

# Appendix C

## Using the WAMIT-RGKernel Interface

J. S. Letcher, Jr.  
AeroHydro, Inc.  
Southwest Harbor, Maine 04679 USA  
207-244-4100

Sept. 8, 2002; revised June 21, 2006; revised Sept. 20, 2008; revised Oct. 20, 2011

### Contents

Using the WAMIT-RGKernel Interface .....	1
Contents .....	1
1. Introduction.....	1
1.1 New features for WAMIT version 6.4.....	2
2. Supported features and options .....	3
3. Required files, versions and file locations.....	4
4. MultiSurf modeling considerations .....	4
4.1 Trimmed surfaces are excluded .....	4
4.2 Optional wetted-surface Entity List.....	5
4.3 Order of patches .....	6
4.4 Surface normal orientations.....	6
4.5 Base plane and waterline .....	7
4.6 Mirror symmetry .....	8
4.7 Rotational symmetry.....	8
4.8 “Fast” vs. “Accurate” evaluation .....	9
4.9 Divisions and Subdivisions.....	9
4.10 Irregular frequency removal .....	10
4.11 Coordinate singularities .....	10
4.12 Breaklines in surfaces.....	11
4.13 “Cosine spacing” .....	11
4.14 Parameters .....	12
4.15 Patch types and color coding.....	12
4.16 Internal tanks .....	13
4.17 Control surfaces for mean drift forces .....	13
5. GDF file format .....	14
6. Short GDF format for low-order analysis .....	16
7. Log file RGKLOG.TXT.....	16
8. Synopsis of operation.....	17
9. Error Conditions .....	18

### 1. Introduction

A joint development effort between AeroHydro, Inc. and WAMIT, Inc. has forged an intimate connection between MultiSurf and WAMIT. As a result, model geometry

developed in MultiSurf's "relational geometry" (RG) framework can be directly accessed by WAMIT for all setup and analysis purposes within WAMIT's "higher-order" solution method. This feature is available in WAMIT versions 6.1 and higher.

RGKernel is the C library of mathematical, geometric and data storage functions that supports MultiSurf, SurfaceWorks and other RG applications. It is compiled as a DLL. In this project a second DLL was developed to provide the interface between WAMIT's Fortran code and RGKernel's C procedures.

The advantages of this integration are:

- (1) **Accuracy.** RG geometry definitions are fundamentally exact, and in practice can be evaluated to a high degree of precision. This is in contrast to faceted and NURBS-based geometric representations, whose accuracy is commonly compromised by approximations. Accuracy of surface-surface junctions is critical in WAMIT's high-order analysis method.
- (2) **Convenience.** When MultiSurf is used to develop WAMIT models for this interface, the user avoids all further effort to develop panel files for the low-order method, or B-spline approximation files as have been needed for the high-order method. The Geometric Data File (GDF) is usually 8 lines for each body defined by this method.

The purpose of this document is to provide instructions for use of this new WAMIT feature. The topics include:

- Supported features and options
- Required files, versions and locations
- MultiSurf modeling considerations
- GDF file format
- Log file RGKLOG.TXT
- Error conditions

## 1.1 New features for WAMIT version 6.4

New features introduced to the RG2WAMIT interface for WAMIT version 6.4 include:

- Dipole patches (Section 4.15)
- Internal tanks (Section 4.16)
- Control surfaces for mean drift forces (Section 4.17)
- Transparent treatment of breaklines in MultiSurf surfaces (Section 4.12)
- Allowing NPATCH = 0 in the short GDF file (Section 5)
- Short GDF format for low-order analysis (Section 6)

## 1.2 New Features for WAMIT version 7.0

New features introduced to the RG2WAMIT interface for WAMIT version 7.0 include:

- Free-surface Pressure patches (Section 4.15)
- Internal tanks (Section 4.16)

## 2. Supported features and options

The WAMIT-RGKernel interface supports all current features of the higher-order solution option.

**Multiple bodies.** Each body is associated with a GDF file in the POT or FNAMES file. IGDEF = 2 is reserved to mean “geometry specified by an .MS2 file.” The different bodies can have various IGDEF types.

**XBODY positioning.** Each body has an associated XBODY transformation to position it in the global coordinate system. The MultiSurf model of a body can be constructed in any convenient position, e.g. aligned and centered in the body coordinate system. Alternatively, one MultiSurf model can be constructed with two or more bodies in their final global positions; this allows visualization of the bodies’ relative and absolute positioning.

**Mirror symmetry.** If the body has mirror symmetry with respect to one vertical plane, only one half of the body needs to be constructed. If the body has mirror symmetry with respect to two orthogonal vertical planes, only one quarter needs to be constructed. These symmetry options are represented similarly in MultiSurf and WAMIT.

**Rotational symmetry.** Alternatively, rotational symmetry is supported. A MultiSurf model can have rotational symmetry about any one of the coordinate axes, with any number of copies (NCOPIES), at an angular spacing of  $360/\text{NCOPIES}$  degrees. This feature greatly simplifies modeling structures that have rotational symmetry; only  $1/\text{NCOPIES}$  of the total model needs to be explicitly constructed.

**Units.** WAMIT is flexible with regard to units, but some of its inputs are quantities with units of length, and the global length unit is implicit in the GRAV parameter. MultiSurf supports meters, centimeters, millimeters, feet and inches as length units, and this choice is specified in the model file. The interface code compares the model and global units, and transparently performs any needed units conversions.

**Parameters.** The GDF file permits overriding floating-point values in the definition of any entity in the model. This allows a single model (parametrically constructed) to be

analyzed in a wide variety of configurations without opening the model and making any changes in MultiSurf.

**Variables and formulas.** Use of numeric variables and formulas in RG models is an advanced feature supported by MultiSurf versions 4.8 and higher. This represents a further powerful capability for constructing parametric model families. A variable can serve as a parameter in the GDF file.

**Irregular frequency removal.** Interior free-surface patches can be constructed in MultiSurf, and included in the data available to WAMIT for use with its  $IRR = 1$  option.

### 3. Required files, versions and file locations

RG2WAMIT.DLL contains the Fortran-C interface code.

RGKERNEL.DLL is the relational geometry C library. RGKERNEL.DLL is a component of any MultiSurf or SurfaceWorks installation, however please NOTE: **The WAMIT-RGKernel interface may use a different RGKernel version from MultiSurf.** These DLL versions are not necessarily interchangeable, and should be kept in separate directories. Although they have the same filename, different DLL versions can be distinguished by different sizes and different dates.

(This said, it has been our uniform experience to date that a newer version of RGKERNEL.DLL can be safely substituted for an older one; RG2WAMIT.DLL does NOT have to be recompiled in this event, and new modeling features incorporated in the newer RGKERNEL.DLL become available for use with WAMIT.)

RG2WAMIT.DLL and RGKERNEL.DLL must be located in the same folder as WAMIT.EXE. Also, any model file named in an  $IGDEF=2$  GDF must be located in the same folder as the GDF. We'll subsequently refer to this folder as the "working folder".

MultiSurf does not need to be running when the WAMIT run takes place. WAMIT does not need to be running when the model is constructed in MultiSurf. Communication between MultiSurf and WAMIT is solely through the .MS2 model file created in MultiSurf and accessed by RG2WAMIT.DLL.

### 4. MultiSurf modeling considerations

#### 4.1 Trimmed surfaces are excluded

Trimmed surfaces (TrimSurf and TrimSurf2 entities) can only be handled for low-order analysis ( $ILOWHI = 0$ ) by the RG2WAMIT facility. This is because WAMIT needs a complete parametric square over which to distribute its tensor-product description of the unknown potential, while a trimmed surface is by definition an arbitrary bounded *portion* of a parametric surface. When  $ILOWHI = 1$ , RG2WAMIT checks for the presence

of Trimmed Surfaces in the wetted-surface Entity List, and returns an error when one is encountered.

SubSurfaces, on the other hand, are perfectly legal for RG2WAMIT, because they have (like all other RG surface types except Trimmed Surface) 4-sided topology and a full  $[0,1] \times [0,1]$  parameter space. One or more SubSurfaces can always cover a complex-shaped region of a base surface, where a single Trimmed Surface might otherwise be used.

## 4.2 Optional wetted-surface Entity List

In order to be read and utilized through the WAMIT-RGKernel interface, an MS2 file (“model file”) may include an Entity List naming the wetted surfaces. When this option is used, WAMIT needs to be given the name of this Entity List (in the GDF file, see below).

Note: In MultiSurf 5 and earlier, an Entity List was called an "ObjectList". This is consistent with the following general terminology conventions:

MultiSurf 4 & 5		Surface Works and MultiSurf 6
"object"	=	"entity"
"entity"	=	"entity type"
"support"	=	"parent"
"dependent"	=	"child"

A simple alternative is to specify “\*” for the Entity List. This will signify “all visible surfaces in the model”. To be visible, a surface has to have a positive visibility index, and be on an enabled layer. Surfaces that are not to be included in the hydrodynamic analysis can either be hidden (by editing their visibility property), or located on a layer that is disabled in the MS2 file (unchecked in Settings/Layers dialog).

Entity List (ObjectList) is an entity type that has been supported since MultiSurf 4.0. It is just a list of other entities, of any type. It is a non-graphical entity; you may or may not be able to see the entities in the list -- that depends on whether each entity is made from a graphical entity type, whether it is visible, and whether it is on a turned-on layer. But, the Entity List itself is never drawn -- it is just a list.

In MultiSurf 6 and later, to create an Entity List, use Insert/ Entity List. You can preselect the “parents”, i.e. the entities to be in the list. You can add or subtract entities just as you would edit the multiple parents of any other entity. Select/ By Name is often the best way to select an Entity List -- or find it in the Entities Manager, under Entity Lists.

This list must contain each wetted surface, and only the wetted surfaces, in your model (or in the half, quarter or sector you are modeling explicitly when using symmetry image options). If you leave a wetted surface out of the list, include a surface twice, or include a construction surface that should not be included in the hydrodynamic solution, you will very likely find volume discrepancies in WAMIT (different volumes

for the X, Y and Z directions). The interface does not currently produce a warning in these situations.

You can have two or more Entity Lists in your model. They (like all other entities) will be distinguished by their unique names. By changing the Entity List name specified in the GDF file, you can conveniently switch from one Entity List to another without touching the model in MultiSurf. For example, you may have two sets of wetted surfaces representing two different drafts. Some surfaces can be common between these lists (e.g., surfaces that are fully wetted at the lightest draft), or the lists might be completely disjoint. (Note that in this scenario, you will need to also change XBODY(3) when the draft changes.)

### 4.3 Order of patches

The ordering of patches for the WAMIT solution will often be immaterial, but in some situations it matters. For example, if you wish to use the NPATCH parameter in the GDF file to exclude one or more patches (such as a free surface patch, in order to compare results with and without irregular frequency removal), then you have to be sure that the patches to be excluded are the last in patch order.

There are two rules governing the order of patches, corresponding to the two alternatives for the Entity List:

- (1) If you use an explicit Entity List, the patch order will be the order in which the surfaces are listed in the Entity List.
- (2) If you use the default identifier "\*" for the Entity List, the patch order will be the order of surfaces in the MS2 file. You can find out this order by searching for "Surf" in Edit/ Model File or Notepad. In MultiSurf another alternative is to turn on only the Surfaces filter; set Select/ Visible to "on"; Select/ All; and read the order in the Selection Set Manager.

### 4.4 Surface normal orientations

For a proper solution of the potential boundary-value problem, WAMIT requires that the positive normal to each surface be directed *into* the body. A patch closing the top of the body for irregular frequency removal must have its normal directed downward (again, into the body). The positive normal *in WAMIT* is calculated as the cross product  $\partial\mathbf{x}/\partial u \times \partial\mathbf{x}/\partial v$ , normalized to unit length.

If normals are backwards on some surfaces, WAMIT's calculation of volumes will very likely show discrepancies between the X, Y and Z directions. If all normals are reversed, the 3 volumes will agree but will be negative. (And the hydrodynamic solution probably won't make much sense.)

When making a surface in MultiSurf, the normal orientation is determined by:

- (1) the orientations of parent entities;
- (2) the order in which the parents are used, and
- (3) *the value of the "orientation" flag.*

The orientation flag is handy, because if a surface comes out with its normal opposite to what you want, you don't have to rebuild the surface another way, you just have to flip its "orientation" switch.

In MultiSurf, we visualize the normal orientation by drawing an arrow at the parametric center of the surface ( $u=0.5$ ,  $v=0.5$ ), in the positive normal direction. This arrow is drawn whenever a surface is selected. To see normal arrows simultaneously on all wetted surfaces, you first need to have an option turned on: this is under Tools/ Options, on the General tab, Auto Orientation needs to be "All selected objects". Then select all the wetted surfaces; for example, select the Entity List, then Select/ Expand Entity Lists/ First Generation. (The orientation of non-wetted construction surfaces doesn't matter, as these are not included in the surface definitions accessed by WAMIT.)

It is generally easier to see normals that point out of the geometry than those pointing inward. For this reason, our convention when making panel models for low-order WAMIT, and B-spline surface models for HIPAN and high-order WAMIT, has always been to require outward normals on all wetted surfaces, and then to reverse all normals automatically in the file conversion utility.

In the WAMIT-RGKernel interface, we have made the global reversal of normals optional. The choice is signaled (on a per-body basis) by the inward-normal flag in the GDF file (see below). This results in two options for the MultiSurf user who is building a model for the WAMIT-RGKernel interface:

- (1) Build the model with all unit normals pointing inward (WAMIT's convention), and set the inward-normal flag to 1; or
- (2) Build the model with all unit normals pointing outward, and set the inward-normal flag to 0.

We prefer option (2), because the outward normals are easier to see. In either case, the orientation flags of the individual surfaces can be freely used in achieving the desired consistent orientations.

#### **4.5 Base plane and waterline**

In MultiSurf, there are no restrictions on the location and orientation of geometry, but there are some sensible choices that will make the process easier.

**Vertical position:** Most models for WAMIT analysis are built in one of three vertical positions. Either

- (1) The model is built above a base plane, so the lowest Z coordinates are zero, and the design waterline is at some positive Z (call it  $Z_{wl} > 0$ ); or
- (2) The model is built with  $Z=0$  as the design waterline, so the lowest parts of the wetted surface are at a negative Z (call it  $Z_{draft} < 0$ ); or
- (3) The model is built with the point representing its center of gravity (C.G.) at the origin.

Any of these approaches works fine. In the first case, you have to specify XBODY(3) = -Zwl to correctly position the body in WAMIT; in the second case, XBODY(3) will be 0; in the third case, XBODY(3) will be the height of the C.G. above or below the waterline.

The WAMIT-RGKernel interface tests the “wetted surfaces” (i.e., the surfaces in the named Entity List, or when \* is used for the Entity List, the visible surfaces) during its initialization, and reports (in RGKLOG.TXT) any case where part of a “wetted surface” extends above the global waterplane. This usually indicates an error, either in the surfaces specified as wetted, or in XBODY(3).

#### **4.6 Mirror symmetry**

If the body has one or two planes of complete mirror symmetry, it is very advantageous to model only the independent half or quarter, and use mirror symmetry options to get the rest.

The WAMIT-RGKernel interface checks the model’s symmetry flags against the ISX, ISY flags specified in the GDF file, and issues a warning (in RGKLOG.TXT) if there is any discrepancy. Sometimes this can be harmless, but you should check your model out to be sure you haven’t either left surfaces out, or covered some areas twice.

#### **4.7 Rotational symmetry**

An object has N-fold rotational symmetry with respect to a particular axis if it is congruent to itself after a rotation of  $360 / N$  degrees about that axis.

Many offshore structures have complete rotational symmetry about a vertical axis. For example, three-leg TLP’s may have 3-fold rotational symmetry; spars and buoys often have rotational symmetry of some order; the Hibernia bottom-mounted platform is a 16-pointed symmetric star. (Four-leg TLP’s also usually have 4-fold rotational symmetry, but this case is best treated as combined X and Y mirror symmetry, because of the large efficiencies that result from exploiting mirror symmetry during the WAMIT solution.)

In MultiSurf, when the object being modeled has complete rotational symmetry, it is very beneficial to take advantage of it, as only a fraction ( $1/N$ ) of the total structure then needs to be explicitly modeled. Rotational symmetry is a model-level property. Z-axis rotational symmetry is only allowed when X- and Y-mirror symmetry flags are OFF. In the Settings/Model dialog, check X-, Y- and Z-mirror symmetry OFF, Z-axis rotational symmetry ON, and enter the number of copies, e.g. 3 for a 3-leg TLP. Then you can model only the “active” or “independent” sector of the model (e.g. 1 leg and 2 half-pontoons of a 3-leg TLP), and the copies will be present implicitly. The symmetry images will be shown automatically in Render view, and can be toggled on and off in Wireframe view with the <F5> key.

In the GDF file for a body using rotational symmetry about the Z-axis, ISX and ISY will be 0, and NPATCH needs to include the symmetry images. For example, if a 3-leg TLP is



modeled with 11 wetted surfaces in the explicitly modeled 120-degree sector, NPATCH will be  $3 \times 11 = 33$ .

Rotational symmetry about the X- or Y-axis is also supported – but only one axis at a time. (In WAMIT, this would have to be a completely submerged body, positioned vertically with XBODY(3).)

Rotational symmetry that is incomplete precludes the use of model-level rotational symmetry. In this situation, the entire geometry has to be built explicitly. But note, the RotatSurf and CopySurf entity type often provide a highly efficient way to construct the portions that are rotationally symmetric.

#### **4.8 “Fast” vs. “Accurate” evaluation**

RGKernel has two evaluation modes, denoted “Fast” and “Accurate”. In “Accurate” evaluation, all curves and surfaces are evaluated recursively, using the actual math functions that provide their definitions. In “Fast” evaluation, curves and surfaces are evaluated approximately by interpolation in a stored tabulation of the curve or surface; the fineness of this tabulation is controlled by the divisions x subdivisions products.

Fast vs. accurate evaluation is specified in the GDF file for each body (see below).

Accurate evaluation is typically slower -- in some cases, orders of magnitude slower; this especially occurs when there are deep levels of dependency in the model, and/or the model involves entities requiring iterative solutions, such as ProjSnakes and IntSnakes. The divisions and subdivisions should have no effect on the accuracy under “accurate” evaluation, since the table is not involved in the calculations.

Fast evaluation is usually much faster, and always more predictable in terms of performance, but involves some compromise of accuracy due to interpolation errors. Our general experience is that with the default 8x4 divisions x subdivisions, most curves and surfaces evaluate to near single-precision accuracy under fast evaluation. The accuracy should improve as the fourth power of the divisions, provided the divisions on all supporting curves and surfaces are increased in constant proportion. The “divisions multiplier” (Tools/ Settings/ Performance tab) is a simple way to make this increase uniformly across the model.

The GDF file format provides for overriding the model’s divisions multiplier (see below).

Pending further investigation of the accuracy of the integrated system, our current recommendation is to use fast evaluation mode, with default divisions and a divisions multiplier of 1. This should have little impact on performance, and should provide WAMIT with full single-precision accuracy for the geometry.

#### **4.9 Divisions and Subdivisions**

In MultiSurf, each surface has “division” and “subdivision” properties that control how the surface is subdivided for tabulation and display. For low-order WAMIT panelizations, we use the divisions x subdivisions to determine the mesh density, and some care with divisions is often required to make neat, watertight junctions between the panels on adjacent surfaces.

Since the surface geometry is more directly accessed in the WAMIT-RGKernel interface, divisions and subdivisions are less important in general. In “Accurate” mode evaluation, they should have no bearing at all. However, in “Fast” mode, there are still accuracy benefits in coordinating divisions between adjacent surfaces: for example, use the same number of (divisions x subdivisions) on a surface and its supporting curves; where possible; use matching divisions on surfaces that share a common edge.

#### **4.10 Irregular frequency removal**

In WAMIT's higher-order solution, irregular frequency removal is effected by providing additional surface patches that cover any “interior” portions of the plane of the free surface (IRR = 1).

Such interior free surface patches can be part of the MultiSurf model. They must be oriented with their positive normal direction consistent with the wetted patches; i.e., if your model has all outward normals, an interior free surface patch must have its normal upward. Interior free surface patches must also be included in the Entity List of wetted surfaces, and in NPATCH. If you put them last in the Entity List, you can easily run WAMIT with and without irregular frequency removal just by setting NPATCH appropriately in the GDF file.

#### **4.11 Coordinate singularities**

A coordinate singularity is a place on a parametric surface where the cross product  $\partial\mathbf{x}/\partial u \times \partial\mathbf{x}/\partial v$  vanishes. This can occur because either  $\partial\mathbf{x}/\partial u$  or  $\partial\mathbf{x}/\partial v$  vanishes, or because they have the same direction. The most commonly encountered coordinate singularity is a “pole”, where one or the other of  $\partial\mathbf{x}/\partial u$ ,  $\partial\mathbf{x}/\partial v$  vanishes along one whole edge of the parameter space, e.g. the type of singularity at the north and south poles of a sphere that is parameterized by latitude and longitude. There are many ways to create surfaces with poles and other coordinate singularities in MultiSurf; for example when a RuledSurf is made between a curve and a point, a pole occurs at the point.

Some kinds of coordinate singularities are associated with numerical ill-conditioning in WAMIT, and so should be avoided as much as possible in modeling for RG2WAMIT. There are situations where coordinate singularities are apparently harmless, e.g. the south pole of a half-submerged sphere, or the “south pole” of a truncated vertical cylinder. There are other situations where a coordinate singularity is distinctly beneficial. (See the remarks below under “Cosine spacing”.)

WAMIT, Inc. is recommending especially against poles on the free surface or at “chines”, i.e. sharp edges that protrude into the water (such as the junction between side and bottom surfaces of a truncated vertical cylinder).

#### **4.12 Breaklines in surfaces**

The RGKernel versions used in RG2WAMIT and MultiSurf recognize the presence of “breakpoints” in curves and “breaklines” in surfaces. A breakpoint is a place where a curve has a discontinuous derivative; if this is the first derivative (a “degree-1 breakpoint”) the curve typically has a knuckle. A “breakline” is a u- or v-constant line across which the surface has a discontinuous derivative; if this is the first derivative (a “degree-1 breakline”), the surface typically has a knuckle line at this parameter value.

Thus one reasonable way to build a truncated vertical cylinder is to construct a meridian curve with a breakpoint at the “chine” (e.g., a type-1 BCurve or CCurve), and revolve it 90 degrees about a vertical axis, creating a single RevSurf covering both the bottom and sides. MultiSurf will be aware of the degree-1 breakline along the chine and will display the sharp edge correctly.

Before WAMIT version 6.4, we had to recommend in general that breaklines **not** be used on surfaces intended for WAMIT consumption. WAMIT was not be aware of the breakline and would have been modeling a smooth, continuous distribution of potential across the chine.

In the version of RG2WAMIT released with WAMIT version 6.4, breaklines are automatically taken into account. A MultiSurf surface with N u=constant degree-1 breaklines and M v=constant degree-1 breaklines is split along these breaklines into (N+1) x (M+1) “panes” and is presented to WAMIT as that many separate patches. This all takes place “under the hood”, with little need for the user to be aware of it. (The splitting at breaklines will be visible in the patch counts seen in RGKLOG.TXT.)

(In the truncated cylinder example, modeling the side and bottom surfaces as two separate surfaces (rather than one surface with a breakline) would have the advantage of making it easier to utilize cosine spacing, to better resolve the rapidly changing potential around the chine.)

#### **4.13 “Cosine spacing”**

The phrase “cosine spacing” refers to systematic refinement of the mesh in regions where gradients are high, to provide better resolution in these areas. Although we don’t have a mesh anymore in the usual sense of a panel file, the concept of cosine spacing is in fact still quite pertinent, and the techniques for achieving it are highly similar to those used for the low-order method.

Exterior corners (“chines”) are the typical places where cosine spacing improves solution accuracy. Also, accuracy benefits are experienced with cosine spacing near the

free surface. In MultiSurf there are two common techniques for achieving mesh concentration:

- (1) Relabeling of lines, curves, snakes and surfaces
- (2) Type-2 B-spline lofted surfaces (BLoftSurfs) with duplicate master curves at the ends; also (usually for flat rectangular patches) B-spline surfaces (BSurfs) with duplicate control points at corners and/or edges.

#### 4.14 Parameters

As described below, the GDF file can contain any number of “parameter lines”, which override specified floating-point data values in the model. The format of a parameter line is just:

*entity-name float-index value*

For example, if Point ‘draft\_pt’ is a point whose Z coordinate (its third floating-point data value) controls the draft of the model, the parameter line:

```
draft_pt 3 -35.5
```

sets the draft to 35.5.

Obviously, carelessly setting parameters can “break” the MultiSurf geometry. For example, the leg radius and pontoon radius for a TLP might be separate parameters. As long as the pontoon radius is smaller than the leg radius, the geometry works; but if pontoon radius is set larger than the leg radius, a projection will fail, and the pontoon surface can’t be evaluated. This failure will be reported in RGKLOG.TXT. If you try the same settings in MultiSurf, you can see what goes wrong.

Even when all surfaces evaluate geometrically, there can be modeling problems that affect the WAMIT solution. When the pontoon radius is somewhat smaller than the leg radius, it’s possible for the two pontoons arriving at the leg to intersect each other, which may or may not cause a recognizable error condition in WAMIT. It is thus a good idea to check all parameter combinations that you plan to use visually in MultiSurf, before starting the WAMIT solutions.

#### 4.15 Patch types and color coding

WAMIT recognizes several types of surface patches with different characteristics for analysis. In the RG2WAMIT interface these patch types are distinguished by color coding -- assigning specific colors to surfaces to designate different patch types, as follows:

- (1) Color 11 (bright cyan) is reserved for dipole patches.
- (2) Color 15 (bright white) is reserved for exterior free surface patches (used for FDF file, for the second-order option).
- (3) Color 4 (red) is reserved for free-surface pressure patches.

Conventional patches (body patches) and interior free surface patches can be any other color besides 4, 11 and 15.

Surface colors are selected in the Properties Manager. The Color Property dialog has a control allowing colors to be specified by color number (0-255).

#### **4.16 Internal tanks**

Internal tanks with liquid contents and a free surface can be represented in a model for WAMIT analysis; see section 10.7 of the WAMIT User Manual.

In MultiSurf, the tanks are modeled with surfaces, and are communicated to WAMIT through a specially composed Entity List, called the "Tank list". The name of the tank list is included in the short-form GDF file as a second token on line 7, following the wetted surface entity list.

The tank list will have one entity for each internal tank; this must be either (1) a surface entity (when a single surface is enough to define the tank), or (2) an entity list of surfaces for the tank (in the usual case where multiple surfaces are required to define the tank).

It is possible to analyze the tanks alone, without the enclosing body. Just insert an empty Entity List (say its name is 'empty') in your model, and use it for the wetted-surface Entity List on line 7 of the GDF file:

```
empty tank_list
```

Note that WAMIT requires the normals on tank surfaces to be opposite those for body surfaces, because the wetted side is the inside of the tank, as opposed to the outside of the body. Thus if your body surfaces have outward normals, the tank surface normals should be inward.

The density of the fluid in each tank is specified in the .CFG file (RHOTANK).

#### **4.17 Control surfaces for mean drift forces**

A new analysis option in WAMIT version 6.3 is evaluation of mean drift forces and moments by means of momentum flux through a control surface surrounding the body. The control surface is defined in a control surface file (.CSF extension), with two format options:

(ICDEF = 0) low-order, similar to a low-order GDF; or  
(ICDEF = 1) higher-order, similar to a higher-order GDF.

The control surface can be modeled in MultiSurf. For ICDEF = 0, it can consist of any type of surfaces, including Trimmed Surface, triangle meshes or Composite Surfaces. For ICDEF = 1, it can consist of any type of surface *except* Trimmed Surfaces and Composite Surfaces.

The control surface can be part of the same MultiSurf model as the body surfaces, or it can be a completely separate model file.

CSF export is now on the MultiSurf menu, under File/ Export3D/ WAMIT.

## 5. GDF file format

A GDF file represents geometry for a single body. WAMIT's GDF file has a new format option, with IGDEF = 2 specifying "geometry to be obtained from an MS2 model file".

The first four lines of the GDF are the same as in other optional GDF formats:

```
header
ULEN GRAV
ISX ISY
NPATCH IGDEF
```

Where:

header is an identifying character string, up to 72 chars.

ULEN is a characteristic length

GRAV is acceleration due to gravity

ISX, ISY are symmetry flags (0 or 1) for X and Y mirror symmetry

NPATCH = number of surfaces in this body (not counting X and Y mirror symmetry images, but counting Z-axis rotational symmetry images)

(Starting with WAMIT v. 6.4, NPATCH can be specified as 0; see discussion below.)

IGDEF = 2, for using geometry from an .MS2 file

In an IGDEF=2 GDF file, there are always 4 more lines:

Line 5: NLINES = number of lines to follow this line (3 plus number of parameter lines)

Line 6: Model filename, including extension .MS2 (no embedded spaces)

Line 7: Name of Entity List to be present in the named MS2 file, listing the wetted surfaces. (Alternative: use '\*' to signify "all visible surfaces" – and make sure only the wetted surfaces are visible in your model when the .MS2 file is saved.)

Optionally, line 7 can additionally have a second Entity List name, specifying internal tank surfaces.

Line 8: 3 integers:

Fast/accurate flag (0 = fast, 1 = accurate);

Divisions multiplier override (0 to 10). A 0 for DivMult will mean "use the value in the model file".

Inward-normal flag (0 = outward normals, 1 = inward normals)

Then there are (NLINES – 3) optional parameter lines, each with 3 items:

*entity-name float-index value*

*entity-name* is the name of an entity in the model that has one or more floating-point numbers in its properties. For example, a Point has 3 floats -- its X, Y and Z coordinates or offsets --; a Bead has just one, its t parameter or offset.

*float-index* is an integer from 1 to the number of floats the entity has, e.g. 1 to 3 for a Point; 1 to 1 for a Bead.

*value* is the floating-point value to substitute.

### **NPATCH = 0 option**

Starting with WAMIT version 6.4, NPATCH can be specified as 0 in an IGDEF = 2 short-form GDF. This means "figure out the number of patches from the Entity List (or '\*' found on Line 7)". This option is highly recommended because it avoids the need to count surfaces, and the potential error of leaving surfaces out by specifying the wrong number for NPATCH.

**Note re symmetry:** The actual number of patches produced internally in WAMIT by this GDF is:

$$(\text{number of wetted surfaces}) \times (\text{ISX}+1) \times (\text{ISY}+1) \times N$$

where N is the number of rotational symmetry images.

The interface code does not need to count the X- and Y-symmetry images in indexing the patches. All mirror-image and XBODY transformations, and mirror-image indexing, are performed inside WAMIT. However, NPATCH does need to count implicit rotational images produced by Z-axis rotational symmetry.

**Note re Fortran READS:** Fortran will read this GDF with (5+NLINES) READ statements, each reading a complete record. This means any of the records (except line 7) can have other text following the required entries, separated only by one or more spaces. This additional data has no effect in Fortran, and is likewise being ignored by C. Line 7 is an exception, because presence of a tank-list on this line is optional; if there is a second token on line 7, Fortran will try to interpret it as a tank-list name.

### **Example GDF:**

```
Tension-leg platform example
120. 32.2 ULEN, GRAV
1 1 ISX,ISY
0 2 NPATCH, IGDEF
3 NLINES
TLP4H.MS2
wetted_surfs
0 2 0 Fast; DivMult; outward normals
```

Explanation: The first line is an identifying message. ULEN is 120 and GRAV is 32.2. This value of GRAV implies that the length unit for the global WAMIT model is feet. (If this MS2 file uses meters for its units, conversion from meters to feet will be performed

in RGKINIT.) The model has mirror symmetry with respect to its X and Y planes (only one quadrant is explicitly modeled). NPATCH is specified as 0; the interface will count the surfaces in Entity List 'wetted\_surfs' to determine NPATCH. IGDEF is 2 to signify geometry from an MS2 file. 3 for N LINES indicates 3 lines to follow. The MS2 file is TLP4H.MS2. This model file must contain an Entity List named 'wetted\_surfs' which names the wetted surfaces. This example has no tank-list. The last line specifies Fast evaluation; a divisions multiplier of 2, overriding any divisions multiplier in the model file; and the model is constructed with outward normals. There are no parameter lines in this example.

## 6. Short GDF format for low-order analysis

Beginning with WAMIT version 6.4, the "short" GDF format used for IGDEF = 2 higher-order analysis can also be used for low-order analysis (LOWHI = 0). The format is almost exactly the same as detailed in the previous section for higher-order analysis. The only difference is that the first token on line 4 must be 0, in place of NPATCH (the number of patches). RG2WAMIT will calculate the number of panels from the model geometry (the sum of panel counts over the wetted-surface patches).

Advantages of this feature are:

- (1) No long-form low-order GDF file needs to be created.
- (2) Parameter lines can be used in the short GDF file to vary the geometry at WAMIT run time.
- (3) Trimmed surfaces, triangle meshes and composite surfaces can be freely used along with other surface types for low-order analysis.

Note that a MultiSurf model that is suitable for higher-order analysis may not be 100 percent suitable for low-order analysis. The primary issue is divisions and subdivisions, which control the subdivision of surfaces into panels. A surface with very low divisions x subdivisions can be plenty accurate for higher-order analysis, but would result in very few panels. This aspect should be investigated, and divisions increased if necessary. CountPanels command will likely be a useful tool in such an investigation.

## 7. Log file RGKLOG.TXT

During operation of the WAMIT-RGKernel interface, a log file is opened and written to in the working folder (the same folder that contains the GDF file). This file, named RGKLOG.TXT, contains an echo of XBODY and GDF data, a reflection of some information extracted from the MS2 model file(s), brief reports on some aspects of interface initialization and utilization, and possible warning and error messages.

RGKLOG.TXT also includes a summary tabulation of surfaces vs. global, body and patch indices, which will give the correct order of surfaces for constructing an SPL file, if desired.

Example:



```

RGKLOG.TXT -- log file for RG2WAMIT.DLL
=====
Starting procedure RGKINIT at 25-Nov-2001 13:59:49
NBODY: 1
-----
Data for body no. 1
XBODY: 0.000 0.000 0.000 0.000
Truncated cylinder example 1.0 m radius, 0.5 m draft
ULEN: 1.000 GRAV: 9.80660
Global length unit (deduced from GRAV) is m.
ISX: 1 ISY: 1
NPATCH: 2 IGDEF: 2
MS2 filename: TRCYL2.MS2
Entity List of wetted surfaces: wetted_surfs
Model will be evaluated in FAST mode.
Unit normals are outward.
Opened RG model file: TRCYL2.MS2
Body 1 patch 1 surface name: side_surf
Body 1 patch 2 surface name: bottom_surf
-----
index  body  igdef  patch  ior  surface name
      1    1     2     1     1  side_surf
      2    1     2     2     1  bottom_surf
-----
Exiting RGKINIT at 25-Nov-2001 13:59:49, err = 0
=====
There were 1 RGKEVAL calls.
Executing RGKDONE at 25-Nov-2001 13:59:49

```

## 8. Synopsis of operation

The user generally will not need to know more about the functioning of the RG2WAMIT interface than has been explained in the operating instructions above. However, a little more perspective may be useful in case error conditions are encountered or suspected.

The sequence of communication between the programs is as follows:

1. During WAMIT's initialization process, it opens and writes to an ASCII text file named RGKINIT.TXT. This file is located in the working folder. It is created for each WAMIT run, whether or not the run uses IGDEF = 2 GDF's. RGKINIT.TXT contains, for each body, XBODY(1:4) plus the first 4 lines of any GDF; in addition it contains the remainder of the GDF file whenever IGDEF = 2.
2. WAMIT calls the procedure RGKINIT in RG2WAMIT.DLL. This initializes the interface, by using information read from RGKINIT.TXT. All specified model files are opened and prepared for evaluation calls.
3. WAMIT makes as many calls as it needs, to the procedure RGKEVAL in RG2WAMIT.DLL. Each such call evaluates one 3-D point at a specified u,v parameter location on one of the wetted surfaces furnished by the RG model.

4. When finished with its RGKEVAL calls, WAMIT makes one call to the procedure RGKDONE in RG2WAMIT.DLL. This frees all memory that was allocated during the RGKINIT procedure, storing data about the open models.

## 9. Error Conditions

There are a number of error conditions that can occur during the initialization phase of RG2WAMIT operation. If one of these occurs, there will be a message at the end of RGKLOG.TXT, and the WAMIT run will abort. The contents of RGKLOG.TXT will often be helpful in identifying the problem.

(Certain errors are also trapped during the WAMIT equation setup phase, but this error handling is primarily for debugging purposes; in our opinion such errors should not occur once the initialization phase has completed successfully.)

### RGKINIT error codes

#### 0. No error

#### 1. Can't open RGKINIT.TXT.

This probably indicates that RG2WAMIT.DLL and/or RGKERNEL.DLL are in the wrong directories. They must both be installed in the same directory as WAMIT.EXE.

#### 2. Unexpected end-of-file in RGKINIT.TXT.

The initialization file did not contain all required information. RGKINIT.TXT is written by WAMIT, and should not be modified by the user. If the file is incomplete, there's likely a bug in WAMIT. If the file appears complete, there's likely a bug in RG2WAMIT.DLL.

#### 3. Can't open a specified model file.

A model file (.MS2 extension) named in an IGDEF=2 GDF file was not found. Check to see that the file exists in the same directory as the GDF file. Check to see if the file name was spelled correctly in the GDF. Blanks (spaces) are not allowed in filenames for RG2WAMIT. (NOTE: Filenames are not case sensitive in Windows, but *are case sensitive* in Linux.)

#### 4. The specified Entity List was not found in the model file.

An Entity List of wetted surfaces may be specified by name in the GDF file. Entity names in RG are case sensitive, and must use an underscore '\_' in place of a space. Check to see that the Entity List name is correctly spelled and correctly capitalized in the GDF. Open the MS2 file in either MultiSurf or Notepad and see that the Entity List is there, and you have given its correct name in the GDF. You can find the Entity List in the Entities Manager under Entity Lists, or use Select/ By Name.

#### 5. Wetted surface Entity List has an error

The RGKernel error code will be given. This is most likely to occur because one of the surfaces failed to evaluate, and the Entity List got an error 284 (support failed) as a result. If the .MS2 opens in MultiSurf without error or warnings, it is very likely to open and evaluate successfully in RG2WAMIT. However, the two RGKernel.dll versions will generally be different and may have some incompatibilities, so it is possible for this error to occur through no fault of the user. This situation should be investigated by AeroHydro.

**6. Failed to build surface table**

Open the MS2 file in MultiSurf and confirm that the surface in question evaluates without error. If the surface evaluates without error in MultiSurf, it is very unlikely to fail in RG2WAMIT. However, because of some incompatibilities between RGKernel.dll versions, this error might occur through no fault of the user. AeroHydro should investigate. (Note: a table is built for each surface in the model even when “Accurate” evaluation is specified.)

**7. Unexpected end-of-line in RGKINIT.TXT**

A file record in RGKINIT.TXT did not contain all required data. For example, if the file line giving fast/accurate, divisions multiplier, and inward-normal settings has only one or two tokens, this error will occur. Check that your GDF file contains all required information, and that NLINES is correct.

**8. Wetted-surface Entity List has fewer than NPATCH surfaces**

The GDF file specifies NPATCH, the number of surfaces to use in the solution for a given body; the Entity List does not contain the full complement. Check that NPATCH is correctly specified in the GDF. Open the MS2 file in either MultiSurf or Notepad and check that all wetted surfaces and/or interior free surface patches are in the Entity List. If the number of surfaces is large, making counting difficult, select the Entity List in MultiSurf (often Select/ By Name is the easiest way), and the number of its parents will be indicated in the Properties Manager. Visually, you can select the Entity List and Select/ Expand Entity Lists/ First Generation; all surfaces in the list will then be highlighted.

**9. Wetted-surface Entity List contains a non-surface entity.**

The wetted-surface Entity List must contain all the wetted patches and/or interior free-surface patches, and no other entities. (An Entity List, in general, can legally contain entities of any type.) Check the contents of the Entity List and see that they are correct in number and identity.

**10. Wetted-surface Entity List includes a Trimmed Surface.**

Trimmed surfaces are not eligible for use in WAMIT models for higher-order analysis, because they do not provide a complete square parameter space for distribution of the unknown potential. You may be able to replace a Trimmed Surface with one or more SubSurfaces.

**11. Surface entity is in error.**

The RGKernel error code will be given. This error code is compatible with the MultiSurf error system, so it can be looked up in the MultiSurf reference manual or help system under "Error codes". The same error should occur when the model is opened in MultiSurf; however, this will not always be true, because of the presence of incompatibilities between the two RGKernel.dll versions. See also remarks for error 6.

**12. The specified tank list was not found in the model.**

A tank list was specified by name on line 7 of the GDF file, but no entity with this name was found in the model. First, confirm that you are trying to perform analysis with internal tanks in this run; if not, line 7 should have only a single token (the wetted-surface Entity List name, or '\*'). Check that the second token on line 7 is the correct name for the tank list. Entity names are case sensitive. Check that an Entity List of this name exists in the model.

**13. The entity named as tank list is not an Entity List.**

A tank list was specified by name on line 7 of the GDF file, but the entity with that name is not an Entity List. Check that the second token on line 7 is the correct name for the tank list. Entity names are case sensitive. Check that an Entity List of this name exists in the model.

**14. The tank-list Entity List is in error.**

A tank list was specified by name on line 7 of the GDF file, but the entity with that name is in error. An error code will be given; this can be looked up in the MultiSurf manual or help system. The most likely cause is that one of the parent lists or surfaces is in error; this will result in error 284 on the tank list. In the Entities Manager, select the tank list and expand the Parents tab to identify the error.

**15. Tank list contents don't qualify for a tank list.**

A tank list was specified by name on line 7 of the GDF file, but the Entity List with that name does not qualify as a tank list. Each element of a tank list must be either (1) a surface, or (2) an Entity List of surfaces.

**16. The wetted-surface Entity List contains duplicate surfaces.**

Duplicate surfaces in the wetted-surface Entities List will lead to singular or very ill-conditioned equations in the WAMIT solution. Select the Entity List and click on "Entities" in the left-hand column to review its contents. Click in the right hand column to edit its contents; remove any duplicates.

**17. IBODY is out of range.**

In a multi-body problem, each IBODY must be in the range 1 to NBODY. This error would probably indicate a bug in WAMIT's writing of RGKINIT.TXT.

**18. Two or more bodies have the same IBODY.**

In a multi-body problem, the IBODY's must be distinct. This error would probably indicate a bug in WAMIT's writing of RGKINIT.TXT.

**19. A surface has a patch type that is invalid for analysis.**

Patch types are controlled by color coding; for example, color 11 (bright cyan) is reserved to indicate a dipole patch. Color 15 (bright white) is reserved for exterior free surface patches; these are excluded from both low-order and higher-order analysis. Review the colors of your surfaces; don't use color 15.

**20. Demo version, used with non-demo model file.**

Wamit, Inc. distributes two different versions of rg2wamit.dll, both with the same filename. The demo version is restricted to only function with a few specific model files, and is freely distributed. The real version is only distributed to users who are licensed to use WAMIT and MultiSurf together. This error occurs when the demo version of rg2wamit.dll is used with a MultiSurf model file other than one of the authorized demo files.

Using WAMIT-RGKernel Interface1.doc