of the equations of motion) and the radiation problem solution (to compute the hydrodynamic memory term on the left-hand side of the equations of motion). However the impulse-response functions for radiation and/or diffraction may be output and used in a post processor to find the force and moment on a body due to these effects alone.

IRAD	IDIF	IFOPTN(4)	Outputs from Options 5-9
0, 1	-1	0	No incident waves.
-1,0,1	0,1	0	No body motions.
0	1	1,2	Specified modes of motion in incident waves.
1	1	1,2	All modes of motion in incident waves.

Table 4: Parameter combinations for the separation of radiation and diffraction effects in frequency-domain computations. Other combinations are nonphysical.

6.7 Inclusion of Nonlinear Damping

The equations of motion (see §5) which are solved in the **respon** module include quadratic and cubic damping terms. These terms $\dot{x}_k|\dot{x}_k|$ and \dot{x}_k^3 have 6×6 coefficient matrices $b_{jk}^{(2)}$ and

$b_{jk}^{(3)}$ respectively. The numerical values for these matrices are loaded via DATA statements in the subroutine SIMULA where the coefficient matrices are called B2 and B3, respectively. All seventy-two of these coefficients are set to zero by default. If you choose to insert nonlinear damping coefficients, you must take care to non-dimensionalize them consistently with the T̄MIT non-dimensionalization. This non-dimensionalization leads to:

$$\bar{b}_{jk}^{(2)} = \frac{b_{jk}}{\rho L^n},$$

where $n = 2$ for $j, k = 1, 2, 3$; $n = 4$ for $j = 1, 2, 3; k = 4, 5, 6$; $n = 3$ for $j = 4, 5, 6; k = 1, 2, 3$; and $n = 5$ for $j, k = 4, 5, 6$. And:

$$\bar{b}_{jk}^{(3)} = \frac{b_{jk}}{\rho L^n \sqrt{L/g}},$$

where $n = 1$ for $j, k = 1, 2, 3$; $n = 4$ for $j = 1, 2, 3; k = 4, 5, 6$; $n = 2$ for $j = 4, 5, 6; k = 1, 2, 3$; and $n = 5$ for $j, k = 4, 5, 6$.

6.8 Generalized Modes

Generalized modal analysis is a logical extension of the analysis of a single rigid body in the context of a linearized problem. Any useful set of modes may be used. The ones most usually used are bending modes for ships, multiple body rigid modes, and modes representing the compliant coupling of multiple units. While the concept of this type of analysis is straightforward, the practice is not. There is a FORTRAN source file `defmod.f` in which the user must define the generalized modes, that is provide the panel-wise values of the generalized normal so that boundary conditions may be set. The user must also specify a symmetry index for each mode by indicating the rigid mode with the same symmetry. For instance the first beam mode is fore and aft symmetric and so shares the symmetry of rigid mode 3, “heave.” The version of `defmod.f` that is provided with the distribution contains the definitions of the first eight Euler-Bernoulli beam modes often used as the “dry modes” for the bending of a slender ship. When using generalized modes such as this, the inertia and stiffness matrices must be provided for these modes as well. T̄MIT will use the mass matrix provided in the BPF rather than the one internally calculated from the hydrostatic analysis. T̄MIT will add the structural stiffness matrix provided in the BPF to the one internally calculated from the hydrostatic analysis. T̄MIT 3.* does not support the addition of structural damping to the hydrodynamic damping. The only output provided for generalized modal analysis is impulse- and frequency-response functions and RAO’s. Users who undertake this analysis usually have their own preferences on how to use these data in simulations.

6.9 Negative Forward Speed

Both the `potent` and `respon` modules trap a $U < 0$ input. The convention that U must have positive sense leaves the following seas case determinable from the wave heading angle alone. However the need may arise to conduct an analysis with negative forward speed. To accomplish this, the geometric input must be reflected about the body coordinate system plane $x = 0$ and the wave heading angle adjusted accordingly. For example, consider the case of a body in head seas ($\beta = 180$ degrees). The positive forward speed case is run conventionally at speed U .

The negative forward speed case is run after the panel vertices have been operated on so that $(x_{\text{new}}, y_{\text{new}}, z_{\text{new}}) = (-x_{\text{old}}, y_{\text{old}}, z_{\text{old}})$, with wave heading angle $\beta = 0$, and speed U . The second case will be treated like a conventional following seas situation as is required for head seas with negative forward speed.

6.10 Very Low Froude Number

There is no difficulty with running T $\bar{\text{I}}$ MIT at *zero* Froude number. However robustness issues arise at very low Froude number in the asymptotic continuation of the $\tau = 1/4$ effect. As $F_n \rightarrow 0$, the critical frequency gets larger or $T_c \rightarrow 0$. For instance in infinite depth $F_n = 0.025 \Rightarrow T_c = 0.63$. For cases like this, there are not likely to be very many time-steps in this period and asymptotic matching may give poor results or fail. If the number of time steps in T_c is less than 4, an error message will be generated by subroutine T14FIT and its derivatives. In Figure /reffig:tauc it can be seen that for a given speed, U , ω_c decreases with depth. Therefore this low Froude number difficulty is less severe in finite depth computations.

6.11 Error Checking

There are routines in T $\bar{\text{I}}$ MIT which attempt to screen out inconsistent or illegal run control parameters and problems with the geometric data. Warnings and error returns are generated when these routines detect exceptions. Warnings do not stop the program, but errors do (and so are referred to as *fatal* errors).

Errors are reported and the run is halted if:

- Any of the parameters discussed in §4.3 through 6.1 are illegal or inconsistent.
- Any of the body panels are found to lie above the free-surface.

An example error message is the following:

TFE-READGM: Number of panels exceeds dimension.

indicating that the dimensions have not been set large enough to accommodate the data in the GDF. The capitalized prelude to the message indicates a T $\bar{\text{I}}$ MIT *Fatal Error*, from the routine READGM.

Warnings are generated but the run is not halted if:

- No waterline panels are detected. In this case the body is assumed to be submerged.
- The three calculations of the body volume do not agree to six significant digits. The volume calculations are not expected to agree perfectly and are printed to allow the user to immediately detect gross errors in the geometric discretization. For example, if one of the three volumes is radically different from the others, the discretization may be incomplete; or, if the volume is negative, the normal to the body has been defined to point in the wrong direction (see §4.3).
- Any two panels are found with nearly coincident centroids. This is an attempt to screen out thin appendages to the body such as the rudder. Panels whose centroids are much closer than a characteristic panel dimension (the panel diagonal for example) will cause difficulties in a panel method and should be removed from the discretization.

An example warning message is the following:

TW-READGM: Extra panels in the GDF.

indicating that the data in the GDF is not consistent with the variable NP. The capitalized prelude to the message indicates a T_IMIT *Warning*, from the routine READGM.

7 Mathematical Formulation

Further details of the theory, mathematics, and computations may be found in references [3], [2], and [11], as well as others cited throughout this section.

We employ body-fixed Cartesian coordinates (x, y, z) which move with the steady forward velocity U of the body, in the $+x_0$ -direction of an earth-fixed Cartesian coordinate system. The origin of the earth-fixed system is in the plane of the free surface with z_0 positive upwards. Six degrees of unsteady motion are defined by the corresponding displacements $x_k(t)$, where $k = 1, 2, \dots, 6$ for surge, sway, heave, roll, pitch, and yaw, respectively.

Assuming the body and its forcing comprise a stable linear system, the equations of motion may be written as [5]

$$\sum_{k=1}^6 (M_{jk} + a_{jk}) \ddot{x}_k + b_{jk} \dot{x}_k + (C_{jk} + c_{jk}) x_k + \int_{-\infty}^t d\tau K_{jk}(t - \tau) \dot{x}_k(\tau) = F_j(t) \quad j = 1, 2, \dots, 6, \quad (1)$$

where an over-dot indicates differentiation with respect to time. It is convenient in the time-domain approach to include nonlinear terms in these equations, in particular roll-damping terms, leading to

$$\sum_{k=1}^6 (M_{jk} + a_{jk}) \ddot{x}_k + b_{jk} \dot{x}_k + b_{jk}^{(2)} \dot{x}_k |\dot{x}_k| + b_{jk}^{(3)} \dot{x}_k^3 + (C_{jk} + c_{jk}) x_k + \int_{-\infty}^t d\tau K_{jk}(t - \tau) \dot{x}_k(\tau) = F_j(t) \quad j = 1, 2, \dots, 6. \quad (2)$$

The body inertia matrix is M_{jk} , and the linearized hydrostatic restoring-force coefficients are given by C_{jk} . The radiation impulse-response functions are composed of the hydrodynamic coefficients and the kernel of the convolution on the left-hand side of (1). A radiation impulse-response function is the force on the body in the j^{th} direction due to an impulsive velocity in the k^{th} direction, with the coefficients a_{jk} , b_{jk} , c_{jk} , accounting for the instantaneous forces proportional to the acceleration, velocity, and displacement, respectively, and the memory function $K_{jk}(t)$ accounting for the free-surface effects which persist after the motion occurs. For the radiation problem we use the term ‘memory function’ to distinguish this portion of the impulse-response function from the instantaneous force components outside of the convolution in (1). For the diffraction problem, the memory function is equal to the impulse-response function. The terms F_j on the right side of (1) are the components of the exciting force and moment due to the incident wave elevation $\zeta(t)$, defined at a prescribed reference point in the body-fixed coordinate system. (‘Force’ is understood hereafter in the generalized sense to include the moment, for $j = 4, 5, 6$.) Following [9], the exciting-force components are expressed by means of convolution integrals

$$F_j(t) = \int_{-\infty}^{\infty} d\tau K_{jD}(t - \tau, \beta) \zeta(\tau) \quad j = 1, 2, \dots, 6. \quad (3)$$

Here, the kernel $K_{jD}(t, \beta)$ is the diffraction impulse-response function: the force on the body in the j^{th} direction due to a uni-directional impulsive wave elevation with a heading angle of β .

The convolution on the left-hand side of (1) could be expressed alternatively in terms of the displacements x_k or accelerations \ddot{x}_k . With the latter choice, (1) is replaced by the following equation

$$\sum_{k=1}^6 (M_{jk} + a_{jk})\ddot{x}_k + b_{jk}\dot{x}_k + (C_{jk} + c_{jk})x_k + \int_{-\infty}^t d\tau L_{jk}(t - \tau)\ddot{x}_k(\tau) = F_j(t) \quad j = 1, 2, \dots, 6, \quad (4)$$

The kernels in (1) and (4) are related to each other by differentiation [2], specifically:

$$K_{jk}(t) = \frac{\partial}{\partial t} L_{jk}(t) \quad (5)$$

7.1 The Inertial Terms

For a body in free stable flotation we have $m = \rho \mathcal{V}$, $x_b = x_g$, and $y_b = y_g$ where m is the body mass and (x_g, y_g, z_g) are the coordinates of the body's center of gravity. In this case the inertia matrix is defined as follows

$$M = \begin{pmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_{11} & I_{12} & I_{13} \\ mz_g & 0 & -mx_g & I_{21} & I_{22} & I_{23} \\ -my_g & mx_g & 0 & I_{31} & I_{32} & I_{33} \end{pmatrix} \quad (6)$$

where the moments of inertia I_{jk} are expressed in terms of the corresponding radii of gyration r_{jk}

$$I_{jk} = \rho \mathcal{V} r_{jk} |r_{jk}|.$$

The dimensional values of the radii of gyration and z_g must be input to T $\bar{\text{I}}$ MIT through the BPF, from which T $\bar{\text{I}}$ MIT computes the inertia matrix as in equation (6).

7.2 The Hydrostatic Terms

All hydrostatic data can be expressed as surface integrals over the body wetted surface S_B , by virtue of Gauss' divergence theorem. The volume is defined

$$\mathcal{V} = - \int \int_{S_B} n_1 x dS = - \int \int_{S_B} n_2 y dS = - \int \int_{S_B} n_3 z dS. \quad (7)$$

The surface area is defined

$$S = \int \int_{S_B} dS. \quad (8)$$

The coordinates of the center of buoyancy are defined

$$\begin{aligned} x_b &= \frac{-1}{2\mathcal{V}} \int \int_{S_B} n_1 x^2 dS, \\ y_b &= \frac{-1}{2\mathcal{V}} \int \int_{S_B} n_2 y^2 dS, \\ z_b &= \frac{-1}{2\mathcal{V}} \int \int_{S_B} n_3 z^2 dS. \end{aligned} \quad (9)$$

The matrix of hydrostatic coefficients is defined

$$\begin{aligned}
C_{33} &= \rho g \int_{S_B} n_3 dS, \\
C_{34} &= \rho g \int_{S_B} y n_3 dS, \\
C_{35} &= -\rho g \int_{S_B} x n_3 dS, \\
C_{44} &= \rho g \int_{S_B} y^2 n_3 dS + \rho g \mathcal{V} z_b - m g z_g, \\
C_{45} &= -\rho g \int_{S_B} x y n_3 dS, \\
C_{46} &= -\rho g \mathcal{V} x_b + m g x_g, \\
C_{55} &= \rho g \int_{S_B} x^2 n_3 dS + \rho g \mathcal{V} z_b - m g z_g, \\
C_{56} &= -\rho g \mathcal{V} y_b + m g y_g.
\end{aligned} \tag{10}$$

Note that $C(jk) = C(kj)$, and for all other values of the indices j, k , not shown above, $C(jk) = 0$.

7.3 The Hydrodynamic Pressure

With the assumptions of linearized potential flow the velocity potential can be expressed in the form

$$\Phi = U \bar{\Phi} + \bar{\phi} + \sum_{k=1}^6 \phi^k + \phi^I + \phi^S, \tag{11}$$

where each term in (11) is a solution of the Laplace equation in the fluid domain. In this decomposition the first term on the right side $U \bar{\Phi}$ represents the steady basis flow about which the problem is linearized. The potential $\bar{\phi}$ is a steady perturbation to this flow, and the remaining terms represent the unsteady perturbations, including six radiation potentials ϕ^k , each corresponding to one rigid-body mode of motion, and the diffraction potential $\phi^I + \phi^S$ corresponding to the incident wave (I) and the scattered disturbance (S) when the body is fixed in its mean position.

To derive appropriate boundary conditions we adopt the Neumann-Kelvin linearization, where we make the assumption that $\bar{\Phi} = -x$. It follows that each of the unsteady potentials in (11) satisfies the free-surface boundary condition

$$\left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right)^2 \phi + g \frac{\partial \phi}{\partial z} = 0, \quad \text{on } z = 0. \tag{12}$$

In this equation ϕ is used to represent any of the perturbation potentials. On the mean position of the body surface, \bar{S}_b , the following boundary conditions are applied:

$$\begin{aligned}
\hat{n} \cdot \nabla \bar{\phi} &= U n_1 \\
\hat{n} \cdot \nabla (\phi^I + \phi^S) &= 0 \\
\hat{n} \cdot \nabla \phi^k &= n_k \dot{x}_k + U m_k x_k.
\end{aligned} \tag{13}$$

The generalized unit normal n_k is defined by

$$\begin{aligned}
(n_1, n_2, n_3) &= \hat{n} \\
(n_4, n_5, n_6) &= \vec{r} \times \hat{n}.
\end{aligned} \tag{14}$$

The steady and the unsteady potentials are coupled through the presence of the m -terms in the body boundary condition, which are

$$m_k = (0, 0, 0, 0, n_3, -n_2), \quad (15)$$

for this linearization.

Appropriate initial conditions must be applied on the free surface. For the radiation problems these are

$$\phi^k = \frac{\partial}{\partial t} \phi^k = 0 \quad \text{on } z = 0 \quad \text{for } t \rightarrow 0, \quad (16)$$

and for the diffraction problem

$$\phi^S \rightarrow 0 \quad \text{on } z = 0 \quad \text{for } t \rightarrow -\infty. \quad (17)$$

The linearized pressure associated with each of these perturbation potentials is given by

$$p = -\rho \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi. \quad (18)$$

A different linearization may be defined by adopting a different choice of $\bar{\Phi}$. A general linearization about $\bar{\Phi}$ results in a more general definition of the m -terms,

$$m_k = \begin{cases} -(\hat{n} \cdot \nabla) \nabla \bar{\Phi} & k = 1, 2, 3 \\ -(\hat{n} \cdot \nabla)(\vec{r} \times \nabla \bar{\Phi}) & k = 4, 5, 6; \end{cases} \quad (19)$$

and of the dynamic pressure,

$$p = -\rho \left[\frac{\partial \phi}{\partial t} + \nabla \bar{\Phi} \cdot \nabla \phi + \frac{1}{2} \vec{\alpha} \cdot \nabla (\nabla \bar{\Phi} \cdot \nabla \bar{\Phi}) \right], \quad (20)$$

where $\vec{\alpha}$ is the vector from the origin to a point on the body in it's disturbed position. The last term in this expression provides a dynamic restoring force on the body. We define the basis flow restoring pressure as

$$p_k^0 = -\rho(\vec{W} \cdot \nabla \vec{W}), \quad k = 1, 2, 3 \quad (21)$$

$$p_k^0 = -\rho[\vec{r} \times (\vec{W} \cdot \nabla \vec{W})], \quad k = 4, 5, 6; \quad (22)$$

(where $\vec{W} = \nabla \bar{\Phi}$) and a matrix of basis flow restoring force coefficients as

$$c_{jk}^0 = - \int \int_{\bar{S}_b} dS p_k^0 n_j. \quad (23)$$

The free-surface condition will also take a more general form when $\bar{\Phi} \neq -x$. TIMIT allows the user to specify the values of $\nabla \bar{\Phi}$, m_k , and $\nabla \nabla \bar{\Phi}$ at the centroid of each panel on the body, thus providing the solution for a mixed linearization of the problem. (“Mixed” because the free-surface condition remains the Kelvin condition equation (12).) This generalization is not made in exposition in this section, that is, all equations reflect the choice of the Neumann-Kelvin linearization. Note however, that when a user defined basis flow is adopted, c_{jk}^0 is computed from the input second derivatives of the basis flow potential and added to c_{jk} [defined in Equation (54) or Equation (55)].

7.4 The Potential Formulation

This initial-boundary-value problem may be recast as an integral equation to be solved on the mean position of the body surface, \bar{S}_b . (See [1] for example.) To this end we apply Green's theorem to ϕ_t , and integrate in time, and after some manipulation obtain

$$2\pi\phi + \int \int_{\bar{S}_b} dS \left(\phi G_n^{(0)} - G^{(0)} \phi_n \right) + \int_{-\infty}^t d\tau \int \int_{\bar{S}_b} dS \left(\phi G_{tn} - G_t \phi_n \right) - \frac{U}{g} \int_{-\infty}^t d\tau \int_{\bar{\Gamma}} dl n_1 \left(\phi (G_{tt} + U G_{t\xi}) + G_t (\phi_\tau - U \phi_\xi) \right) = 0. \quad (24)$$

Here $\bar{\Gamma}$ is the waterline contour and the Green function is $G(\vec{x}; \vec{\xi}, t) = G^{(0)} + G^{(f)}$, where

$$G^{(0)} = \left(\frac{1}{r} - \frac{1}{r'} \right) \\ G^{(f)} = 2 \int_0^\infty dk [1 - \cos(\sqrt{gk} t)] e^{kZ} J_0(kR) \\ \left. \begin{array}{l} r \\ r' \end{array} \right\} = \sqrt{(x - \xi)^2 + (y - \eta)^2 + (z \mp \zeta)^2} \\ Z = (z + \zeta) \\ R = \sqrt{(x - \xi + U[t - \tau])^2 + (y - \eta)^2}, \quad (25)$$

and J_0 is the Bessel function of order zero. The arguments of the functions appearing in equation (24) have been omitted for brevity, but note that the temporal argument of the Green function in the convolution integrals is retarded [*i.e.* $G_t = G_t(\vec{x}; \vec{\xi}, t - \tau)$]. Equation (24) is valid for any point \vec{x} on the body surface, and the spatial integrations are performed in the dummy variable $\vec{\xi}$. In the case of finite depth, the Green function is modified so that it satisfies a no-flux condition on a horizontal plane some distance below the free surface as described in § 7.6.

7.5 Source Only Integral Equations

In computing fluid velocities we prefer to avoid the difficulties associated with taking spatial derivatives of the potential directly by using a source formulation. The source formulation may be derived by defining a flow in the region interior to the body specified by the scalar potential ϕ' . The integral equation for ϕ' is combined with equation (24), and we define the source and dipole strengths respectively as

$$\sigma = \frac{1}{4\pi} (\phi_n - \phi'_n), \quad \mu = \frac{1}{4\pi} (\phi - \phi'). \quad (26)$$

A source only formulation results by choosing $\phi' = \phi$ on the body boundary. This formulation is a first-kind integral equation on \bar{S}_b for σ

$$\phi(\vec{x}, t) = \int \int_{\bar{S}_b} dS G^{(0)}(\vec{x}; \vec{\xi}) \sigma(\vec{\xi}, t) + \int_{-\infty}^t d\tau \int \int_{\bar{S}_b} dS G_t^{(f)}(\vec{x}; \vec{\xi}, t - \tau) \sigma(\vec{\xi}, \tau) - \frac{U^2}{g} \int_{-\infty}^t d\tau \int_{\Gamma_0} n_1^2 dl G_t^{(f)}(\vec{x}; \vec{\xi}, t - \tau) \sigma(\vec{\xi}, \tau). \quad (27)$$

Note that this equation is not immediately useful as it requires the knowledge of the potential (the Dirichlet condition). However if we operate under the integrals in this equation with $\hat{n}_x \cdot \nabla_x$ (where the subscript x indicates that the operation is with respect to the \vec{x} variables rather than the $\vec{\xi}$ variables), we have a second-kind equation for σ which requires the known Neumann boundary condition for the potential.

$$\begin{aligned} \sigma(\vec{x}, t) = & \int \int_{\bar{S}_b} dS G_{n_x}^{(0)}(\vec{x}; \vec{\xi}) \sigma(\vec{\xi}, t) + \int_{-\infty}^t d\tau \int \int_{\bar{S}_b} dS G_{tn_x}^{(f)}(\vec{x}; \vec{\xi}, t - \tau) \sigma(\vec{\xi}, \tau) \\ & - \frac{U^2}{g} \int_{-\infty}^t d\tau \int_{\Gamma_0} n_1^2 dl G_{tn_x}^{(f)}(\vec{x}; \vec{\xi}, t - \tau) \sigma(\vec{\xi}, \tau) = \hat{n} \cdot \nabla \phi(\vec{x}, t) \end{aligned} \quad (28)$$

A vector expression for the fluid velocity may also be derived by similarly operating on equation (27) with the operator ∇_x

$$\begin{aligned} \int \int_{\bar{S}_b} dS \nabla_x G^{(0)}(\vec{x}; \vec{\xi}) \sigma(\vec{\xi}, t) + \int_{-\infty}^t d\tau \int \int_{\bar{S}_b} dS \nabla_x G_t^{(f)}(\vec{x}; \vec{\xi}, t - \tau) \sigma(\vec{\xi}, \tau) \\ - \frac{U^2}{g} \int_{-\infty}^t d\tau \int_{\Gamma_0} n_1^2 dl \nabla_x G_t^{(f)}(\vec{x}; \vec{\xi}, t - \tau) \sigma(\vec{\xi}, \tau) = \nabla \phi(\vec{x}, t) \end{aligned} \quad (29)$$

To obtain the velocity vector at any field point, equation (28) is first solved for the source strength, and then the source strength is used in equation (29) to calculate the three components of the velocity.

7.6 Finite Depth

The finite-depth analysis is carried out as described above for the infinite-depth approach, the only difference being the definition of the Green function.

The transient, free-surface, finite-depth Green function may be written as the sum of a Rankine and a wave part as usual, which we define in the forms (*cf.* [20] equation 13.53)

$$G^{(0)} = \frac{1}{r} + \frac{1}{r'} - 2 \int_0^\infty \frac{e^{-kh}}{\cosh kh} \cosh k(z+h) \cosh k(c+h) J_0(kR) dk \quad (30)$$

$$G_t^w = 2 \int_0^\infty \frac{\sqrt{gk \tanh kh}}{\cosh kh \sinh kh} \sin(t\sqrt{gk \tanh kh}) \cosh k(z+h) \cosh k(c+h) J_0(kR) dk \quad (31)$$

In [16] effective algorithms for the evaluation of the integral in (30) and the fundamental difficulties associated with the efficient evaluation of (31) are described. The approach which we have implemented is to express (31) as the sum of two terms involving the normalized function $F(X, V, T)$, as defined by [16] equation (25), and then to consider the difference function $F - F_\infty$ where F_∞ can be evaluated from the infinite-depth Green function (25). Then we expand this difference function in triple Chebyshev expansions, in unit squares of the rectangular domain $0 \leq X \leq 16$ and $0 \leq T \leq 33$. (Physically, the variables X and T correspond to the horizontal distance from the source to the field point, and the time t , nondimensionalized in terms of the depth h and gravity g .) The coefficients of these Chebyshev expansions are pre-evaluated, and stored for routine use. At each time step of the convolution, $T = \text{constant}$ and the triple expansions are reduced to double expansions in the normalized horizontal and vertical coordinates X and V , which are then evaluated for each combination of source and field

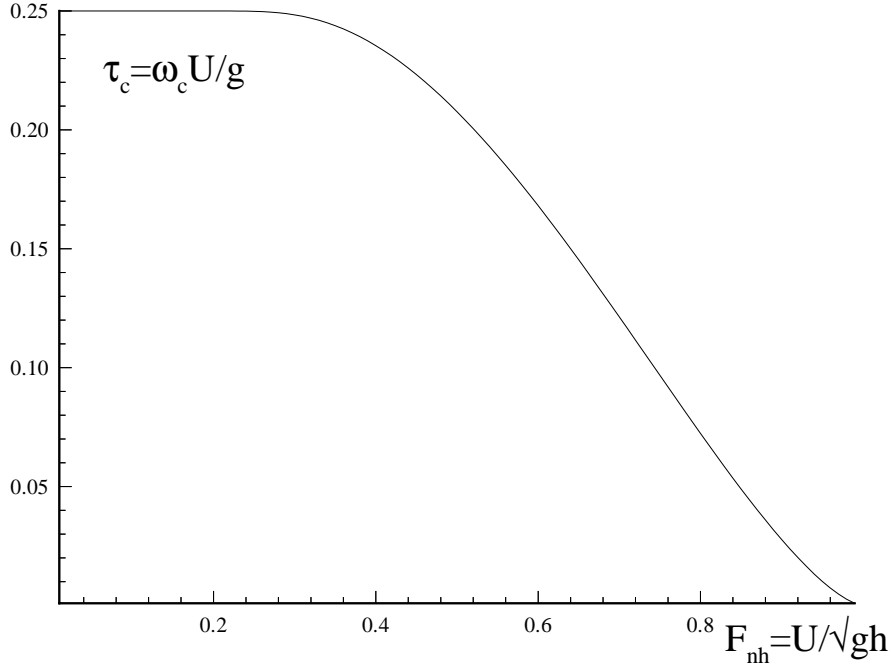


Figure 3: The depth dependence of τ_c .

points. One point to note is the large-time asymptotic behavior of the Green function, since this affects the corresponding behavior of computed hydrodynamic forces. In the infinite-depth case (31) is exponentially small, but for $h < \infty$ $G_t^w \rightarrow \mathcal{O}(1/t)$. When $U \neq 0$, and coordinates are used which move with the ship, $G_t^w \rightarrow \mathcal{O}1/t + \mathcal{O} \sin(\omega_c t)/t$, where ω_c denotes the critical frequency. This is given in the infinite-depth limit by $\tau_c = \omega_c U/g = 1/4$, while the variation of τ_c with depth is shown in Figure 3. From the asymptotic behavior of the Green function it is clear that the time record has to be longer for finite water depth than infinite water depth. Alternatively we could approximate the large time behavior based on the asymptotic analysis. Further discussion of finite-depth analysis may be found in [4].

7.7 Solving for the Radiation Potentials

Since the memory functions for different impulsive motions are related by differentiation, as in (5), the specification of the canonical radiation problems is somewhat arbitrary. For the computation of ϕ^k in TĪMIT, we define the motion $x_k(t)$ to be an impulsive velocity, *i.e.* $\dot{x}_k = \delta(t)$, and $x_k = h(t)$, where $\delta(t)$ is the Dirac function, and $h(t)$ is the Heaviside function. With this choice, the last line of (13) becomes

$$\hat{n} \cdot \nabla \phi^k = n_k \delta(t) + m_k h(t). \quad (32)$$

Anticipating the form of the solution, it is convenient to decompose the potential as follows

$$\phi^k(\vec{x}, t) = \mathcal{N}_k(\vec{x}) \delta(t) + \mathcal{M}_k(\vec{x}) h(t) + \psi^k(\vec{x}, t) h(t). \quad (33)$$

The two time-constant potentials, \mathcal{N}_k and \mathcal{M}_k , are *pressure release* problems satisfying the boundary value problems

$$\begin{aligned} \nabla^2 \mathcal{N}_k &= 0 & \nabla^2 \mathcal{M}_k &= 0 & \text{in } \mathcal{V} \\ \mathcal{N}_k &= 0 & \mathcal{M}_k &= 0 & \text{on } z = 0 \\ \hat{n} \cdot \nabla \mathcal{N}_k &= n_k & \hat{n} \cdot \nabla \mathcal{M}_k &= m_k & \text{on } \bar{S}_b. \end{aligned} \quad (34)$$

The Green function corresponding to these problems is $G^{(0)}$, and these potentials are the solutions to the following two integral equations

$$\begin{aligned} 2\pi \mathcal{N}_k + \iint_{\bar{S}_b} dS \mathcal{N}_k G_n^{(0)} &= \iint_{\bar{S}_b} dS n_k G^{(0)} \\ 2\pi \mathcal{M}_k + \iint_{\bar{S}_b} dS \mathcal{M}_k G_n^{(0)} &= \iint_{\bar{S}_b} dS m_k G^{(0)}. \end{aligned} \quad (35)$$

The transient or *memory* potential, ψ^k , satisfies the following initial-boundary value problem

$$\begin{aligned} \nabla^2 \psi^k &= 0 & \text{in } \mathcal{V} \\ \hat{n} \cdot \nabla \psi^k &= 0 & \text{on } \bar{S}_b. \\ \mathcal{L}(\psi^k) &= -g \frac{\partial}{\partial z} \mathcal{M}_k & \text{on } z = 0, t > 0, \\ \psi^k &= 0 & \text{on } z = 0, t = 0. \\ \frac{\partial}{\partial t} \psi^k &= -g \frac{\partial}{\partial z} \mathcal{N}_k & \text{on } z = 0, t = 0. \end{aligned} \quad (36)$$

where \mathcal{L} represents the linearized free-surface boundary condition operator (equation (12)).

The integral equation for the memory potential may be derived directly from the boundary-value problem, or by substituting the appropriate decomposition of both the potential and the body boundary condition into integral equation (24) and using the conditions (and integral equations) satisfied by the pressure release potentials. The result is

$$\begin{aligned} 2\pi \psi + \iint_{\bar{S}_b} d\vec{\xi} \psi G_n^{(0)} + \int_0^t d\tau \iint_{\bar{S}_b} d\vec{\xi} \psi G_{tn} \\ - \frac{U}{g} \int_0^t d\tau \int_{\bar{\Gamma}} dl (\hat{n}_{2D} \cdot \hat{i}) [\psi (G_{tt} + U G_{t\xi}) + G_t (\psi_\tau - U \psi_\xi)] \\ = \iint_{\bar{S}_b} d\vec{\xi} [n_k G_t - \mathcal{N}_k G_{tn} + m_k G^{(f)} + \mathcal{M}_k G_n^{(f)}] \end{aligned} \quad (37)$$

7.7.1 Decomposition of the radiation source strengths

If radiation fluid velocities are of interest then it is convenient to develop a decomposition of the source strengths corresponding to the decomposed radiation potentials introduced above. We assume a decomposition of each source strength analogous to that for the corresponding potential

$$\sigma_k = \alpha_k \delta(t) + \beta_k h(t) + \gamma_k h(t) \quad (38)$$

where

$$\mathcal{N}_k(\vec{x}) = \iint_{\bar{S}_b} d\vec{\xi} \alpha_k G^{(0)}, \quad \mathcal{M}_k(\vec{x}) = \iint_{\bar{S}_b} d\vec{\xi} \beta_k G^{(0)}. \quad (39)$$

An integral equation for γ_k is obtained by substituting the decomposition of the source strength into equation (27) and using the decomposition of the source strength appearing in equation (38) along with the relations in equation (39). The result is

$$\begin{aligned} \psi(\vec{x}, t) = & \int \int_{\bar{S}_b} d\vec{\xi} \gamma_k(t) G^{(0)} \\ & + \int_0^t d\tau \int \int_{\bar{S}_b} d\vec{\xi} [\alpha_k + \beta_k \tau + \gamma_k(\tau)] G_t(t - \tau) \\ & - \frac{U^2}{g} \int_0^t d\tau \int_{\bar{\Gamma}} n_1^2 dl [\alpha_k + \beta_k \tau + \gamma_k(\tau)] G_t(t - \tau). \end{aligned} \quad (40)$$

Operating on equations (39) and (40) with $\hat{n}_x \cdot \nabla_x$ gives three second-kind integral equations for computing the source strengths α , β , and γ , while the vector form of these equations is used to compute the gradients of the potentials \mathcal{N}_k , \mathcal{M}_k , and ψ^k from the source strengths.

7.8 Solving for the Scattered Potential

Finding the velocity potential for the case of the body fixed to its mean position in the presence of an incident wave is referred to as the *diffraction* problem, although we actually solve for the *scattered* potential, ϕ_S . The diffraction potential itself is the sum of the scattered and incident potentials:

$$\phi^D(\vec{x}, t) = \phi_I(\vec{x}, t) + \phi_S(\vec{x}, t). \quad (41)$$

There is an integral equation which may be solved for the diffraction potential directly but, in general $\phi_S \ll \phi_I$ so it is numerically advantageous to solve for the scattered potential, and then add the known incident potential to this solution.

Equation (24) may be solved for the scattered potential by using the body boundary condition:

$$\frac{\partial \phi_S}{\partial n} = -\frac{\partial \phi_I}{\partial n}, \quad (42)$$

where the incident potential is taken to be the real part of

$$\phi_I(\vec{x}, t) = \frac{g}{\pi} \int_{-\infty}^{\infty} d\omega \frac{i}{\omega_0} \exp[kz - ik[x \cos \beta + y \sin \beta] + i\omega t]. \quad (43)$$

Here the encounter frequency ω is related to the absolute frequency by $\omega = \omega_0 - Uk \cos \beta$. This incident velocity potential is a uni-directional wave system which contains all frequencies, and it describes a wave elevation which is the Dirac function in time, $\delta(t)$, when viewed from the origin of the body-fixed reference frame.

Example free-surface profiles are illustrated in Figure 4 for the case of $U = 0$, $\beta = \pi$, $t = -10, 0, 10$. At times other than $t = 0$, the waves are dispersed over just one half of the free-surface. For any time $t < 0$, the waves are only in the $x > 0$ half-space, while for $t > 0$, the waves are only in the $x < 0$ half-space. In the former, the waves are coalescing to the impulse, and in the latter they are dispersing from the impulse, hence the Fourier components are always ordered such that the wave length increases with $|x|$.

There is no particular significance in choosing to have this temporal impulse occur along a line through the origin. Any convenient location is acceptable as long as it is accounted for in

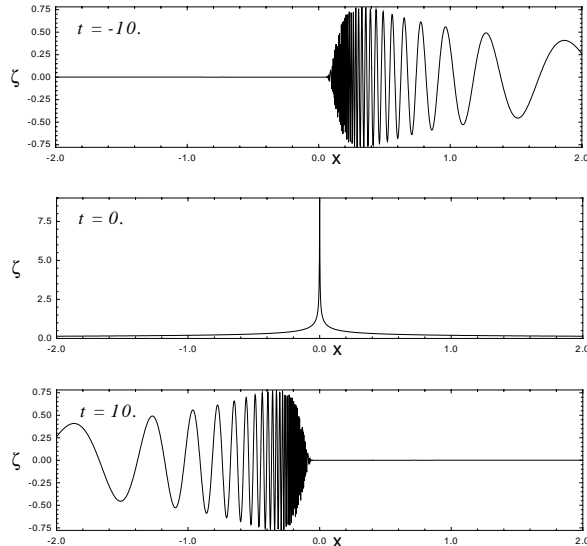


Figure 4: Example of the impulsive incident wave elevation.

the interpretation of the impulse-response function or its Fourier transform. In [10] it is shown that such a shift in the location is equivalent to a phase shift in the frequency domain.

The source strength σ_S , corresponding to the scattered potential ϕ^S , can be computed directly from equation (28), after which equation (29) is used to compute $\nabla\phi^S$.

7.9 First Order “Steady” Forces

In the Neumann-Kelvin linearization of the seakeeping problem, the steady perturbation potential can be considered as the large time limit of a radiation potential forced by impulsive surge acceleration.

$$\bar{\phi}(\vec{x}) = \lim_{t \rightarrow \infty} \phi^{1a}(\vec{x}, t). \quad (44)$$

Thus the steady wave resistance, sinkage force, and trim moment can be obtained from the large time limits of

$$F_j(t) = -\rho \int \int_{\bar{S}_b} dS \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi^{1a} n_j, \quad (45)$$

where $j = 1, 3, 5$ correspond to the x and z components of the force and the moment about the y -axis respectively. Since only the steady-state limits of these forces are of interest, T̄MIT computes only the convective term in this expression

$$F_j(t) = \rho U \int \int_{\bar{S}_b} dS \frac{\partial}{\partial x} \phi^{1a} n_j. \quad (46)$$

These forces are output in non-dimensional form as discussed in §5.2.

7.10 Transient Integrated Quantities

The `potent` module of T̄MIT calculates the time history of the requested radiation and diffraction potentials on each panel of the body due to certain canonical forcing. In order to obtain

the impulse-response functions, the pressures due to these potentials must be integrated over the body surface. One of the jobs of the `respon` module is to perform these integrations.

The canonical radiation and diffraction potentials allow the calculation of the potential on the body due to either an arbitrary motion of the body, or an arbitrary incident wave elevation. For example, the potential due to an arbitrary forced motion of the body in mode k is,

$$\Phi^k = \int_{-\infty}^{\infty} d\tau \phi^k(t - \tau) \dot{x}_k(\tau). \quad (47)$$

The force on the body in mode j due to this arbitrary motion in mode k is found by integrating the consequent linearized pressure over the body surface

$$F_{jk} = -\rho \int \int_{\bar{S}_b} dS \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \Phi^k n_j. \quad (48)$$

Similarly,

$$\Phi^D(\vec{x}, t) = \int_{-\infty}^{\infty} d\tau \phi^D(\vec{x}, t - \tau) \zeta(\tau) \quad (49)$$

is the potential due to an arbitrary history of incident wave elevation, and replacing Φ^k with Φ^D in equation (48) gives the consequent force on the body.

A variant of Stokes' theorem, which has been attributed to Tuck [18], can be used to simplify the calculation of the impulse-response function

$$\int \int_{\bar{S}_b} dS \left[m_j \Phi^k - n_j (\nabla \bar{\Phi} \cdot \nabla \Phi^k) \right] = - \int_{\bar{\Gamma}} dl n_j \Phi^k (\vec{l} \times \hat{n}) \cdot \nabla \bar{\Phi}. \quad (50)$$

Here \vec{l} is the unit vector tangent to the mean waterline, so for a wall sided body the line integral is identically zero. Note however, that in the derivation of this theorem it is assumed that the basis flow (represented by $\bar{\Phi}$) satisfies the no flux condition on the body boundary, and therefore that the theorem is not strictly valid for the Neumann-Kelvin linearization (unless the body is infinitely thin). Nevertheless, the theorem is often applied in practice and TIMIT provides the option of doing so, in which case the force on the body is computed from

$$F_{jk} = -\rho \int \int_{\bar{S}_b} dS \left(n_j \frac{\partial}{\partial t} - m_j \right) \Phi^k. \quad (51)$$

The six-by-six matrix of impulse-response functions which appears on the left hand side of the equations of motion (1) is calculated from the combination of (47) and (48), using the six canonical radiation potentials. By inserting the decomposition of the potential into equation (47), a general radiation potential can be written as

$$\Phi^k = \mathcal{N}_k \dot{x}(t) + \mathcal{M}_k x(t) + \int_{-\infty}^t d\tau \psi^k(t - \tau) \dot{x}_k(\tau) \quad (52)$$

and the time derivative of this potential as

$$\frac{\partial}{\partial t} \Phi^k = \mathcal{N}_k \ddot{x}(t) + \mathcal{M}_k \dot{x}(t) + \psi^k(0) \dot{x}(t) + \int_{-\infty}^t d\tau \frac{\partial}{\partial t} \psi^k(t - \tau) \dot{x}_k(\tau). \quad (53)$$

When these two expressions are inserted into equation (48) the complete radiation impulse-response function can be expressed in terms of the canonical radiation potentials as follows:

$$\begin{aligned}
a_{jk}(\vec{x}) &= \rho \int \int_{\bar{S}_b} dS \mathcal{N}_k n_j \\
b_{jk}(\vec{x}) &= \rho \int \int_{\bar{S}_b} dS \left(\mathcal{M}_k - U \frac{\partial}{\partial x} \mathcal{N}_k \right) n_j \\
c_{jk}(\vec{x}) &= -\rho \int \int_{\bar{S}_b} dS U \frac{\partial}{\partial x} \mathcal{M}_k n_j \\
K_{jk}(t) &= \rho \int \int_{\bar{S}_b} dS \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \psi^k(t) n_j
\end{aligned} \tag{54}$$

where we have used the fact that $\psi^k(\vec{x}, 0) = 0$. If the option of using Tuck's theorem is chosen, then the impulse-response functions are computed from

$$\begin{aligned}
b_{jk}(\vec{x}) &= \rho \int \int_{\bar{S}_b} dS (\mathcal{M}_k n_j - \mathcal{N}_k m_j) \\
c_{jk}(\vec{x}) &= -\rho \int \int_{\bar{S}_b} dS \mathcal{M}_k m_j \\
K_{jk}(t) &= \rho \int \int_{\bar{S}_b} dS \left(n_j \frac{\partial}{\partial t} - m_j \right) \psi^k(t)
\end{aligned} \tag{55}$$

where a_{jk} is the same in either case.

The coefficient a_{jk} is a genuine added-mass coefficient which is independent of both time (or frequency) and forward speed. The coefficients b_{jk} and c_{jk} are, on the other hand, functions of the forward speed. The constants b_{jk} , when computed using Tuck's theorem, can be shown to satisfy the following relations,

$$\begin{aligned}
b_{jk} &= 0 & \text{for } j = k \\
b_{jk} + b_{kj} &= 0 & \text{for } j \neq k.
\end{aligned} \tag{56}$$

When a direct pressure integration is used, $b_{jk} \neq -b_{kj}$ for $j \neq k$ in many cases.

A similar manipulation is used to obtain the expression for the diffraction impulse-response function in terms of the diffraction potential

$$K_{jD}(t) = -\rho \int \int_{\bar{S}_b} dS \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi^D(t) n_j. \tag{57}$$

Or if Tuck's theorem is applied

$$K_{jD}(t) = -\rho \int \int_{\bar{S}_b} dS \left(n_j \frac{\partial}{\partial t} - m_j \right) \phi^D(t). \tag{58}$$

The definitions of the scattering impulse-response functions $K_{jS}(t)$ are analogous.

7.11 Following Seas When $U > 0$

In order to extend this technique to include waves which are incident from abaft the beam, consider the relationship between the frequency of encounter, ω , and the absolute wave frequency, ω_0 : $\omega = \omega_0 - \omega_0^2 \frac{U \cos \beta}{g}$ or

$$\omega_0 = \frac{g}{2U \cos \beta} \left(1 \pm \sqrt{1 - \omega \frac{4U \cos \beta}{g}} \right). \quad (59)$$

For waves incident from ahead of the beam ($90^\circ < \beta < 270^\circ$), $\cos \beta < 0$ and $\omega \geq 0$, so there is only one root to (59) giving a unique relationship between the two frequencies. For waves abaft the beam ($-90^\circ < \beta < 90^\circ$), $\cos \beta > 0$ and there can be as many as three waves producing the same frequency of encounter. When $0 \leq \omega \leq g/(4U \cos \beta)$ there are two roots to (59), and when $-\infty < \omega \leq 0$ there is a third. The first of these roots corresponds to long waves which are overtaking the body

$$\omega_{01} = \frac{g}{2U \cos \beta} \left(1 - \sqrt{1 - \omega \frac{4U \cos \beta}{g}} \right), \quad \begin{cases} 0 \leq \omega \leq g/4U \cos \beta \\ 0 \leq \omega_{01} \leq g/2U \cos \beta \end{cases} \cdot \quad (60)$$

The second root corresponds to waves which appear to be overtaking the body, because the phase velocity is greater than U , but in fact are not because the group velocity is less than U

$$\omega_{02} = \frac{g}{2U \cos \beta} \left(1 + \sqrt{1 - \omega \frac{4U \cos \beta}{g}} \right), \quad \begin{cases} 0 \leq \omega \leq g/4U \cos \beta \\ g/2U \cos \beta \leq \omega_{02} \leq g/U \cos \beta \end{cases} \cdot \quad (61)$$

A negative frequency of encounter simply results in a change in the phase, so there is a third root corresponding to waves which have both phase and group velocities which are less than U

$$\omega_{03} = \frac{g}{2U \cos \beta} \left(1 + \sqrt{1 + |\omega| \frac{4U \cos \beta}{g}} \right), \quad \begin{cases} -\infty < \omega \leq 0 \\ g/U \cos \beta \leq \omega_{03} < \infty \end{cases} \cdot \quad (62)$$

Since the integral equation is on the body's surface, that is, the problem is solved in the body-fixed reference frame, the relevant frequency is the encounter frequency, and to analyze the body's motions in waves incident from abaft the beam, the three ranges of frequencies identified above must be treated separately.

In following seas, the incident wave in equation (43) is split into three parts, each of which contains frequencies from only one of the three ranges in (60), (61), and (62)

$$\phi^I(\vec{x}, t) = \sum_{m=1}^3 \phi^{Im} = \sum_{m=1}^3 \Re \left\{ \frac{ig}{\pi} \int d\omega_{0m} \frac{1}{\omega_{0m}} \left(1 - \frac{2\omega_{0m}}{g} \right) e^{\frac{\omega_{0m}^2}{g} \{z - i[(x+Ut) \cos \beta + y \sin \beta]\}} e^{i\omega_{0m} t} \right\}. \quad (63)$$

The first spatial and temporal derivatives of these potentials can be related to the complex error function and computed using algorithms found in [7]. Figure 5 illustrates the incident-wave elevations for each of the three parts of the potential.

For waves abaft the beam, the three incident wave potentials are used to solve three diffraction problems, resulting in three impulse-response functions for a single wave heading. Consequently the expression for the exciting forces appearing in (3) becomes the sum of three separate convolutions

$$F_j(t) = \sum_{m=1}^3 \int_{-\infty}^{\infty} K_{jDm}(t - \tau) \zeta_m(\tau) d\tau \quad (64)$$

Here the actual incident wave record which is used for the simulation must also be decomposed into three incident wave records, each each of which contains frequencies from only one of the three ranges. (This means that when a simulation is performed in following seas, the user must give the time histories of ζ_m , $m = 1, 2, 3$ as input to the `respon` module.) Given an arbitrary time history of incident wave elevation at some point in an *earth*-fixed coordinate system, $\zeta_0(t)$, the necessary decomposition into $\zeta_m(t)$ can be made using

$$\zeta_m(t) = \Re \left\{ \frac{1}{\pi} \int d\omega_{0m} \tilde{\zeta}_0(\omega_{0m}) e^{i\omega_{0m} \left(1 - \frac{\omega_{0m} U \cos \beta}{g}\right) t} \right\}, \quad m = 1, 2, 3; \quad (65)$$

where $\tilde{\zeta}_0(\omega_0)$ is the Fourier transform of $\zeta_0(t)$ with respect to absolute wave frequency. Equation (65) is equivalent to the transformation presented in [8], but is computationally more convenient since it contains no singularities.

7.12 The Haskind Relations

The exciting force impulse-response function, in a given mode, may also be computed from the solution of the radiation problem for that mode. The relations between the solutions of the radiation problems and the exciting forces are known as the Haskind relations. See [13] and [14] for these relations in the frequency domain for the zero- and nonzero-forward-speed cases respectively. The zero-speed relations are computationally efficient because a single radiation solution can be used to compute the exciting forces for multiple angles of wave incidence without solving any additional hydrodynamic problems. Moreover, these relations are exact. For forward speed, the relations are less efficient because a fictitious boundary-value problem must be solved, and their use is less attractive since they are only approximate. If more specific aspects of the diffraction problem are required, such as wave elevations or fluid velocities, then the scattering problem must be solved in any case.

The time-domain Haskind relations for zero speed are derived in [19] and in our notation may be written

$$K_{kH}(t, \beta) = -\rho \left\{ \iint_{\bar{S}_b} d\vec{\xi} \left[\phi_t^I(\vec{\xi}, t, \beta) n_k(\vec{\xi}) - \phi_{nt}^I(\vec{\xi}, t, \beta) \mathcal{N}_k(\vec{\xi}) \right] - \int_{-\infty}^t d\tau \iint_{\bar{S}_b} d\vec{\xi} \phi_{nt}^I(\vec{\xi}, \tau, \beta) \psi^k(\vec{\xi}, t - \tau) \right\}, \quad (66)$$

where $\mathcal{N}_k(\vec{x})$ and $\psi^k(\vec{x}, t)$ are the solution to the radiation problem for mode k . Note that the first term in the first integral is the Froude-Krylov contribution; evidently the remaining terms are due to scattering.

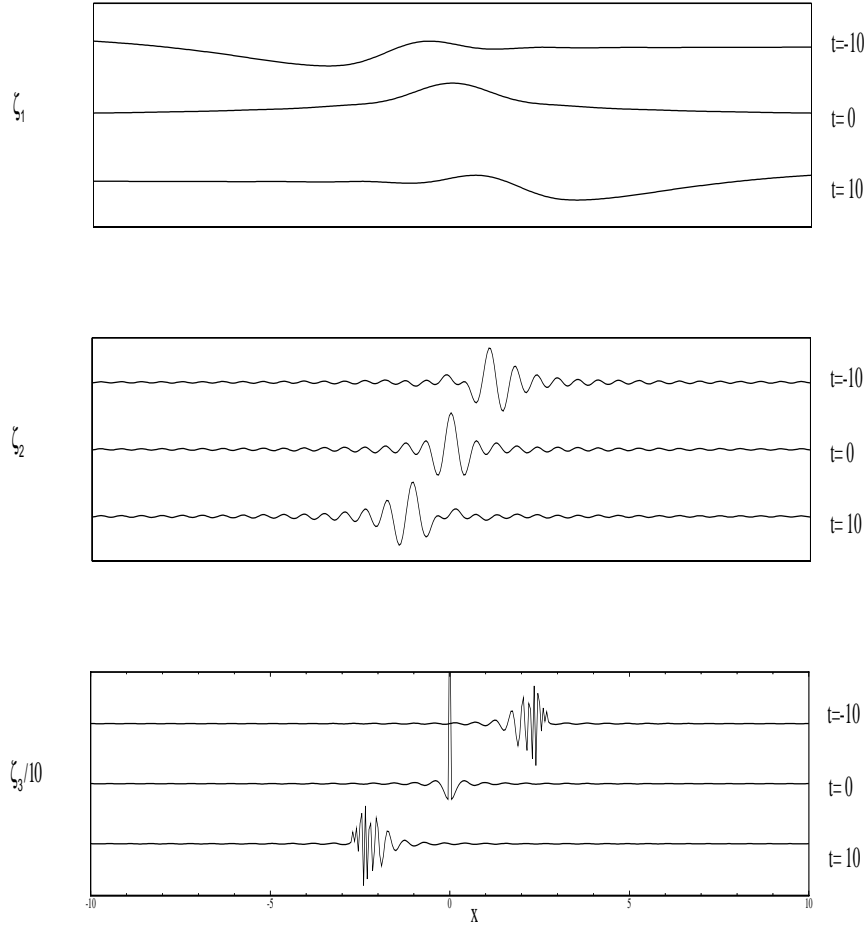


Figure 5: The three parts of the impulsive incident wave elevation shown in a cut along the x -axis. The waves are incident from $\beta = 0^\circ$, and evaluated at $y = 0, z = -.01$ for three different points in time (shifted vertically to show the progression in time). (Note that the vertical scale used for ζ_3 differs from the other two by a factor of ~ 10 .) The coordinate system is fixed to the body which is traveling at $Fn = 0.3$ in the positive x direction (time is non-dimensionalized by $(g/L)^{\frac{1}{2}}$ and lengths by the body length L .)

7.13 Transient Local Quantities

Local quantities such as the fluid velocity and pressure have decompositions which parallel the integrated quantities and impulse-response functions may be defined for these local fluid quantities.

The dynamic perturbation pressure at any point in the fluid is given by Bernoulli's equation (18) which is repeated here

$$p(\vec{x}, t) = -\rho \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi \quad (67)$$

where ϕ is again used to represent any one of the canonical perturbation potentials. The corresponding first-order wave elevation at a point on the $z = 0$ plane is obtained from the dynamic free-surface condition

$$\zeta(\vec{x}, t) = \frac{-1}{g} \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi \Big|_{z=0}, \quad (68)$$

where ζ represents any of the first-order perturbation wave elevations. In TĪMIT, the wave elevation along the waterline of the body is computed from the value of the potential and its gradient at the centroid of the panels bordering the free-surface.

A general radiation pressure in mode k at the point \vec{x} on the body surface due to an arbitrary motion of the body $x_k(t)$, can be expressed in terms of the canonical radiation potentials as

$$p_k(\vec{x}, t) = a_{kp} \ddot{x}_k(t) + b_{kp} \dot{x}_k(t) + c_{kp} x_k(t) + \int_{-\infty}^t d\tau K_{kp}(\vec{x}, t - \tau) \dot{x}_k(\tau), \quad (69)$$

where the radiation pressure impulse-response function on the body is

$$\begin{aligned} a_{kp}(\vec{x}) &= -\rho \mathcal{N}_k \\ b_{kp}(\vec{x}) &= -\rho \left(\mathcal{M}_k - U \frac{\partial}{\partial x} \mathcal{N}_k \right) \\ c_{kp}(\vec{x}) &= \rho U \frac{\partial}{\partial x} \mathcal{M}_k \\ K_{kp}(\vec{x}, t) &= -\rho \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \psi^k. \end{aligned} \quad (70)$$

Similarly, we can write a general diffraction pressure at the same point as

$$p_D(\vec{x}, t) = \int_{-\infty}^{\infty} d\tau K_{Dp}(\vec{x}, t - \tau, \beta) \zeta(\tau) \quad (71)$$

where the diffraction pressure impulse-response function is

$$K_{Dp}(\vec{x}, t, \beta) = -\rho \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi^D. \quad (72)$$

Using equation (68) we can define the analogous radiation and diffraction wave elevations, and wave elevation impulse-response functions at the point \vec{x} along the body waterline.

$$\zeta_k(t) = a_{k\zeta} \ddot{x}_k(t) + b_{k\zeta} \dot{x}_k(t) + c_{k\zeta} x_k(t) + \int_{-\infty}^t d\tau K_{k\zeta}(\vec{x}, t - \tau) \dot{x}_k(\tau) \quad (73)$$

where,

$$\begin{aligned}
a_{k\zeta}(\vec{x}) &= -\frac{1}{g} \mathcal{N}_k \\
b_{k\zeta}(\vec{x}) &= -\frac{1}{g} \left(\mathcal{M}_k - U \frac{\partial}{\partial x} \mathcal{N}_k \right) \\
c_{k\zeta}(\vec{x}) &= \frac{U}{g} \frac{\partial}{\partial x} \mathcal{M}_k \\
K_{k\zeta}(\vec{x}, t) &= -\frac{1}{g} \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \psi^k
\end{aligned} \tag{74}$$

and

$$\zeta_D(t) = \int_{-\infty}^{\infty} d\tau \ K_{D\zeta}(\vec{x}, t - \tau, \beta) \zeta(\tau) \tag{75}$$

where

$$K_{D\zeta}(\vec{x}, t, \beta) = -\frac{1}{g} \left(\frac{\partial}{\partial t} - U \frac{\partial}{\partial x} \right) \phi^D. \tag{76}$$

A general radiation velocity vector at the point \vec{x} on the body can be expressed in terms of gradients of the canonical radiation potentials at that point by

$$\nabla \Phi^k(\vec{x}, t) = \nabla \mathcal{N}_k \dot{x}_k(t) + \nabla \mathcal{M}_k x_k(t) + \int_{-\infty}^t d\tau \ \nabla \psi^k(\vec{x}, t - \tau) \dot{x}_k(\tau). \tag{77}$$

Gradients of the canonical radiation potentials, as they appear in equation (77), will be referred to as the radiation velocity impulse-response functions. Similarly, a general diffraction velocity vector at the point \vec{x} is given by

$$\nabla \Phi^D(\vec{x}, t, \beta) = \int_{-\infty}^{\infty} d\tau \ \nabla \phi^D(\vec{x}, t - \tau, \beta) \zeta(\tau), \tag{78}$$

where the diffraction velocity impulse-response function is $\nabla \phi^D(\vec{x}, t, \beta)$.

7.14 Asymptotic Continuation

At zero forward speed, the canonical potentials decay sufficiently rapidly with time that truncation after an appropriate time is an effective strategy. At non-zero forward speed however, the radiation potential $\psi^k(t)$ may exhibit large-time behavior which complicates the use of the memory function $K_{jk}(t)$ in convolutions and Fourier transforms. For some modes of motion, $K_{jk}(t)$ tends to a constant as $t \rightarrow \infty$. For these modes, the constant which remains in $K_{jk}(t)$ at large time should be evaluated, subtracted off, and added to c_{jk} .

Computational efficiency dictates that the hydrodynamic problems should be solved over as short a time range as possible. However due to forward speed, the memory functions display a slowly decaying oscillation in time at the non-dimensional critical frequency $\omega_c = (4F_n)^{-1}$ where $F_n = U/\sqrt{gL}$ is the Froude number. Our approach is to truncate the computations as soon as there is sufficient information in each memory function to accurately estimate this large-time behavior. We assume, based on the Green function, that the large-time oscillation of the memory function is of the form

$$K_{jk}(t) \sim a_0 + \frac{1}{t} [a_1 \cos(\omega_c t) + a_2 \sin(\omega_c t)], \tag{79}$$

as $t \rightarrow \infty$. The constants in equation (79) can be determined by a least squares fit, so that integrations involving K_{jk} may be divided into two ranges: one which terminates at the end of the computed time-history, and one which continues with the asymptotic form (79). A more precise form for the solution, as $t \rightarrow \infty$, is given [12] by

$$\phi \sim C_0 + \Re \left(C_1 \frac{e^{-\alpha t}}{t} + C_2 \frac{1}{t^2} \right) e^{i\omega_c t}, \quad (80)$$

in which C_i and α are parameters depending upon the body geometry and the forward speed. However this distinction is unimportant at realistic body speeds [1].

A similar procedure may be used when using the radiation impulse-response functions for local quantities.

7.15 Time Harmonic Integrated Quantities

If the body motion and the incident wave elevation are assumed to be time harmonic at the frequency of encounter ω (*i.e.*)

$$x_k(t) = \Re \xi_k(\omega, \beta) e^{i\omega t}, \quad \text{and} \quad \zeta(t) = \Re \mathcal{A}(\omega, \beta) e^{i\omega t}, \quad (81)$$

then as $t \rightarrow \infty$ the equations of motion can be written

$$\sum_{k=1}^6 \left\{ -\omega^2 [M_{jk} + A_{jk}(\omega)] + i\omega B_{jk}(\omega) + C_{jk} + \bar{c}_{jk} \right\} \xi_k(\omega, \beta) = X_j(\omega, \beta). \quad (82)$$

The frequency-dependent coefficients in equation (82) are related to the impulse-response functions through the Fourier transforms

$$A_{jk}(\omega) = a_{jk} + \Re \frac{-i}{\omega} \int_0^\infty dt K_{jk}(t) e^{-i\omega t}, \quad (83)$$

$$B_{jk}(\omega) = b_{jk} + \Im i \int_0^\infty dt K_{jk}(t) e^{-i\omega t}, \quad (84)$$

$$X_j(\omega, \beta) = \int_{-\infty}^\infty dt K_{jD}(t, \beta) e^{-i\omega t}. \quad (85)$$

At zero forward speed, the exciting force coefficients may also be computed via the Haskind relations and the following transform

$$X_j(\omega, \beta) = \int_{-\infty}^\infty dt K_{jH}(t, \beta) e^{-i\omega t}. \quad (86)$$

These integrals are treated as suggested in §7.14. If the memory function tends to a constant at large time, this is subtracted off and added to c_{jk} . Then, the contributions to (83) from the computed range of the memory functions are evaluated numerically by Filon quadrature. The remaining contributions, associated with the oscillatory terms in (79), are expressed in terms of the sine and cosine integrals.

In a physical experiment, the forces associated with c_{jk} are usually measured together with the added-mass, so these terms are included with the added-mass coefficient output from TIMIT :

$$\bar{A}_{jk} = \frac{A_{jk}}{\rho L^n} - \frac{\bar{c}_{jk}}{\omega^2}. \quad (87)$$

7.16 Time Harmonic Local Quantities

Local quantities in the frequency domain such as the fluid velocity and pressure are usually required in the form of response-amplitude operators (RAO). RAO's may be computed by two alternative methods. One method would be to conduct a simulation in the time domain with a time-history of (pseudo-)random wave elevation as excitation and compute a time history of a local quantity as the sum of a convolution of the body motion with the radiation impulse-response function for the quantity and a convolution of the incident wave with the diffraction impulse-response function for the quantity (these impulse-response functions are defined in §7.13). Then the response-amplitude operator is the result of dividing the Fourier transform of the response by the Fourier transform of the excitation. While this technique provides a useful check of time-history data from simulations, it is not the method used in TIMIT to compute local response-amplitude operators. The more direct approach is used, which consists of working directly with the Fourier transforms of the canonical potentials and the body motion response-amplitude operators. The RAO's for the fluid pressure and velocity are defined in terms of the frequency domain potentials by:

$$p(\vec{x}, \omega) = \left(\omega - U \frac{\partial}{\partial x} \right) (\phi^D(\vec{x}, \omega) + \phi^j(\vec{x}, \omega) \xi_j(\omega)), \quad (88)$$

and

$$\nabla \phi(\vec{x}, \omega) = \nabla \phi^D(\vec{x}, \omega) + \nabla \phi^j(\vec{x}, \omega) \xi_j, \quad (89)$$

in which the repeated indices indicate summation.

It remains only to define the Fourier transforms of the canonical potentials. The Fourier transform for the diffraction potential is defined by

$$\phi^D(\vec{x}, \omega) = \int_{-\infty}^{\infty} dt \phi^S(\vec{x}, t) e^{-i\omega t} + \frac{ig}{\omega_0} e^{kz} e^{-ik(x \cos \beta + y \sin \beta)}. \quad (90)$$

The Fourier transform for the radiation potential is defined by

$$\phi^j(\vec{x}, \omega) = \mathcal{N}_j(\vec{x}) + \frac{1}{i\omega} \mathcal{M}_j(\vec{x}) + \int_{-\infty}^{\infty} dt \psi^j(\vec{x}, t) e^{-i\omega t}. \quad (91)$$

7.17 Second Order Steady Forces

The second-order steady forces are frequency-domain quantities. Moreover, they must be computed from quadratic products of local frequency-domain quantities. For the Neumann-Kelvin linearization, the second-order steady force is:

$$\begin{aligned} \vec{F}^{(2)}(\omega) = & \frac{-\rho \iint_{\bar{S}_b} \hat{n} \left[\frac{1}{2} (\nabla \phi)^2 + (\vec{\xi} + \vec{\alpha} \times \vec{x}) \cdot \nabla (\phi_t - U \phi_x) \right]}{dS} \\ & - \frac{\rho \iint_{\bar{S}_b} \hat{n} [(\vec{\alpha} \times \hat{n})(\phi_t - U \phi_x)]}{dS} \\ & + \frac{\rho g}{2} \int_{\bar{\Gamma}} \hat{n} (\zeta - \xi_3 - \alpha_1 y + \alpha_2 x)^2 dl \\ & + F_H^2 \\ & - \rho U \iint_{\bar{S}_b} \hat{n} \phi^{(2)} dS, \end{aligned} \quad (92)$$

and the second-order steady moments are:

$$\begin{aligned}
\vec{M}^{(2)}(\omega) = & \overline{-\rho \iint_{\bar{S}_b} (\vec{x} \times \hat{n}) \left[\frac{1}{2}(\nabla\phi)^2 + (\vec{\xi} + \vec{\alpha} \times \vec{x}) \cdot \nabla(\phi_t - U\phi_x) \right]} dS \\
& - \overline{\rho \iint_{\bar{S}_b} (\vec{x} \times \hat{n}) [(\vec{\alpha} \times \hat{n})(\phi_t - U\phi_x)]} dS \\
& + \frac{\rho g}{2} \int_{\bar{\Gamma}} (\vec{x} \times \hat{n}) (\zeta - \xi_3 - \alpha_1 y + \alpha_2 x)^2 dl \\
& + M_H^2 \\
& - \rho U \iint_{\bar{S}_b} (\vec{x} \times \hat{n}) \phi^{(2)} dS.
\end{aligned} \tag{93}$$

In these equations the over-bar denotes the average over one period and (with one exception noted below) all of the quantities in the integrand are frequency-domain quantities at a particular frequency. Most of the quantities in the integrands have been defined previously. Quantities particular to these expressions are: \vec{x} , the local position vector; ϕ , the sum of the perturbation potentials; and $\vec{\xi}$ and $\vec{\alpha}$, the linear and angular first-order body motions. F_H^2 and M_H^2 are the second-order hydrostatic force and moment. $\phi^{(2)}$ is the second-order steady potential.

Examining equation (92) or (93) term-by-term, reveals that the first term in the first line is the contribution from the dynamic pressure; the second term in the first line and the entire second line are the contributions from body's motion effect on the pressure and first-order force (and moment) computed on the mean body surface; the third line is the contribution from the vertical wave elevation and vertical motion of the body, which change the wetted surface in the waterline region; the fourth line is the contribution from the hydrostatic corrections resulting from the body motions; and the last line is the contribution from the second-order steady potential.

The Version 3.* release of T̄MIT does not support the contributions from the second-order steady potential $\phi^{(2)}$, or the double spatial derivative in the first line $U\nabla\phi_x$.

The evaluation of (92) and (93) proceeds by first computing all of the necessary frequency-domain quantities via Fourier transform from the canonical solutions in the time domain, and then taking the mean value of the quadratic products by the usual manipulation of the complex quantities.

7.18 Exact Equations of Motion and Nonlinear Hydrostatics

Simulations of body motions in the time domain may include the nonlinear hydrostatic force (and moment) and exact inertia effects. This is an extension of equation (1). For nonlinear hydrostatics, the hydrostatic pressure is integrated over the instantaneous position of the body below the plane $z_0 = 0$. Nonlinear inertia effects are included by temporal integration of the Euler rotations instead of the linearized rotations. This is often referred to as the “exact equations of motion,” although it is important to remember that the hydrodynamic forcing is still the linearized radiation and diffraction forcing.

The displacement of a rigid body is described by a position vector and a set of Euler angles. We define the sequence of rotations to be: yaw (ξ_6) \rightarrow pitch (ξ_5) \rightarrow roll (ξ_4). This gives the

following rotational transformation matrix

$$L = \begin{pmatrix} c_6 c_5 & s_6 c_5 & -s_5 \\ -s_6 c_4 + c_6 s_4 s_5 & c_6 c_4 + s_4 s_5 s_6 & s_4 c_5 \\ s_6 s_4 + c_6 c_4 s_5 & -c_6 s_4 + s_6 s_5 c_4 & c_5 c_4 \end{pmatrix} \quad (94)$$

where,

$$\begin{aligned} c_4 &= \cos(\xi_4) & s_4 &= \sin(\xi_4) \\ c_5 &= \cos(\xi_5) & s_5 &= \sin(\xi_5) \\ c_6 &= \cos(\xi_6) & s_6 &= \sin(\xi_6). \end{aligned}$$

Angular velocity resolved into the body frame is

$$\vec{\omega}' = \begin{bmatrix} -\dot{\xi}_6 s_5 + \dot{\xi}_4 \\ \dot{\xi}_6 s_4 c_5 + \dot{\xi}_5 c_4 \\ \dot{\xi}_6 c_4 c_5 + \dot{\xi}_5 s_4 \end{bmatrix} \quad (95)$$

Thus, the time rate change of the Euler angles may be related to angular velocity by the following matrix multiplication,

$$\dot{\phi} = \mathbf{W} \vec{\omega}' = \begin{bmatrix} 1 & s_4 s_5 / c_5 & c_4 s_5 / c_5 \\ 0 & c_4 & -s_4 \\ 0 & s_4 / c_5 & c_4 / c_5 \end{bmatrix} \vec{\omega}'. \quad (96)$$

7.18.1 Exact equations of motion

The equations governing the motion of a rigid body in its six modes may be derived by applying Newton's Second Law at the body's center of gravity. The angular momentum is a free vector which is most easily defined in the body frame but must be differentiated in an inertial reference frame. The six equations of motion may be written in matrix form as

$$\mathbf{E} \begin{bmatrix} \dot{\vec{v}}_{o'} \\ \dot{\vec{\omega}}' \end{bmatrix} = \vec{q} \quad (97)$$

where,

$$\mathbf{E}_{6 \times 6} = \begin{bmatrix} \mathbf{m} & -[\vec{r}_g \times \mathbf{m}] \mathbf{L}^T \\ \vec{r}_g \times \mathbf{m} & \mathbf{L}^T \mathbf{I}_g' - [\vec{r}_g \times (\vec{r}_g \times \mathbf{m})] \mathbf{L}^T \end{bmatrix} \quad (98)$$

and

$$\vec{q} = \begin{bmatrix} \vec{F} - m\{\vec{\omega} \times (\vec{\omega} \times \vec{r})\} \\ \vec{M} - m\{\vec{r}_g \times (\vec{\omega} \times (\vec{\omega} \times \vec{r}_g))\} - \mathbf{L}^T (\vec{\omega}' \times \mathbf{I}_g' \vec{\omega}') \end{bmatrix}. \quad (99)$$

Here, \mathbf{I}_g' is the moment of inertia about the center of gravity expressed in the body-fixed frame; \mathbf{m} is a 3×3 matrix with the body mass on its diagonal; and \vec{r}_g is a free vector from the body origin to the center of gravity. These six equations reduce to linear form if the rotational transformation matrix is set to identity and higher-order terms in \vec{q} are neglected.

7.18.2 Nonlinear hydrostatics

The nonlinear restoring force is computed by integrating the hydrostatic pressure over the instantaneous wetted surface and adding the body weight

$$\vec{F} = \vec{F}_b + \vec{W} = \rho g \int \int_{S_b(t)} (z\hat{n} - mg\hat{k})dS \quad (100)$$

The restoring moment (about the body origin) due to gravity and buoyancy is computed directly in the earth-fixed frame by

$$\vec{M} = (\vec{r}_g \times \vec{W}) + (\vec{r}_b \times \vec{F}_b) \quad (101)$$

where \vec{r}_b and \vec{r}_g are vectors from the body origin to the centers of buoyancy and gravity, respectively. All vectors are resolved into the earth-fixed frame.

7.19 Generalized Modes

The fact that the initial-boundary-value problem is linearized allows the hydrodynamic analysis of any modes which one cares to define. Beyond the six rigid-body modes, modes describing body deformations (perhaps modeled as a beam), modes describing rigid sections connected by compliant couplings, and modes describing multiple bodies are typically of interest. In the case where the modes describe non-rigid motions, the analysis may be hydroelastic or if the motion in the non-rigid modes is small compared to that in the rigid modes the radiation by these modes may be neglected.

The details of generalized modal analysis are too lengthy to present here; see [17, 6] for the theory and example applications.

7.20 Numerical Solution

The integral equation (24) is discretized spatially by subdividing the surface \bar{S}_b into N planar quadrilateral (or triangular) panels, on which the potential is assumed to be constant. A system of N linear equations is generated by collocation, satisfying the discretized form of (24) at the panel centroids. The integrals over each panel involving $G^{(0)}$ and its normal derivative are performed using the algorithms in [15]. The time-dependent function $G^{(f)}$ and its derivatives are evaluated for each pair of panel centroids using the algorithms described in [16].

The solution of the diffraction problem for $U = 0$ at arbitrary heading angle and for $\frac{\pi}{2} < \beta < \frac{3\pi}{2}$ (head seas) when $U > 0$, is initiated smoothly at an appropriate negative time, before there is significant wave amplitude in the vicinity of the body. The computation is terminated at a similarly appropriate positive time. When $U > 0$ and $-\frac{\pi}{2} < \beta < \frac{\pi}{2}$ (following seas) the situation is more complicated (See §7.11) and it is not practical to start the problem with effectively zero wave amplitude in the vicinity of the body. Further details of the numerical aspects of the diffraction problem may be found in [11].

The solution for either $\phi^S(t)$ or $\psi^k(t)$ is carried out at a sequence of equal time-steps Δt . The convolution integrals are evaluated using the trapezoidal rule, with the same Δt . Since $G_t(0) = 0$, there is no contribution to the convolutions in (24) at the upper limit of integration ($\tau = t$). Thus the convolution terms involving G_t only require values of the potential at preceding time-steps, and can all be placed on the right-hand side of the linear system as known quantities. However there is a contribution to the left-hand side from the convolution of

the waterline integral on the third and fourth lines, since $G_{tt}(0) \neq 0$. Thus there are two left-hand-side matrices, one for $t = 0$ and the other for $t > 0$, but since these are each independent of time they only need to be evaluated and factored once.

The dominant computational burden is the evaluation of the convolution integrals at each time-step. Since the time-steps are equal and the coordinate system is fixed to the translating body, values of $G^{(f)}$ (and its derivatives) can be saved to avoid redundant computation, but the large number of such values (proportional to the product of N^2 and the number of time-steps) usually requires disk storage with relatively slow access time for retrieval. If the storage required exceeds the available disk space, the values of $G^{(f)}$ must be re-computed at each time-step.

For conventional bodies with port-starboard symmetry, the potentials can be decomposed into symmetric and antisymmetric components to reduce the number of unknowns by a factor of two. The left-hand side work and storage is then reduced by a factor of four, and the burden of evaluating the right-hand-side convolution integrals is reduced by a factor of two. Since the latter is dominant, the overall reduction in computational cost is effectively one-half.

Once the potentials resulting from the impulsive forcing are known, impulse-response functions for the radiation and diffraction forces are computed by integration of the pressure (18) over the body surface using the same discretization as above.

The impulse-response functions provide the complete hydrodynamic characterization of the body hull and are the basis for subsequent time-domain simulation or frequency-domain analysis. Further computations made using the impulse-response functions entail computational burdens which are relatively small.

The temporal integration of the equations of motion (1) is carried out by a fourth-order Runge-Kutta scheme. Fourier transforms of the impulse-response functions are evaluated by Filon quadrature.

References

- [1] H. B. Bingham. *Simulating ship motions in the time domain*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 1994.
- [2] H. B. Bingham, F. T. Korsmeyer, and J. N. Newman. Predicting the seakeeping characteristics of ships. In *Twentieth Symp. on Naval Hydrodynamics*, Santa Barbara, CA, 1994. ONR.
- [3] H. B. Bingham, F. T. Korsmeyer, J. N. Newman, and G. E. Osborne. The simulation of ship motions. In *6th Int. Conf. on Numerical Ship Hydrodynamics*, Iowa City, 1993.
- [4] A. K. Bratland, F. T. Korsmeyer, and J. N. Newman. Time domain calculations in finite water depth. In *12th Int. Workshop on Water Waves and Floating Bodies*, Marseilles, 1997.
- [5] W. E. Cummins. The impulse response function and ship motions. *Schiffstechnik*, 9:101–109, 1962.
- [6] T. H. Farstad. Transient seakeeping analysis using generalized modes. Master’s thesis, MIT, Cambridge, MA, 1997.
- [7] W. Gautschi. The complex error function. *Collected Algorithms from CACM*, 1969.
- [8] B. W. King. Time-domain analysis of wave exciting forces on ships and bodies. Technical Report 306, The Department of Naval Architecture and Marine Engineering, The University of Michigan, Ann Arbor, 1987.
- [9] B. W. King, R. F. Beck, and A. R. Magee. Seakeeping calculations with forward speed using time domain analysis. In *Seventeenth Symp. on Naval Hydrodynamics*, The Hague, Netherlands, 1988.
- [10] F. T. Korsmeyer. The time domain diffraction problem. In *The Sixth Int. Workshop on Water Waves and Floating Bodies*, Woods Hole, 1991.
- [11] F. T. Korsmeyer and H. B. Bingham. The forward speed diffraction problem. *J. Ship Research*, 42(2):99–112, June 1998.
- [12] Y. Liu and D. K. P. Yue. On the solution near the critical frequency for an oscillating and translating body in or near a free surface. *J. of Fluid Mechanics*, 254:251–266, 1993.
- [13] J. N. Newman. The exciting forces on fixed bodies in waves. *J. Ship Research*, 6(4):10–17, 1962.
- [14] J. N. Newman. The exciting forces on moving bodies in waves. *J. Ship Research*, 9(3):190–199, 1965.
- [15] J. N. Newman. Distributions of sources and normal dipoles over a quadrilateral panel. *J. Engrg. Math.*, 20(2):113–126, 1986.

- [16] J. N. Newman. The approximation of free-surface Green functions. In P. A. Martin and G. R. Wickham, editors, *Wave Asymptotics*, pages 107–135. Cambridge University Press, 1992.
- [17] J. N. Newman. Wave effects on deformable bodies. *Applied Ocean Research*, 16:47–59, 1994.
- [18] T. F. Ogilvie and E. O. Tuck. A rational strip theory for ship motions, part 1. Technical Report 013, The Department of Naval Architecture and Marine Engineering, The University of Michigan, Ann Arbor, 1969.
- [19] J. V. Wehausen. Initial value problem for the motion in an undulating sea of a body with fixed equilibrium position. *J. of Engineering Mathematics*, 1:1–19, 1967.
- [20] J. V. Wehausen and E. V. Laitone. Surface waves. In *Handbuch der Physik*, pages 446–778. Springer, 1960.

A Description of the Code Structure

This appendix contains an outline of the structure of the TIMIT modules `potent` and `respon`. The `potent` module solves the canonical radiation and diffraction problems; and the `respon` module uses the canonical potentials (and velocities) to compute global and local impulse response functions, perform a simulation in the time domain, and Fourier transform transient quantities to obtain frequency dependent coefficients. All of the TIMIT code is written in FORTRAN. A complete TIMIT distribution must have at least the following files.

<code>tdpot.f</code>	Subroutines used only in the <code>potent</code> module.
<code>tdres.f</code>	Subroutines used only in the <code>respon</code> module.
<code>mdfp.f</code>	Subroutines used in the <code>respon</code> module for mean force and moment.
<code>tdio.f</code>	Subroutines used in both modules, typically for input and output.
<code>dattim.f</code>	Subroutine to call the system date and time routines.
<code>geom.f</code>	Subroutines used to compute geometric and hydrostatic quantities.
<code>rpan.f</code>	Subroutine used to compute the Rankine part of the Green function.
<code>vtgn98.f</code>	Subroutines used to compute the free-surface part of the Green function.
<code>EVALPV.DAT</code>	ASCII Data used to compute the free-surface Green function.
<code>iphi.f</code>	Subroutines used to compute the impulsive incident potential.

A.1 A Task Oriented Outline

- `potent` module
 - **Preliminary Processing:** Read input, non-dimensionalize, check logic, set up run, compute geometric quantities, compute the impulsive incident wave.
 - **If `IPOT = 1` (Potential Formulation):** Compute radiation and diffraction potentials via the potential formulation (equation (24)). Write output to `TIMIT.P2R` every ten time steps.
 - **If `ISOR = 1` or `IVEL = 1` (Source Formulation):** Compute radiation and diffraction source strengths via the source formulation (equation (28)).
 - **If `ISOR = 1` and `IPOT = 0` (Potentials from Source Strengths):** Compute the potentials from the source strengths via the single-layer formulation (equation (27)). Write output to `TIMIT.P2R` every ten time steps.
 - **If `IVEL = 1` (Velocities from Source Strengths):** Compute the gradients of the potentials via the gradient of the single-layer formulation (equation (29)). Write output to `TIMIT.P2R` every ten time steps.
 - **Output:** Write results to `TIMIT.P2R`.

- **respon** module
 - **Preliminary Processing:** Read input, non-dimensionalize, check logic, set up run, compute geometric and hydrostatic quantities, compute the impulsive incident wave.
 - **Transient Quantities**
 - * **if ITOPTN(6 or 7) or IFOPTN(5) > 0** Write supplementary output files of panel attributes: `itoptn.pnl` and/or `ifoptn.pnl`.
 - * **If ITOPTN(1) = 1:** Compute the time histories of first order calm water force and moment (equation (46)).
 - * **If any of ITOPTN(I)=1, I=2,5; or IFOPTN(I)=1, I=1,4:** Compute the radiation impulse-response functions (equation (54)).
 - * **If ITOPTN(3)=1 or IFOTPN(2)=1 and $U = 0$:** Compute the exciting force impulse-response functions via the Haskind relations (equation (66)).
 - * **If any of ITOPTN(I)=1, I=4,5; or IFOPTN(I)=1, I=3,4:** Compute the exiting force impulse-response functions (equation (57)).
 - * **If ITOPTN(5) > 0:** Compute the body motions forced by the time history of incident wave elevation in the IWF by solving the equations of motion (1) or (97).
 - * **If ITOPTN(5) > 0 and ITOPTN(6) = 1:** Compute the time history of the fluid pressure on the panels specified in the RCF.
 - * **If ITOPTN(5) > 0 and ITOPTN(7) = 1:** Compute the time history of the fluid velocity on the panels specified in the RCF.
 - **Time Harmonic Quantities**
 - * **If IFOPTN(I)=1, I=1,4:** Compute the added mass and damping coefficients (equation (83)).
 - * **If IFOTPN(2)=1 and $U = 0$:** Compute the exciting force coefficients via the Haskind relations (equation (86)).
 - * **If IFOPTN(I)=1, I=3,4:** Compute the exiting force coefficients (equation (83)).
 - * **If IFOPTN(4) = 1:** Compute the body response amplitude operators (RAO) (equation (82)).
 - * **If IFOPTN(5) = 1, or 3:** Compute the fluid pressure on every panel.
 - * **If IFOPTN(5) = 2, or 3:** Compute the fluid velocity on every panel.
 - * **If IFOPTN(9) = 1:** Compute the second-order steady force on the body by pressure integration.

A.2 A Subroutine Oriented Outline

A listing of the major subroutine names in the order in which they are called can be found at the top of the files `tdpot.f` and `tdres.f`. The listings are reproduced here, providing subroutine names, the files in which they may be found, and a short description of their function. Basic utility routines which zero arrays, compute derivatives of functions, perform interpolations, *etc* are not included

- `potent` module:

Preliminary Work

HEADR	tdio.f	Outputs header
INPUTP	tdio.f	Reads input files
DATTIM	dattim.f	Inputs system date and time
READPT	tdio.f	Reads the run control data in the PCF
READGM	tdio.f	Reads gravity, body offsets, length and speed in the GDF
READBF	tdio.f	Reads basis flow characteristics in the BFF
CKPAR	tdpot.f	Checks the legality of the input parameters
NONDIM	tdio.f	Non-dimensionalizes the input
PANEL	geom.f	Evaluates geometric data for each panel
MCALC	geom.f	Evaluates m-terms for each panel
STATIX	geom.f	Evaluates the body hydrostatic coefficients
DEFMOD	defmod.f	Extends C, VEL, MTERM, and IMODE arrays for generalized modes
DEFINE	defmod.f	Defines the generalized mode shapes
WLSEG	tdio.f	Select panels adjacent to waterline
CKBODY	tdio.f	Checks for problems in the geometric discretization
PREMOD	tdio.f	Preliminary notation correction for the MODE routine
MODE	tdio.f	Sets up mode-symmetry indices
MCALC	tdio.f	Sets up the m-terms on each panel
CLIPHI	tdpot.f	Sets up the call to compute the incident-wave potential
IPHI	iphi.f	Evaluates the incident-wave potential
OPENBN	vtgrn.f	Opens binary file needed for finite depth calculations

Potential Formulation

GCALC	tdpot.f	Sets up the call to compute the free-surface Green function
VTGRN	vtgrn.f	Evaluates the free-surface Green function
GWCALC	tdpot.f	Sets up the call to compute the free-surface Green function for waterline panels
VTGRN	vtgrn.f	Evaluates the free-surface Green function
TZERO	tdpot.f	Sets up the t=0 problems
PREPAN	tdpot.f	Sets up the call for the Rankine Green function
RPN	rpan.f	Evaluates the Rankine Green function
FACTOR	tdpot.f	Sets up left-hand sides for factoring
SGETRF		LAPACK routine to factor matrices
SOLVPT	tdpot.f	Sets up vectors for backsubstitution
SGETRS		LAPACK routine to solve a factored matrix eq.
RHS0	tdpot.f	Calculates the $\psi^{(0)}$ RHS at t=0

		-either-
TSTEPS	tdpot.f	Computes the RHS's using the stored Green function
SRFCE	tdpot.f	Performs the integration over the body surface
WTRLNE	tdpot.f	Performs the integration around the body waterline
NONCON	tdpot.f	Calculates the non-convolution terms
		-or-
TSTEPR	tdpot.f	Computes the RHS's re-computing the Green function
SRFCE	tdpot.f	Performs the integration over the body surface
WTRLNE	tdpot.f	Performs the integration around the body waterline
NONCON	tdpot.f	Calculates the non-convolution terms
		—
SAVPOT	tdpot.f	Writes potentials to TIMIT.P2R every ten time steps
DATTIM	dattim.f	Gets a time-stamp for completion of the problem to this point

Source Formulation

GCLCSR	tdpot.f	Sets up the call to compute the free-surface Green function
VTGRN	vtgrn.f	Evaluates the free-surface Green function
GWCLCS	tdpot.f	Sets up the call to compute the free-surface Green function for waterline panels
VTGRN	vtgrn.f	Evaluates the free-surface Green function
TZERO	tdpot.f	Sets up the T=0 problems
PREPAN	tdpot.f	Transforms coordinates and coefficients
RPAN	rpan.f	Evaluates 1/R influence coefficients
FACTOR	tdpot.f	Sets up left-hand sides for factoring
SGETRF		LAPACK routine to factor matrices
SOLVPT	tdpot.f	Sets up vectors for backsubstitution
SGETRS		LAPACK routine to solve a factored matrix eq.
		-either-
TSTPSS	tdpot.f	Computes the RHS's using the stored Green function
SSRFCE	tdpot.f	Performs the integration over the body surface
WTRLNS	tdpot.f	Performs the integration around the body waterline
NONCON	tdpot.f	Calculates the non-convolution terms
		-or-
TSTPRS	tdpot.f	Computes the RHS's re-computing the Green function
SSRFCE	tdpot.f	Performs the integration over the body surface
WTRLNS	tdpot.f	Performs the integration around the body waterline
NONCON	tdpot.f	Calculates the non-convolution terms
		—
SAVPOT	tdpot.f	Writes strengths to TIMIT.P2R every ten time steps
DATTIM	dattim.f	Gets a time-stamp for completion of the problem to this point

Potentials from Source Strengths

		-either-
SC2PTS	tdpot.f	Computes the potential using the stored Green function
		-or-

SC2PTR	tdpot.f	Computes potential re-computing the Green function —
DATTIM	dattim.f	Gets a time-stamp for completion of the problem to this point

Gradients of the Potentials from Source Strengths

GCLCSR	tdpot.f	Sets up the call to compute the free-surface Green function
GWCLCS	tdpot.f	Sets up the call to compute the free-surface Green function for waterline panels
VTGRN	vtgrn	Evaluates the free-surface Green function
		-either-
SC2VLS	tdpot.f	Computes velocities using the stored Green function
		-or-
SC2VLR	tdpot.f	Computes velocities re-computing the Green function —
SAVPOT	tdpot.f	Complete output to TIMIT.P2R
DATTIM	dattim.f	Gets a time-stamp for completion of the problem to this point

- `respon` module:

Preliminary Work

HEADR	tdio.f	Outputs header
INPUTR	tdio.f	Reads input files
DATTIM	dattim.f	Gets a time-stamp
READRS	tdio.f	Reads the run control data in the RCF
READMD	tdio.f	Reads the mass, damping and stiffness data in the BPF
READIW	tdio.f	Reads the incident wave elevation from the IWF
READHD	tdres.f	Reads the control parameters from the .P2R file
CHECKR	tdres.f	Checks the legality of the input parameters
READPH	tdio.f	Reads the potentials/velocities from the .P2R file
NONDIM	tdio.f	Non-dimensionalizes input
XGFBOD	geom.f	Evaluates extended body attributes for nonlinear simulation
FREEBD	tdres.f	Builds the body mass and hydrostatic matrices
PREINC	tdres.f	Sets up the call to evaluate the incident potential
IPHI	iphi.f	Evaluates the incident potential
OPPNL	tdres.f	Writes supplementary panel files

Transient Quantities

FORCES	tdres.f	Computes steady forces, and impulse-response functions
SIMULA	tdres.f	Integrates the equations of motion (Runge-Kutta 4th)
TAIL	tdres.f	Extends the impulse-response function with asymptotics
SGELS		LAPACK least squares fit
SGETRF		LAPACK matrix factorization
SGETRI		LAPACK matrix inversion
GETF	tdres.f	Evaluates the equations of motion
LOCAL	tdres.f	Computes time histories of local quantities from simulation

Time-Harmonic Quantities

OPFRQ	tdres.f	Writes a file of equivalent frequency, period, wave-number <i>etc</i>
THFRCS	tdres.f	Computes frequency-domain coefficients and RAO's
T2FVPB	mdfp.f	Sets up transforms of velocity and pressure on body
; T2F*mdfp.f		Takes Fourier transforms of velocity or pressure on body
T14FIT	tdres.f	Computes coefficients of large-time asymptotics
SICI	tdres.f	Computes the sine and cosine integrals
MDFP	mdfp.f	Computes mean second-order force and moment
OPBPRS	tdres.f	Writes local pressure
OPVLS	tdres.f	Writes local velocity
OPDRFT	tdres.f	Writes mean second-order force and moment