Hullform for Windows

Statics Reporting Module

"HULLSTAT"

Blue Peter Marine Systems

Contents

Index 47

1. Overview

Hullstat is an optional component of the hull design program Hullform, which allows production of formatted reports to the standards required by regulatory authorities. It is designed to provide maximum flexibility, accommodating both the wide variation of regulatory formats, and individual preferences for the style of report.

1.1. How Hullstat Works

Hullstat is integrated within Hullform, and uses its input / output, graphical and user interface services, so that it occupies very little extra disk space. It communicates with Hullform using the information pathway provided by Hullform's DDE service, for data enquiries and to set Hullform's own internal parameters (like hull displacement).

Hullstat can, however, do much more than probe Hullform for answers to enquiries. It provides two important sophistications:

- Generation of reports based on the format of provided, or user-designed templates;
- A full interpretive programming language which allows complicated operations to be performed on Hullform's calculations, including graphical displays, and to be repeated for different configurations of the design's operating conditions.

These mean that, in a single operation, Hullstat can produce an almost-complete stability report for any design Hullform can handle.

1.2. The Hullstat Report Model

Hullstat generates reports for a set of the design's intended operating conditions, defined by

- static loads, of any mass, and any longitudinal, lateral and vertical position,
- internal tanks, holding fluids of arbitrary density, and which may be partially or completely filled,
- immersion points (also known as downflooding points), being points on the hull where immersion below the waterline is an issue.

The program can repeat any combination of statics calculations, using Hullform's own statics capacities, and additional internal calculations performed using its own "Hullform statics language" (subsequently referred to as "HSL"). HSL is an interpretive language, comparable to Basic, but customised to simplify insertion of Hullform's statics calculations, and of parameters defined by Hullstat's set of operating conditions.

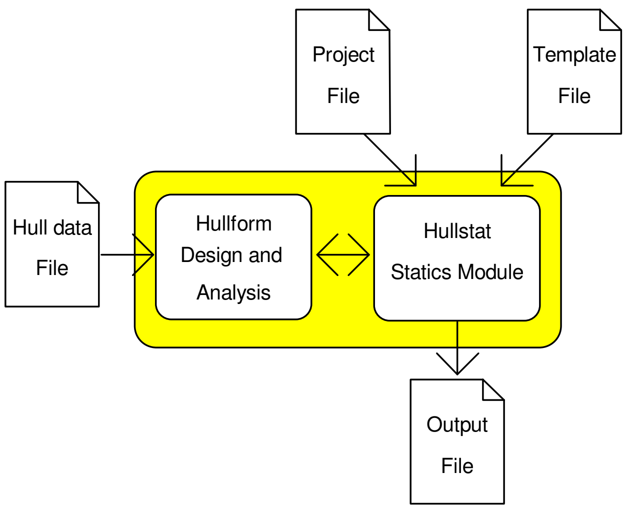
Program output may be in the form of a text file, or as a formatted "rich text format" ("RTF") file readable by word processors such as Wordpad and Word for Windows. In both cases, the HSL commands, and the formatting information required to generate the output file, are embedded in a file known as a template.

The template is also a text or RTF file, directly modifiable by the user. This ensures maximum flexibility, since it means that Hullstat can generate reports:

- in any language which can be represented in a text or RTF file,
- of almost any form (stability table, graphical cross curve, hydrostatic table, text summary) which can be put into a word processor document.

The intent is that a Hullstat report can be presented almost unedited, to a regulatory agency for approval.

In this manual, Hullstat is treated (and referred to) as a separate module within Hullform. This is to identify functions within the statics reporting process, and does not reflect any lack of appreciation that Hullstat is integrated within the enhanced version of Hullform which embodies both functions.



1.3. Quality Assurance Aspects

Quality assurance procedures are the ultimate responsibility of the program's user. However, the report generation procedures within Hullstat are designed so that a complete record is kept of all sources of information. This simplifies the record-keeping aspects of any quality assurance process.

There are two aspects to this, within Hullstat:

1. All output from the program is directed to a single file directory (“folder”). When Hullstat is run, not only the stability report, but also copies of the hull data file, the project file and the template used to generate the report are placed there. The output file directory retains a complete record of all files input and output in the report generation process.
2. The name, modification date and modification time of each file used - hull data file, project and template - are logged in the output file. In addition, any user inputs via dialog boxes are also written into the output file. This means that no inputs contributing to the report go unreported.

These two features mean that, as long as the one file directory is used to accumulate only the output for one project, the key items to reproduce the program's output at any arbitrary time later will be retained.

However, there are other aspects of which users must be aware. It is recommended that other precautions be followed:

3. The Hullstat output directory may receive the output generated using other templates for the same project - the other templates will also be copied, with no harm to previous output. But output from other projects should be written to other directories.
4. When finalising a report, to which other text might be added (e.g., summaries of regulations, descriptive text), the report may be created as a document within the output directory, but should not overwrite to Hullstat output files. Their contents should be included in the final report using the Windows' “Copy - Paste” approach, or a word processor's “insert file” function. It is safest to retain the Hullstat output files unaltered.

   This step has the added benefit that the quality assurance information in the Hullstat output files is retained, but can be omitted from the final presentation document.

If these approaches are taken, then after a stability project is completed and the final report is presented, a diskette backup of the output directory will retain all the information required to reproduce the whole output set.

1.4. A Note About Manuals in General, and this Manual in Particular

This manual attempts to cover in detail a software tool which is in many ways different from those you have used before, and is in other ways very similar.

As for many sophisticated systems, you can treat it as a “black box”, using only its operations at “user interface” level. You can miss section 5 of the manual, dealing with the mechanics of Hullform’s templates. Hullstat will still work well for you, but you will not access the full flexibility available.

To access the power of Hullstat's templates, you will need to apply a deeper level of skills. This is where you might be tempted to question the adequacy of the manual. If you have used modern word processors for presentation of output, and made wide use of a programming language like basic, you may find its contents boringly repetitive. On the other hand, if you lack experience in both these areas, you might find it too thin. This only proves that it is impossible to please all users.

If you are in the more expert group, a few experiments with the provided templates and good read of sections 3 and 4 might be enough to get you going. However, if you are in the latter group, please read this manual thoroughly. It is quite repetitive - sections 3 and 4 cover much the same material, and in section 5 some points are touched on up to three times. This is deliberate, because we all learn in different ways.

If you like a “from the bottom” explanation, start reviewing Hullstat templates from section 5. Whatever approach you take, you will need to refer back to this section, anyway - it contains all the definitions you will need. If you want to work through some examples, section 5.6 will be useful. If you like to see the examples explained, use section 5.7.

And remember, at all times, that product support is an in-built component of the price for Hullstat. If you have questions, do not be afraid to ask - see section 8.

2. Getting Started

2.1. A Test Run

In the set of sample projects provided with Hullstat is one entitled “Hullstat demonstration”. It uses a design with several internal tanks, with name “hst-demo.hud". This will allow you to run a quick test, to see how the system operates.

Firstly, load Hullform, and run Hullstat, either by

- selecting menu items “Statics”, “Report” (the old entry path, compatible with Hullform 7),
- selecting main menu item “Hullstat”, or
- pressing the Hullstat button on the toolbar. You will see a display like that on the following page.

To begin, press the “Open project” button, find and select file “hst-demo.hsp”.

Next, you will need to confirm the directories of the files needed, so press the “File list” button. Against each of the edit boxes labelled File name, Editor, Template and Output is a “browse” button. Check that each entry is correct by pressing each “browse” button in turn.

If you see the named file or directory each time, you need make no change, so just press the “Cancel” button on the file dialog box. If the required entry is not present, you will have to search for it, and select either

- the file required, for the File name and Editor entries, or
- a file in the required directory for the others. When you press the “Open” button, the required path will appear in the associated edit box.

You can, at this stage, also configure the program to use the editor and template format of your choice. The default is the Windows Notepad editor, which is satisfactory for text templates only. If you prefer to use an RTF-compatible word processor, such as Windows’ Wordpad editor, you should change the editor name to point to the correct executable file, and select the “rtf” template directory rather than the “txt” one.

You are now in a position to try a running a template, to generate some output. Press the “Run a template” button, and you should see a set of available templates. If you do not, it is possible that you are using the incorrect file type ("txt” rather than “rtf", or vice versa). You can change the type using the two radiobuttons below the file list boxes.

Now select one template from the list (The “deadweight and stability” one is a good one to try first). Initially, your output directory will be empty, so you will have to type a name in the output file edit box. You need only type in the name itself - the default extension (".txt” or “.rtf”) will be added if the file name does not include a “.”.

(Later, when the output directory includes files which can be overwritten, you can simply select the input template and output file using the mouse)

When you have nominated the required files, press the “Ok” button, and processing will proceed. You will probably see output in the trace window of the Hullstat dialog box, and also output in the Hullform window - showing clearly that Hullstat and Hullform are sharing the work.

When processing ends, you can press the “Edit output". You will see the end result of the job you have just completed.

2.2. A Quick Review of Input and Output

Based on commands and formatting information in the template you used, you have generated stability report. To see what has been done, you need to look at both the template and the output file. Press the “Edit template” button to load the template into your nominated editor, then load the output file into your editor too. Using software which can handle multiple documents, this second stage may be most simply achieved within your editor, but otherwise you can return to Hullstat using the Alt-Tab key combination, then press the “Edit output” button.

If you look first at the template, and if you have used the “deadweight and stability” template, you will see two header lines, the first containing text “`state` ”, the second text “Deadweight Table”. Looking in the output file, you will see that the corresponding lines are “Arrival With Partial Catch” and “Deadweight Table”.

Obviously, the second of these lines has been copied straight from the input file to the output file (nothing remarkable here). The first line, however, has been replaced with text, in this case using information from the Hullstat project file. If you use Alt-Tab to switch to Hullstat (leaving both edit files loaded), you can press the “Operating conditions” button, then the “Edit” button, and you will see this text in the second edit box. (Press “Quit” one this dialog box, and on its predecessor, to clear these dialog boxes and return to Hullstat’s main window)

This is a basic example of the way information from the template is merged from information held by Hullstat (or retrieved from Hullform by Hullstat). The back-quote symbol, `, is used within the template to request substitution. The text within the quotes was in this case a permanent Hullstat variable name, “state”, but it could have been a variable defined within the HSL commands in the template, or any arithmetic expression.

As you look further down the template, you will see further examples, some with a % symbol; this controls the number of decimal places included in the report. You will also see an arithmetic expression, “VCG+FSM/totdisp”, array references (e.g., “area[i30]”).

After the heading “GZ - Righting Lever Curve”, or earlier in the template, you may see lines commencing with the # symbol.

If you do not see any # symbols, ensure that you have set your editor or word processor to show hidden text. Under Word for Windows, the option is set using menu item “Tools”, “Options”, under the “View” dialog box tab.

These lines are HSL commands. If you think they make the template look like a computer program, you are right - in fact, there is very little difference between a document template used by many of today’s software offerings and a computer program. The main difference in the case of Hullstat is that the internal structure of the template is in text, rather than binary, format.

You don’t need to use HSL, or to modify the commands, at this stage - later you can if you want to (See Chapter 5). But just for now, notice the occurrence of words and word-like commands, such as “loop” and “foreach”. These force repetition of sections of the template, sometimes for a range of angles of heel, displacement, centre of gravity (etc), sometimes for a range of static loads or operating conditions. If is by using these that Hullstat can repeat calculations, to generate tables, graphical cross-curves and analyses for a full range of operating conditions for your design.

3. Components of a Project

In Hullstat, you commence by creating a “project”, which identifies the design, and the loading conditions which are to be analysed.

Project files have extension “HSP” (representing “Hull Stability Project”). Some sample projects have been provided with the distribution set, but you may need to modify one, or create a new one, to generate a report for a design of your own.

The project file, plus the Hullform hull data file and the Hullstat report template used, include all information required to regenerate a report. These files provides a basis for any quality assurance process, and should be retained together when a record of the report generation process is required (See section 1.3).

3.1. Vessel Name

The first entry in the project definition is the name of the design. This name is normally limited to 40 characters, but can be larger. If you want to use a longer name, press the “Preferences” button, and enter a new text data size (up a maximum of to 128 is allowed). The name could be simply the vessel name, or include a brief comment as well (e.g., “Ocean Chief (as revised 27 February 1999)”).

Most templates copy this name to a header line, when any output file is generated.

3.2. Hull Data File Name

Independently of the vessel name, the name of the Hullform hull data file to be used is also included (e.g., “Ocean Chief.hud”). The name may be up to 128 characters long. The “.hud” extension is not required (but may be included if preferred).

If a directory path specification is included (for example, “C:\HULLFORM\HULLDATA\Ocean Chief.HUD” or a shorter form such as “..\hulldata\96043\final”) this will be used as specified. Otherwise, the current default hull data directory will be used. Since the default directory can be changed, almost inadvertently, it is recommended that a path specification be included in the hull data file name.

The hull data file name is reported in the “run summary” which precedes to main output of a Hullstat report file, but is not available as a variable name for inclusion in the main report body. This is because the name is specific has no significance outside the Windows operating environment in which Hullform and Hullstat run.

3.3. Editor Name

Hullstat reports may be generated in bare text form, or as “rich text format” ("RTF”) files. Any editor can be used to manipulate text files, but a word processing program which accepts RTF files (such as Word for Windows or Wordpad) is required if the latter are used.

Unless the editor is accessible in the Windows file search path, you will need to include its full file path name, as well as its file name. If you locate the file using the “Browse” button, the full path name will be included.

3.4. Template Directory

A template is a file which defines the details and format of the output. You may generate a set of reports from one project, and it is a policy enforced by Hullstat's own output procedures (See section 1.2) to retain copies of the templates used in a single directory.

The Hullstat project includes specification of the directory (“folder”) from which templates are read. Files in this directory will normally have extensions “RTF” or “TXT” and, at least initially, will comprise sample templates provided with the program.

Each template includes three forms of content, namely regular text copied to the output file, fields which are replaced by numeric values or text, and processing commands. Templates are explained in detail in section 5.

3.5. Output Directory

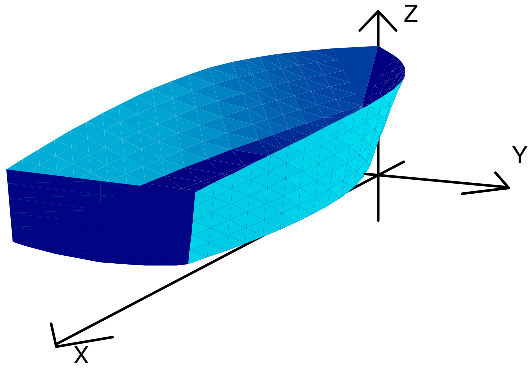
This where the files receiving the program's output are written. It also receives copies of the Hullform hull data file, of the Hullstat project file and of the templates used.

It is recommended that the output directory be specific to one design. The logic is related to quality assurance matters - see section 1.3 for details.

3.6. Static Load List

Static loads remain in the same position for all operating conditions, but may vary in mass. They have longitudinal ("X”), transverse ("Y”) and vertical ("Z”) coordinates, length, width and depth, as well as mass.

The direction convention used is X positive sternward, Y positive to starboard and Z positive upward. While Hullform allows Z to be positive upward or downward, Hullstat adopts the positive-upward convention which is standard for statics reports.



The dimensions of the static loads currently have only a little significance. They are used in the graphical editing facility of Hullstat, but there is no attempt within the program to prevent loads overlapping.

There is currently a limit of 20 static loads. This limit is, however, arbitrary, and will be increased if any user finds it inadequate.

3.7. Immersion Point List

An immersion point (also known as a downflooding point) is a location, on or near the hull surface, whose immersion may affect regulatory approval. Hatches and portholes which may be opened are examples. Immersion points are normally fixed points on the hull, so remain the same for all operating conditions. Each immersion point is specified by its longitudinal, transverse and vertical coordinate.

Not all templates use the immersion points. However, those investigating heeled stability normally test for their immersion, since stability criteria are often affected - e.g., maximum heeling lever at or before the first such point is immersed.

There is currently a limit of 20 immersion points. This limit is, however, arbitrary, and will be increased if any user finds it inadequate.

3.8. Tank List

While static loads and immersion points are specific to Hullstat, internal tanks are defined within Hullform. However, to allow full analysis, the contents (as a percentage), permeability and the specific gravity of each tank may be altered by Hullstat.

The number of tanks specified in the project file must match the number defined for the hull. If not, an error message is generated, and no output is produced.

There is a limit of 20 internal tanks.

3.9. Operating Conditions

An “operating condition“ is defined by the mass of each defined load, and by the percent contents and specific gravity of each defined internal tank. Typically, they correspond to the state of the hull at various stages of a voyage, or loaded and equipped for different tasks.

For each operating condition, the following terms are defined:

3.9.1. Brief Description of Operating Condition and State

These may be up to 128 characters long, but their size in the statics report is limited by the user-defined text data size (See section 7.4). They specify the nature of the operating condition - e.g., “Arrival, 10% Fuel, 20% Catch". Information may be split across the “condition” and “state” definitions, to allow more details to be entered into the output file.

3.9.2. Static Loads

The mass of each static load is a property of each operating condition. Loads are measured in whatever units are used for the design being analysed - kilograms, tonnes, pounds or tons.

3.9.3. Tank Contents

For each of the internal tanks, the percent contents, permeability and specific gravity are defined. Tank contents should range from 0 to 100, and specific gravity will normally be in the range 0.8 to just over 1. Permeability corresponds to the fraction of the tank volume which is available to the liquid

4. Formulating a Project

This section forms one way of introducing yourself to the use of Hullstat. It is a step-by-step explanation of the creation of a Hullstat report. You may also find section 5.7, "Sample Templates", useful.

Obviously, you need to start up Hullstat, within Hullform, and you should have read Chapter 3, detailing the meaning of the project elements. Then, press the "New Project" button. When aksed "Do you want to be guided through your new project?", press the "Yes" button. You will then be taken sequentially through the project initialisation phase.

4.1. Project File Definition

The first dialog box you see is the one you obtain when you press the "File list" button. You have several items to enter here:

Vessel name	This may be up to 128 characters. This is not the name of the hull data file, but the name which will probably be written on the bow / and or stern.
File name	The name of the Hullform hull data file to use. This must already exist, and you can use the "Browse" button on the right to ensure you have located it correctly.
Editor	The name of the program with which you will edit both the template and output file. It must be a text editor (if you are working with a text-format template) or a word processor able to read and write RTF files. Even if you have installed no such software, the standard Windows programs Notepad and Wordpad can be used. (Notepad will be in your Windows directory; Wordpad is probably in directory c:\Program Files\Accessories)
Template directory	The directory which holds the files which will define the format and content of your report. When you installed Hullform+Hullstat, one named "Template" was created in you Hullform directory. Look in this, and you will see two folders, "Txt" and "Rtf", corresponding to directories containing text and richt-text-format templates. Open one (corresponding to the file format you require), and double-click on one of the templates.
Output directory	The directory to which all output files will be written. it also receives copies of the hull data file, the project file and all templates used. A standard output directory was created at installation, off your Hullform directory. It is recommended that you create a separate directory for each project. You might like to start by using the "Browse" button for this entry to locate the Output folder, opening it, and creating a new folder here using the "New folder" button of the file dialog box.

Details of these terms were given in section 3. After entering these, click "Ok", and then the "Save as ..." button to ensure your work will be accessible later.

4.2. Adding, Deleting and Editing Static Loads

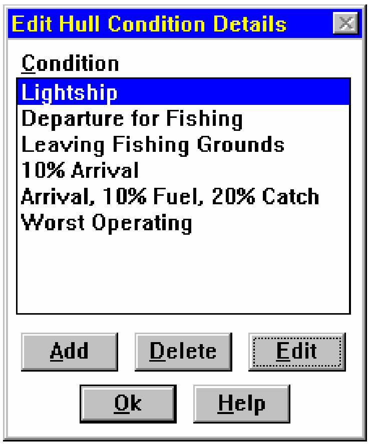
Your next step is to create the list of static loads. These are identified by a name, a three-dimensional position about the hull, and a mass which varies with the operating condition. (Note that Hullstat does not provide for movable loads - but if you need to move a load between operating conditions, all you need to do is to define two loads, each of zero mass in one condition, non-zero mass in the other)

You will initially see an empty list box, with buttons "Add", "Delete" and "Edit" at the right hand side. To add a new static load, press the "Add" button.

This generates a dialog box, allowing entry of descriptive text at the "Description" edit box, load position at the "X-position (LCG)", "Y-position" and "Z-position (VCG)" edit boxes, a default load mass at the "Mass" edit box, and a length, width and depth. The "description" may be any text string, up to the user-defined limit of text data (See section 7.4), while the position coordinates are specified in terms of the coordinates and measurement units used for the analysed hull (noting that positive upward, positive to starboard and positive sternward conventions are used).

The mass you enter is the value which is entered when you create a new hull operating condition. You can edit that value, allowing for varying static loads which vary between operating condition (e.g., freight in a hold).

The length, width and depth of the load only affect its display in the "Graphical edit" section of Hullstat.



Before commencing entry of the load position, it may be necessary to view carefully the hull design, to determine exactly the coordinates of the load, in the hull reference frame. A useful technique is to use the “Edit sections” window, since, when a particular section is selected, the lateral and vertical position of the cursor in terms of the hull coordinates is shown on-screen.

The mass entered here becomes the default for each operating condition, and is the first value seen in the “Mass” edit box when the added load is configured for a new operating condition.

When you have added a load, it will appear in the list box, and you may then add further loads using the “Add” button again.

To delete a load, you only need highlight it in the list box, and press the “Delete” button.

To edit a load, highlight it in the load list box, and press the “Edit” button. You will be presented the same dialog box as for the “Add” option, with the current defaults displayed for alteration.

When entry of all loads has been completed, press the “Quit” button.

4.3. Adding, Deleting and Editing Immersion Points

Immersion points are manipulated exactly as for static loads (See section 5.3), except for the absence of a mass and dimensions from their list of properties.

An immersion point is not always located on a part of the hull represented within Hullform - e.g., a raised vent. As a result, immersion points may be placed anywhere relative to the hull.

Immersion points are taken to correspond to fixed points on the hull, so do not vary between operating conditions. As a result, you can not edit them from within the “Operating Conditions” dialog box.

4.4. Create Your Operating Conditions

The next stage involves defining the design’s operating conditions. You will next see the “Operating conditions” dialog box (the one you would otherwise see when you pressed the “Operating conditions“ button). One of the defaults created by use of the “New” button is an initial operating condition, “Lightship". This is the condition of the hull with no added static loads, and all internal tanks empty.

You can edit this condition, but since “Lightship” (or another name for this state) is a common component of any statics report, you may want to leave it untouched. So, press the “Add” button in the dialog box. You will see another dialog box, with two edit boxes into which you may enter text.

The “Condition” edit box is intended to receive a brief statement of the operating condition of the hull - e.g., “Departure from Port". The “State” edit box is intended to receive a summary of the state of the hull - e.g., “Full fuel tanks, empty freezer room". How they are used is up to you, however, since their location in the report can be changed by alteration of the report template.

When you press the “Ok” button, you will be returned to the “Operating conditions” edit box. you will see your new condition highlighted in the list box provided.

You can now select the “Edit” button, and you will see another dialog box, with empty lists entitled “Static loads” and “Tanks". For each, there is an “Edit” button, allowing editing of listed loads and tanks. But there is no “Add” button for either.

This is because loads and tanks have existence independent of operating conditions. Loads normally represent items fixed within the hull (e.g., hold contents), but which may vary in mass between operating conditions. Tanks are fixed within the hull, and while they may contain different volumes, and occasionally even different fluids, they must be represented for each operating condition.

You will also see that there is no capacity for editing immersion points within the “Operating condition” dialog box. Immersion points also exist independently of the loading condition of the hull, are so are also defined independently.

4.5. Defining Tank Properties

All necessary characteristics of a tank are defined within Hullform, so that the Hullstat module only needs to adjust those which can vary between hull conditions - normally, the percent filling of a tank, and the density of its contents. Hullstat also provides for alteration of the tank's permeability, but normally there is no need to change this term

When a hull condition is edited, all current tank information (with the exception of percent full and specific gravity) is obtained from Hullform, so that editing is appropriate to the hull. These are accessed through the Edit, Operating conditions function of Hullstat (See section 5.7, below).

4.6. Saving the Project

Once you have created your list of operating conditions, you can save the details of the project in a file, so you can re-use it later. This step is the next offered in the "New project" sequence, and it is clearly a sound step to take. Use a project name which reflects the name of the design, so you can find it easily later.

4.8. The Graphical Edit Window

You can modify static load and immersion point locations very easily using the "Graphical edit" button. This opens a new window, in which you may display a "wire frame" view of the hull in profile, end or plan views. All loads and immersion points are shown, loads as blue rectangles, and immersion points as small red squares. You can modify their properties in any of several ways:

1. Select a load or immersion point by placing the mouse cursor over the object and pressing the left mouse key, and "dragging" it to a new position while keeping the key pressed.
2. Press Alt-I or Alt-L to select immersion points or loads, then use the Tab and Alt-Tab keys to move up or down the list of items. As you change your selection, the mouse cursor will be moved over the centre of the currently-selected object.
3. Select an immersion point or load using either of the above means, then either select menu item "Properties", or press the right mouse key ("right-click") while the mouse cursor is over an immersion point of load. A dialog box allowing you to edit all properties for the object will appear.
4. Select an immersion point or load, then use the arrow keys on the keyboard to move it. You can half or double the movement increment per keystroke using the D ("double") and H ("half") keys.

Naturally, you can only move an immersion point or load in two dimensions, but the three alternative views allow you to choose any pair, of the three available, at a time.

The program does not prevent loads from overlapping. This is necessary, because the masses of some loads may be zero (corresponding to absent loads) in some hull conditions. In these cases, it must be possible for other loads to be placed in positions overlapping the vacant space left.

4.7. Running a Template

Once you have saved your project, you can give it a try by applying one of Hullstat's templates to it. One of the commoner ones, like "Deadweight and Stability" is a good start point. If you select this (or any other), type in a name for the output file, and press the Ok button, the program will commence processing.

You will see some lines displayed in the white window at the bottom of Hullstat's window. When processing completes, you should see the message "** Run successfully finished **".

You can then view the result using Hullstat's "Edit output" button.

5. Templates

5.1. Template Formats

A template may be generated by any text editor, or by a word processor capable of generating rich-text-format ("RTF") files. The RTF format has advantages, not only in the provision for output of calculations in well-structured tables, control of fonts and so on, but also in that it permits inclusion of graphics directly in a report.

Text-format reports, however, can only be formatted using the "tab" character. When graphics are generated, a note is added indicating the graphic file to be inserted, but insertion itself is not possible.

The format of the template defines the format of the output file - i.e., text-format templates create text-format output, and RTF templates create RTF output.

5.2. The Hullform Statics Language

Hullstat templates include embedded commands which control the repetition of calculations for different operating conditions, communication with Hullform, and allow performance of arithmetic calculations. The command set functions in the same manner as a programming language, like Basic, and is termed the "Hullform Statics Language" ("HSL").

While programming languages like Basic, Fortran, C and Pascal look different on casual inspection, a more detailed review shows that they do much the same job with very similar commands. HSL should be viewed in this context - it is rather like Basic, but includes simplifications and extensions. The former are invoked in the interests of system management, the latter to handle better the processing tasks involved in processing statics data.

Also in the interests of simplicity, all commands and variable names are case-insensitive. Spaces are ignored, except within text strings.

In a Hullstat template, HSL commands are identified by lines commencing with the "#" character. This is often not shown in the explanations below, but is always required.

The details below presume the reader has some experience with the use of another programming language. The techniques of programming are presumed known already.

5.2.1. Variable Names

Size and format: HSL variables may have length up to 12 characters, and may comprise letters and/or numbers, as long as the first character is a letter. Names are case-insensitive, so the following names are all equivalent and legitimate:

TankMass03

TANKMASS03

tankmass03

Arrays: These may be created using the "dim" (for "dimension") command.

Array usage is very similar to that in Basic and Fortran, with the limitation that arrays may only be one-dimensional, and array indices are enclosed by square brackets ("[" and "]") rather than round brackets ("(" and ")"). The change - which parallels usage in C and Pascal - simplifies command interpretation, allowing round brackets to be reserved for expression grouping only.

Array size may be specified by a constant or an expression - for example, both of the following are valid:

```
#dim angle[20]
#dim GZtable[maxangle/increment+1]
```

Array indices can run from 1 to the array size. If an index out of this range is used, an error is reported, and processing terminates. When a non-integer index value is used, it is truncated down to the next highest integer - so arr[4.3], arr[4.9] and arr[4] all address the same data.

Data represented: A variable may contain either a text string, limited by the user-defined text data size (with a default of 40 characters), or a floating-point number. Integers are not supported, because all integer operations can be performed using floating point numbers, and integer output can be simulated using zero decimal places.

The logic behind the dual string / numeric identity of HSL variables is illustrated by the sample code below (which presumes understanding of common "if" block structures, and assignment operations)

```
#if divisor = 0
#  ratio="INVALID"
#else
#  ratio=dividend/divisor
#endif
```

The variable "ratio" may then be entered into a location in the output file, with a text or numeric value according to its definition. (See below for the full syntax of the conditional and assignment expressions used in the above code section)

Data limits: There is a default provision for up to 400 variables (including array elements) in Hullstat. This limit can be increased in the Edit, Preferences dialog box (at a small cost of memory available to other processes - see section 7.4), but is normally well in excess of requirements.

5.2.3. Constants

A constant is a numeric value or a text string. After an expression has been "parsed" into arithmetic operators and operands, a numeric value is identified as an operand commencing with any non-alphabetic character. This means that an invalid variable name like "4sale" will be identified as an invalid constant, but the nature of the error is actually arbitrary (and Hullstat has more important tasks to perform than discriminating between such errors!)

A numeric constant may comprise a leading + or - sign, followed by digits 0-9 and at most one decimal point.

The maximum length of a text string definable by the user is 128 characters (See section 7.4), but defaults to 40 characters. All strings must be enclosed between double quotation marks (for example, "Text data").

5.2.2. Assignment Commands

An HSL line commencing with a space, tab or the word "set" is an assignment statement. Assignment statements have form

value=expression

"value" may be any variable or array element. The array index may be a numeric expression.

"expression" may be an arithmetic expression, a text constant, or a text-valued variable or array element. Text expressions are not supported.

Arithmetic Operators: Arithmetic expressions may include arithmetic operators +, -, / , * and ^ (the last being the exponentiation operator, as in Basic).

Logical Operators: & ("and"), | ("or"), > ("greater than"), = ("equal to") and < ("less than") are also allowed. They presume that operands are either zero (the "false") value, or nonzero (the "true") value. Logical results are either 0 (false) or 1 (true). There is no restriction on the mixing of logical and arithmetic operations - for example, "(a | b)*4" is valid, giving potential results of 0 or 4.

The operator & has greater precedence than the operator |. Operators >, = and < have identical, lower precedence. Numeric operators all have precedence over logical operators. The precedence order is therefore

^	highest
*, /	
+, -	
&	
|	
> = <	lowest

Operators for "greater or equal to" and "less than or equal to" are not provided, because two-character operators are not yet handled by Hullstat.

Special Functions: Hullstat also allows use of common transcendental functions sin ("sine"), cos ("cosine"), tan ("tangent"), sqrt ("square root"), log10 ("logarithm to base 10"), log ("natural logarithm"), exp ("natural exponential"), abs ("absolute value"), atan ("inverse tangent"), acos ("inverse cosine"), and asin ("inverse sine"). For trigonometric functions, angles are presumed to be in degrees. (This is non-standard compared to other languages, but fits with the more common set of units in statics analysis)

Round brackets are used to control precedence of operations, as normal. Sample numeric expressions are

sqrt(a^2+400.0)

(a1*a2-a3*a4)/(a2-a3)

2.0+angle[j-3]/5.5

Expressions commencing with a "-" sign are currently not supported, although negative constants are. So "x = -4" is allowed, as is "x = -4*y", but "x = -y" is not (Use "x = 0-y" instead)

5.2.3. Comments

A comment may be inserted anywhere in a Hullstat template, the command commencing with a "!" character, or the word "rem" (as in Basic). A typical comment might look like:

```
#!
#!  Accumulate total hull mass
#!
```

(Observe that mainly-blank comment lines are placed before and after the comment text line. This is for clarity. Were a blank line only inserted, this would be copied to the output file)

5.2.4. Conditional Branching

Hullstat offers a full logical "if" structure, permitting and indefinitely long sequence of options to be tested. An "if" may be followed by an "endif" only, at the end of the block to be executed conditionally, one or more "elseif" commands, and one final "else" command.

Following the "if" and "elseif" there must be an arithmetic expression. If the expression evaluates nonzero (i.e., true), commands following (to the next "elseif", "else" or "endif") will be processed. Commands until the first after the "endif" will then be skipped.

If the expression evaluates as zero (i.e., false), the commands following (to the next "elseif", "else" or "endif") will be skipped. An "elseif" will then be treated as an "if", while commands following an "else" will always be executed.

An illustrative example follows:

```
#if x > 0 & y > 0
  # a=1
#elseif y < 0
  # a = 2
#else
  # a=3
#endif
```

5.2.5. Looping Constructs

Hullstat offers no "goto" for transfer of control to another point of the template. All such operations must involve explicit loops. This feature is not only good, standard programming practice, but also simplifies command processing.

There are five classes of loops in Hullstat, although four are specific cases of a standard loop construct. Of these, a two-way split occurs because loops can repeat while a condition is true, or until a condition is true. A further split occurs because the loop termination test can be performed at the start of a loop, or at the end of a loop. A preliminary overview of the loop forms associated with two-by-two splitting is provided by a table:

Termination test at ...	Start of loop	End of loop
Exit when expression is:		
true	**#loopuntil *expression*** ***commands here*** **#enduntil**	**#loop** ***commands here*** **#until *expression***
false	**#loopwhile *expression*** ***commands here*** **#endwhile**	**#loop** ***commands here*** **#while *expression***

Loops are repeated while “expression” is true, or until “expression” is true. Some examples are below:

```
# imax=10
# i=0
#dim y[imax]
#loopuntil i > imax
   # i=i+1
   # y[i] = i^2
#enduntil
```

This loop stores a table of the first 10 squares in array “y".

```
# i = 1000000
#loop
   #print i
   # i=i/2
#while i > 1
```

This displays a decreasing geometric sequence on-screen, and terminates when variable “i” reduces to a value less than or equal to 1.

Although Hullstat currently has no direct equivalent of the Basic “for ... next” loop, all normal looping constructs can be obtained from the above four.

The fifth option, “foreach", loops automatically through hull conditions, loads, immersion points or tanks. During these loops, standard variable names are generated, and related statics data are entered into them. The syntaxes for each of the four, and the variable names, are as shown in the following table.

```
#foreach condition
(HSL commands)
#next condition
```

condition	description
state	description

```
#foreach load
(HSL commands)
#next load
```

xload	longitudinal position
yload	lateral position
zload	vertical position
mload	mass of load
loaddesc	description

```
#foreach immersionpoint
(HSL commands)
#next immersionpoint
```

ximme	longitudinal position
yimme	lateral position
zimme	vertical position
immedesc	description

```
#foreach tank
(HSL commands)
#next tank
```

ftank	percentage full
gtank	specific gravity
ptank	permeability

(Note that more tank information is available from Hullform, using the “get” HSL command - see section 5.3)

Until the “foreach condition” loop has been commenced, the other loops have little meaning, since many parameters relate to both the hull condition and well as a specific load, immersion point or tank.

It is also possible to choose a hull condition, load, immersion point or tank individually - see the “select” command (section 5.2.9).

5.2.6. Data Transfer

The ready exchange of data between Hullform and the Hullstat module is central to most statics reporting tasks. This communication is effected by the “get“ and “put“ commands. Because of their importance, these are allocated separate sections in this manual (5.3 and 5.4).

5.2.7. Output and Input Commands

These allow monitoring of the progress of calculations, in the window at the bottom of the Hullstat dialog box, and in message boxes which can remain on-screen until discarded.

ondebug
nodebug These activate and deactivate display of every expression result, as it is evaluated. This slows processing of the template, but allows the proper functioning of the analysis to be checked.

decimal This is one way of controlling the precision of output. For example, “decimal 4” will result in four decimal places in output fields - until overridden by another “decimal” command, or a “%” in a substitution field (See section 5.5). “decimal 0” results in output without any decimal places, or decimal point.

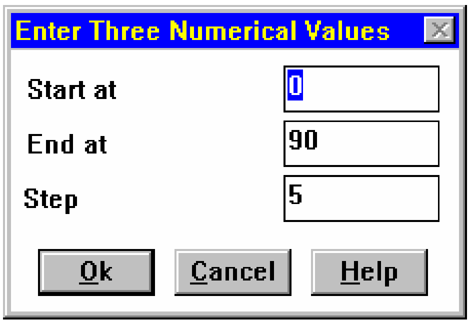
show The result of any expression may be shown in a dialog box (e.g., “show 4*(a-4)/(b-4)”). This command also halts processing until the dialog box has been closed. If the user clicks “Cancel” in the dialog box, processing is terminated.

print The result of any expression may be shown in the Hullstat dialog box window, providing well-controlled tracing of the progress of processing.

ask This provides a way of entering numerical values for variables used in the template. The syntax is “ask prompt, variable, prompt, variable,...". “Prompt” is a text string used in the request, and “variable” is the variable to which the input value is assigned. Up to three such requests can be placed in the same “ask” line. For example, the HSL commands -

```
# val1 = 0
# val2 = 90
# inc = 5
#ask "Start at",val1,"End at",val2,"Step",inc
```

- produce the dialog box below:



To ensure that the details of these inputs are formally recorded, they are always written to the output file. For the above “ask” format, the details entered would be:

Interactively Defined:
Start at (internal variable val1) = 0.000
End at (internal variable val2) = 90.000

Step (internal variable inc) = 5.000

The occurrence of this output should be borne in mind when designing a template. A useful approach is to place all such requests at the start of the template, with a page break following. The result will be a cover page with all “ask” values formally recorded. This policy has been followed in the provided sample templates.

5.2.8. Hullform Execution Commands

The “perform” command instructs Hullform to perform one of a range of actions, within its own command set. The syntax is “perform action arguments", where “action” may be one of:

open
The command has a further argument, the name of the hull file to be opened. “Filename” must include the “hud” extension, but need not include an explicit path specification. If it does not, Hullform will try to find the file in its default hull data directory.

balance
balanceall
evaluate
These commands equate to the Statics menu items “Balance", “Balance all” and “Evaluate". They are used once the “put” command has set up items such as displacement, LCG, heel, pitch and waterline offset, to update the program's hydrostatics variables. They require no further command arguments.

menunumber
This invokes a menu item with the specified number. Further arguments may be provided, to complete entries to a resulting dialog box.

Menu items are numbered from 101 upward in the File menu, 201 upward in the Edit menu, and so on.

Not all dialog boxes will accept the further arguments. The file dialog box is the most common case, but the text-mode section edit dialog box, and any dialog box which is generated by pressing a dialog box push-button, also do not support this feature.

5.2.9. Selection of Hull States

It is possible to choose any individual combination of operating condition, load, immersion point and tank using the “select“ command. The syntaxes are of form

select condition <index>
select immersionpoint <index>
select load <index>
select tank <index>

where “<index>“ may range from 1, to the maximum number defined. Once this command has been processed, all the variables listed

5.2.10. Graphics Commands

opengraphics
This requires no further command arguments. It initialises graphics output, to either a HPGL file or Windows metafile. It is required prior to any other graphics command.

The choice between HPGL and Windows metafile format is specified in the Preferences dialog box (See section 7.3).

closegraphics
This requires no further command arguments. It terminates graphics output, and is required for output to be complete.

graph x,y,n,xlabel,xmin,xmax,xinc,ylabel,ymin,ymax,yinc
This command automatically produces a graph which may be included with the statics report. The command arguments are as follows:

x an array of values which define the horizontal coordinates of the points on the line to be plotted.

y an array of values which define the vertical coordinates of the points on the line to be plotted

n the number of elements in the arrays x and y, above.

xlabel a variable or text string (enclosed in double quotation marks), used to label the horizontal axis.

xmin a constant or variable (not an expression), defining the value of the horizontal axis variable at the left edge of the plot.

xmax a constant or variable (not an expression), defining the value of the horizontal axis variable at the right edge of the plot.

xinc the increment on the horizontal axis at which intercepts are marked.

ylabel a variable or text string (enclosed in double quotation marks), used to label the vertical axis.

ymin a constant or variable (not an expression), defining the value of the vertical axis variable at the lower edge of the plot.

ymax a constant or variable (not an expression), defining the value of the vertical axis variable at the upper edge of the plot.

yinc the increment on the vertical axis at which intercepts are marked.

All of the above are required, but defaults are available for the plotting range parameters. If the provided value of xinc or yinc is zero, a value based on the range of data values is used. If the provided value of xmax is less than or equal to xmin, or ymax is less than or equal to ymin, the limits are taken from the range of data values, increased to ensure an integer number of increments “xinc” or “yinc” occur along the plot axis.

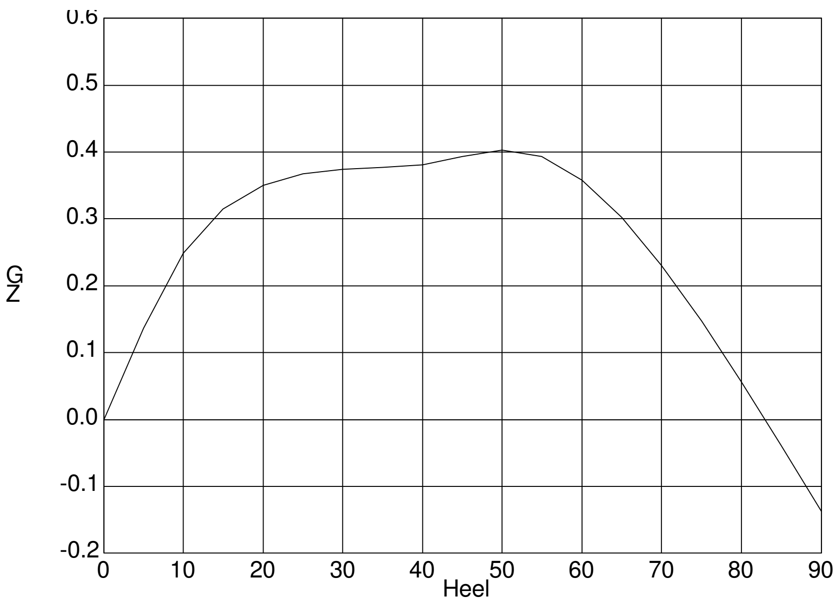
Below is an example of a graph generated by Hullstat. The command used was:

#Graph deg,gzt,N,"Heel",0,Heelmax,10,"GZ",0,0,0

Where “deg” and “gzt” were arrays, and “Heelmax” was 90. The effect of the automatic range calculation for the vertical axis is clear.

You can control the line width, colour and font used for a graph. Whatever setting these have when the “graph” command is used persists through the whole of the graphing operation.

The “graph” command also leaves the plotting ranges matching those used for the graph. This means you can annotate the graph, using the same co-ordinates as used for the horizontal and vertical axes on ther graph. For example, the command “move 45,0.2” locates the drawing position at the centre of the graph below.



move x,y
draw x,y
These allow arbitrary drawing on the plot area. The “move” command locates the drawing point at location (x,y), where the plot scaling has been predefined by use of the “graph” command, or by use of the “xmin", “xmax", “ymin” and “ymax” commands (See below). The “draw” command draws a line from the last drawing point to the specified one, leaving the drawing point at the end of the line drawn.

plottext string
This command adds text in variable or string constant “string", with its top left corner at the current drawing point.

plotvalue expression
This command writes the numeric value of “expression", with its top left corner at the current drawing point. Numeric values are written with the currently-defined output precision (See “decimal", page 19)

xmin expression
xmax expression
ymin expression
ymax expression
These define the limits of the plot area. The argument “expression” is any numeric-valued expression. For example, the following sequence plots a rectangle half the width and height of the plot area:

```
#xmin -10
#xmax 10
#ymin -20
#ymax 20
#move -5,10
#draw 5,10
#draw 5,-10
#draw -5,-10
#draw -5,10
```

The limits of the plot area are also set by the “graph” command, so these commands are only rarely needed.

colour expression (Windows metafile only)
color expression (Windows metafile only)
The integer part of “expression” ensures drawing in the following colours:

1	white	2	yellow	3	cyan	4	magenta
5	green	6	blue	7	red	8	black

The alternate term “color” is to ensure that either English or American spelling will work.

linewidth expression (Windows metafile only)
This sets the width of drawn lines. A value of zero generates a line of minimum drawable width.

linestyle expression (Windows metafile only)
Sets the style of lines to be drawn. Codes for supported styles are:

0	Solid	1	Dashed	2	Dotted
3	Dash-dot	4	Dash-dot-dot		

The nominated styles only take effect if the specified line width is zero. If a line width greater than zero is specified, a solid line is drawn.

textsize expression (Windows metafile only)
"Expression” is the text character size, in points. The default text size is 12 point.

textfont expression (Windows metafile only)
"Expression” is a text constant or text-valued variable, which defines the font used for subsequent plotting of text. It must match a defined font on your system (e.g., “Courier", “Arial” or “Times New Roman”). The default is “Arial".

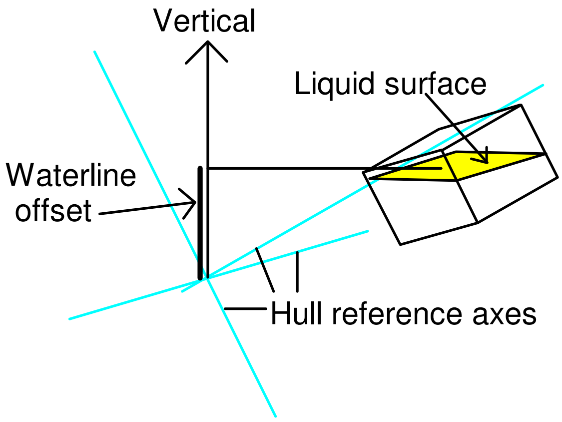
5.3. Data Available from Hullform - the “Get“ Command

This command requests information from the program. The information comprises any of the program's internal statics variables, internal tank information, or character strings defining measurement units.

The following item names are accepted by Hullform for transmission to Hullstat. When indexed ("[index]”), the supplied index must be an integer from 1 to the number of tanks for the currently-loaded hull design, and the returned value applies to the indicated tank. Again, the program's request input is case-insensitive.

Tank Properties

tankct	number of tanks defined for the currently-loaded hull design
tankds[index]	text descriptor for each tank
tanksg[index]	specific gravity of contents for each tank
tankfr[index]	fraction full, as a percentage, for each tank
tankpm[index]	permeability, as a fraction from 0 to 1, for each tank
tankvl[index]	total volume of a tank. The volume contained is tankfr[index] x tankvl[index]
tankwl[index]	the level of the fluid surface ("waterline”) within the tank. The value returned is the intersection coordinate of the plane of the fluid surface and a true vertical line at the hull centreline and at the longitudinal position zero.



tankvg[index]	vertical centre of gravity position of the tank contents, measured relative to the hull reference axes.
tanklg[index]	longitudinal centre of gravity position of the tank contents.
tanklf[index]	longitudinal centre of flotation position of the tank contents.
tankfm[index]	free surface moment of the tank's contents.
tankmo[index]	longitudinal moment of the tank's contents
tankms[index]	mass of tank's contents

Measurement Units

lenun	length unit, text string of two characters, either “ft” or “m".
masun	mass unit - text string, “lb", “ton", “kg” or “tonne"

Hydrostatic Parameters

aabove	profile area of the hull above the waterline
abelow	profile area of the hull below the waterline
displa	displacement, in the current mass measurement units

bmax	maximum beam
bwl	maximum waterline beam
cm	midsection coefficient
cp	prismatic coefficient
cb	centre of buoyancy
densty	density of water, in the current measurement units.
draft	maximum draft (actual, not of the vertical zero reference line)
draftf	draft of the hull at the base of the stem section
draftm	draft of the hull at the section nearest midships
drafta	draft of the hull at the stern section
entang	waterline entry angle - the angle made by the hull's two-dimensional waterline and the hull axis, at the waterline entry point.
freeba	freeboard at the stern section
freebd	freeboard of the hull at midships
freebf	freeboard of the stem section
freebm	freeboard of the hull at the section nearest midships
gravty	acceleration due to gravity, in the current measurement units
gz	horizontal distance between the centre of mass and buoyant centre of the hull
heel	heel, in degrees to starboard
lcb	longitudinal position of the centre of buoyancy of the hull, in the current length units, relative to the chosen zero location
lcf	longitudinal position of the centre of flotation of the hull, in the current length units, relative to the chosen zero location
loa	length overall, taken as the distance from the foremost point of the stem to the transom section
lwl	length on the waterline
mct	moment to change trim, per centimetre (for metric units) or per inch (for British units)
mpi	extra displaced mass per unit distance of immersion
mxarea	maximum cross-sectional area of the hull below the waterline
pitch	pitch up of the hull, in degrees
sarm	"small amplitude righting moment", in units of (mass unit) x (length unit) per degree of heel
spgrav	specific gravity of water
volfor	volume displaced by the hull, forward of the section of largest area
volste	volume displaced by the hull, sternwards of the section of largest area
volume	displaced volume of the hull
waterl	waterline offset relative to the chosen hull baseline, positive upward
wetsur	wetted surface of the hull
wplane	waterplane area
xatbwl	longitudinal position, relative to the chosen longitudinal zero position, of the maximum waterline beam of the hull

xcofb	longitudinal position, relative to the chosen longitudinal zero position, of the centre of buoyancy of the hull
xcofm	longitudinal position of the centre of mass, in the current length units, relative to the chosen zero location
xentry	longitudinal position of the waterline entry point, relative to the chosen longitudinal zero position
xlcf	longitudinal position, relative to the chosen longitudinal zero position, of the centre of flotation of the hull
xmid	longitudinal position, relative to the chosen longitudinal zero position, of the midships point of the hull
xstem	longitudinal position of the foremost point of the hull, relative to the chosen longitudinal zero position
xstern	longitudinal position of the sternmost point of the hull, relative to the chosen longitudinal zero position
zabove	mean height of the area of the hull above the waterline (positive)
zbase	vertical position of the lowest point of the hull
zbelow	mean height of the area of the hull below the waterline (negative)
zcofb	vertical position of the centre of buoyancy of the hull
zcofm	vertical position of the centre of mass of the hull
zbase	vertical coordinate of the lowest point of the hull
zmeta	vertical coordinate of the buoyant metacentre of the hull

5.4. Sending Data to Hullform - the “Put“ Command

In a manner equivalent to “get", some variables within Hullform can be altered through Hullstat. The “put” command can be used to set only the parameters which determine the hydrostatic state of the hull. Derived parameters (such as heeling moment) may not be set, since there is no purpose in doing so.

Specifically, the parameters to which access is permitted are:

waterl	waterline offset, positive upward from the chosen zero reference (or any name commencing with these letters)
heel	heel to starboard, in degrees
pitch	pitch up of stem, in degrees
disp	displacement, in the chosen measurement units (or any name commencing with these letters)
lcg (or xcofm)	longitudinal position of the centre of gravity, in length units.
vcg (or zcofm)	vertical position of the centre of gravity, in length units, positive upward
tanksg[number] tankfr[number] tankpm[number]	specific gravity of the contents, permeability and fractional filling, of the tank whose index number is provided in the brackets. “Number” must be a valid numeric expression, whose integer equivalent ranges from 1 to the number of tanks defined for the current hull.

If the “put” item name does not match one of these, Hullform reports an error, and processing terminates.

5.5. Substitution Fields

In blocks of text, output may be modified using three operators. These are the back-quote ("`”), two of which enclose a numerical expression whose value replaces the expression and back-quotes, the percent ("%”), which alters output precision, and the backslash ("\”), which assists column formatting.

The “Back-Quote“ Operator

Estimates of hydrostatic variables are inserted into the output report using substitution fields. These are numerical expressions or text variables, enclosed within back quote characters ("`”) in the non-HSL components of the template. A sample line of text in a template might therefore read:

The displacement of the hull is `displa+loaddisp+tankdisp` `masun`. At a heel of `heel` degrees, its righting lever GZ is `gz` `lenun`.

Here, “displa", “loaddisp” and “tankdisp” correspond to the displacement due to the bare hull, due to static loads and to the contents of internal tanks. Variables “masun” and “lenun” contain the mass and length units, as text strings. “Heel” and “GZ” have their conventional meanings.

The line would appear in the output file in a form such as:

The displacement of the hull is 8730.102 kg. At a heel of 20.000 degrees, its righting lever GZ is 0.305 m.

String expressions can be created by insertion of substitution fields into a string constant. For example,

```
# x = 1.234
# y = 5.0
# label = "value of x+y is"
# result = "The `label` `x + y`"
```

Evaluates to give “result” equal to “The value of x+y is 6.234".

The “Percent“ Embedded Operator

The above lines are written with a fixed output precision of three decimal places. However, variation of output precision from field to field in one line is often needed. A “precision” HSL command would require a separate line in the template between substitution fields, and so would force a line break.

For this reason, a substitution field may also include a precision modifier. The specified precision must be an integer, defines the number of decimal places to be used, and is indicated within a substitution field using the “%” character, in form `%p expression` (for precision “p”). For example, the previous paragraph could have been written:

The displacement of the hull is `%0 displa+loaddisp+tankdisp` `masun`. At a heel of `heel` degrees, its righting lever GZ is `%4 gz` `lenun`.

and the result would have been

The displacement of the hull is 8730 kg. At a heel of 20 degrees, its righting lever GZ is 0.3054 m.

Observe that “%0” results in data output without the decimal point.

When a precision modifier is used, it also defines output precision for all following substitution fields (until changed by another % specification, or a “decimal” HSL command).

The “Backslash” Operator

Another special substitution syntax is necessary only when a text-form template is used. Due to the variable size of substitution strings, it maybe difficult to be locate following columns properly. The “\” syntax places text exactly at standard tab columns (at 8-character intervals).

For example, the template text

```
`index`  `parameter`       `value`
         `par`             `unit`
```

looks good in the template, but the corresponding text in the output file would be like

```
4        Mass    12.345
         Data              kg
```

The “12.345” text has moved to the left because the evaluated result, “Mass", is much shorter than the substituted text ("`parameter`”). To avoid this problem, we can use syntax

```
`index`  `parameter`       \3`value`
```

`par` \3`unit`

The “\3” operator instructs Hullstat to commence following text at the third tab point, if possible - giving, in the above case,

4	Text	12.345
	Data	kg

We could have used “\2” instead, giving an equally well formatted result:

4	Text	12.345
	Data	kg

However, “\1” would not have worked, since text preceding the “\” character has already extended past the first tab point.

The “\” may be following by any digit from one to nine, inclusive, locating text explicitly at up to nine columns across the page.

5.6. How to Create a Template

Creating a Hullstat template “from scratch” involves a combination of document formatting and programming. Many users will have experience in setting up tables in word processor or text documents, but equally many will not be comfortable with writing in a simple programming language. The following presumes your have experience with both.

It also presumes (in general, but not totally) that you have access to a Windows word processor, capable of generating RTF files, and of incorporating formatted tables.

5.6.1. Entering Header Information

Normally, you will have in your possession a previous stability report, following the desired format. This is an excellent start point, but if you have not, you can achieve the same end with some keyboard input.

Firstly, check how well your intended report format matches the “operating condition” set used by Hullstat. If you can select a page (or a few pages) corresponding to one operating condition, this or these will form the basis of your report template. Copy the text, tables and any graphs to a new document.

If you do not have a suitable report available, you will have to create the pages by hand. At this stage, do not worry about anything but the layout - enter hypothetical titles, leave tables empty, and so on.

When you have a layout completed, and entered into a file, you can start converting it to a Hullstat template.

Firstly, you should identify parts of the page(s) which correspond to a description of the “operating condition” and the “state” of the vessel. The text you have used as the page or section header is probably the operating condition. So, for example, if your heading reads something like ...

Fully Loaded, Departure for Fishing

... you need to be able to replace it with the operating condition of the hull. The way you do this is to use the back-quote substitution field, with variable name “condition". Keep the same text font, size and attributes (15 point bold Arial, border below, in the above case), and just replace the text - to give ...

`condition`

When you process the template, the condition description will replace the above field.

Next, see if there is subsidiary descriptive text which fits the “state” variable. This can be up to 128 characters (as long as you set such a limit under the “Preferences” button - see section 7.4), extending over a line or two. If so, replace it in the document with the field “`state`".

Before you look into the development of data to go into a table or graph, you should look at any other information on the page. Where the vessel name occurs, replace it with field “`name`".

There are other, non-automatic parameters you may want to enter into the template header details. For example, you may want to log the density of water. You can obtain the specific gravity from Hullform, and report it as follows:

```
#get spgrav
```

Water S.G.	`spgrav`

The “#” line is a HSL command, which will be discarded from output. It will provide a value for variable “spgrav", which is used on the second of the above lines.

A different font is used above for the HSL command - this is not essential, and in some cases can cause confusion (e.g., if hidden text is used in Word for Windows, any text its generates it is also hidden).

5.6.2. A Simple Table

You may also want to enter some summary details about the design in this condition - for example, freeboard forward (Hullform recognises “freebf” for this) or freeboard aft ("freeba”). Lets say you want freeboard forward and aft, and draft forward and aft. To extract these values, you would use HSL commands

```
#set heel=0
#put heel
#perform balance
#get freebf
#get freeba
#get draftf
#get drafta
#get masun
#get lenun
```

Remember that the returned hydrostatic values have no meaning unless the hull is properly trimmed before the “get” commands are issued. It is also wise to set heel to zero, using the “put” command, since there is no way of being sure the hull is upright when Hullstat is started.

Notice that the length and mass units “masun” and “lenun” have also been obtained from Hullform.

When you have retrieved the values you want, you can enter them into a table as substitution fields. For an RTF file, you can enter the table in your word processor's normal manner. Notice the use of the “%” operator to produce output to 3 decimal places:

	Forward	Aft
Freeboard (`lenun`)	`%3 freebf`	`%3 freeba`
Draft (`lenun`)	`draftf`	`drafta`

For a text-form template, you can use the Tab key format the table. The backslash operator (page 26) is also useful here. For example, the following three lines will result in the location of a “forward freeboard” column at tab position 2, and an “aft freeboard” column at tab position 3:

```
\2Forward\3Aft
Freeboard (`lenun`)\2`%3 freebf`\3`%3 freebf`
Draft (`lenun`)\2`draftf`\3`drafta`
```

You can add spaces or tabs into these lines, for clarity. As long as they do not force the text beyond the required tab spots, they will do no harm.

5.6.3. A Hydrostatic Table, Arranged Down the Page

There are two ways to arrange a table of hydrostatic parameters. Your independent variable (the one that you define, and is not defined by another variable - e.g., heel angle) may run across the page or down the page.

If it runs across the page, you will be limited in the number of values you can use before the table runs off the right edge. If it runs down the page, you will be less limited, and setting up the template will also be a little easier. We will start with the easier option.

Firstly, you must decide what is to be tabulated. For this example, we will try a table of draft, initial stability, wetted surface, waterline length and LCF (if only to prove that any such combination is possible), defined by a range of displacements. We start by creating three tables: the first holding the header line, the next to receive the results, the last to form the bottom of the table. Initially, they should look something like this:

Disp. (`masun`)	Draft (`lenun`)	Stability (`masun` `lenun` per degree)	Wetted Surface (sq `lenun`)	LWL (`lenun`)	LCF (`lenun`)
`disp`	`draft`	`sarm`	`wetsur`	`lwl`	`lcf`

(Remember “sarm” is short for “small angle righting moment”) The header table line includes the mass and length units, as substitution fields. It has been formatted with full drawn borders, but this is a matter of preference only. The next line includes the required hydrostatic values as substitution fields, while the last is empty. (The last has been formatted to a height of only 3 points - a quarter of a line - because it is a bottom border only).

With the table lines created, we now must interpose some HSL commands to repetitively generate the middle line, for each required displacement. This is easily done, using a loop construct (and a little programming experience).

But what displacements do you want to use? Will they always be the same? Of course not. You need some way to define these, differently for each use of the template.

While you could customise the template each time you used it, there is a better way. The “ask” command is provided for exactly this purpose. Go back to the start of your document, and enter the line:

#ask “Start disp.",startdisp,"End disp.", enddisp,"Increment",inc

Leave it in normal text - it generates output to your document as well as defining useful parameters, and you want to see the result as normal text. You can also create default values, by pre-defining the variables - e.g., you can enter the following lines before the “ask” command:

```
#get displa
#set startdisp=0.5*displa
#set enddisp=1.5*displa
#set inc=0.1*displa
```

Be sure the “ask” command is located outside the “foreach condition” loop which repeats the middle line of the table, or you will be asked the same question over and over again.

Now you can construct your table. Immediately after the table header line, enter the HSL commands

```
#set disp=startdisp
#loopwhile disp < enddisp + 0.5*inc
```

These two lines initialise the loop variable “disp", and commence a loop which will continue until the end value is reached. The “0.5*inc” is added because arithmetic with real numbers is never 100.00000 % accurate, and because you will probably want to include the value of “enddisp” in the table.

Now you must balance the hull for the given displacement, and extract the required hydrostatic variables. Hullform will not recognise your parameter “disp", so the required commands are:

```
#set displa=disp
#put displa
#perform balance
#get draft
#get sarm
#get wetsur
#get lwl
#get lcf
```

The second table line must follow these. After the table line, you complete the loop by incrementing the displacement, and ending the “while” construct:

```
#set disp=disp+inc
```

```
#endwhile
```

When all displacements have been processed, the third table line now must follow, ending the table.

5.6.4. A Hydrostatic Table, Arranged Across the Page

In the previous section, table construction was simplified because one line of output corresponded to one hydrostatic state. With a variable like heel or displacement changing across the page, all such states must be analysed before one line is completed. The best way to manage such a combination is to use Hullstat's array handling ability.

However, before you start, remember that you can not format a variable number of columns in a table. You can respond to this limitation by setting a start and end for your table variable, and using the column limit to define an increment, or setting a start and increment, so that the column limit defines the end value. You can use even more sophisticated tricks - see section 5.7.1, for example.

Anyway, lets presume you want a table at uniform heel intervals, and have room for eight columns. At the top of your template, you can define the values used, using the “ask” command - e.g.,

```
#set startangle=10
#set incr=10
```
#ask “Start angle",startangle,"Increment",incr

These three commands will result in a proposal which will give eight columns, for heel angles of 10 to 90^{o}. Then you add lines (also at the top) which reserve space for your eight columns of statics calculations:

```
#dim heeltab[8]
#dim rmtab[8]
#dim kntab[8]
#dim wstab[8]
#dim lcftab[8]
```

The table data are then generated in a loop, of form such as

```
#set heel=startangle
#set index=1
#loopwhile index < 9
  #put heel
  #perform balance
  #get gz
  #get kn
  #get wetsur
  #get lcf
  #set heeltab[index]=heel
  #set rmtab[index]=gz*displa
  #set kntab[index]=kn
  #set wstab[index]=wetsur
  #set lcftab[index]=lcf
  #set heel=heel+incr
  #set index=index+1
#endwhile
```

To create the table, generate a table with nine columns and five rows. The first column should contain descriptions (Heel, Righting moment etc). Enter from column two of the first row of the table the text (in order) \`heeltab[1]\`, \`heeltab[2]\`, \`heeltab[3]\`, and so on. Place in the second row \`rmtab[1]\`, \`rmtab[2]\`, \`rmtab[3]\`, and so on, and repeat the same sequence for KN, wetted surface and LCF.

5.6.5. A Graph

After generating arrays such as in section 5.6.4, it is very easy to produce a graph. You need only enter as few as three commands - to open graphics, plot the graph and close graphics. For example, to plot the ten values of plot wetted surface against heel, you could use

```
#opengraphics
```

```
#set endangle=startangle+10*incr
#Graph heeltab,wstab,10,"Heel",0,endangle,10,"WS",0,0,0
#closegraphics
```

5.7. Sample Templates

Below is a review of the methods used in one of the sample templates, "KNTABLE". It should give you some perspective of the data processing techniques available through Hullstat.

In the sample below, HSL commands are shown bold. Descriptive comments are shown in italics

5.7.1. Template "KN Table, horizontal format"

KN tables for Design `hull`

This is a header line, shown only once on what is to become a cover page for the report. Observe the "hull" variable name - this is automatically generated, and contains the text entered at the "Vessel name" edit box in the Edit, Project dialog box

```
#runsummary
```

This line will write the names and file dates of the hull data file, project file and template used to generate the calculations shown in the following report. It is a crucial component of the quality assurance aspect of Hullstat (See section 1.2)

```
# numcond=0
#foreach condition
# numcond=numcond+1
#next condition
```

This block of code counts the number of operating conditions, for later use.

```
# fineinc=5
# fineend=20
# coarseinc=10
#ask "Fine heel inc.",fineinc,"Fine inc. end",Fineend,"Coarse heel inc.",coarseinc
```

A dialog box results from processing of this command. You can enter whatever values you require here. The parameters defined are always written to the output file, as part of the quality assurance process.

```
#get displa
#get zcofm
#get xcofm
# startdisp=displa
# startzcofm=zcofm
# startxcofm=xcofm
#get masun
#get lenun
```

In these lines, Hullstat obtains values for three hydrostatic parameters from Hullform, and preserves them for later re-use. The units of mass and length used for design of the hull are also retrieved.

```
# startdisp=0.5*displa
# enddisp=1.5*displa
# dispinc=0.10*displa
#ask "Start disp.",startdisp,"End disp.",enddisp, ..
```

These four lines define default values for the displacement range and increment, then present a dialog box allowing you to modify them.

```
#dim heelv[11]
#dim knv[11]
```

The hydrostatic table has up to 11 columns - these tables hold the corresponding heel and the KN estimates from Hullform.

```
#get tankct
#dim tankfr[tankct]
#dim tanksg[tankct]
```

Here we have found the number of tanks in the design, and reserved space for them. If you have no tanks - i.e., tankct = 0, you must be careful not to refer to the arrays, since an error will result (e.g., refer to them only in a loop "foreach tank")

```
# i=0
# h=0
#loopwhile i < 12
  # i=i+1
  #if h < fineend
    # h=h+fineinc
  #else
    # h=h+coarseinc
  #endif
  # heelv[i]=h
#endwhile
```

This is a common type of HSL loop. If steps through all columns of the table, finding the heel value to use for each.

```
# cond=0
#foreach condition
  # cond=cond+1
```

Here we are starting the main program loop - repeating all analyses for each operating condition. Observe that "cond" holds the condition index.

```
  #print condition
```

The progress of template processing can be traced, by printing suitable values in the display window. Remember that "print" displays a value there and continues; "show" displays a value in a dialog box, and waits.

```
  # i=0
  #foreach tank
    # i=i+1
    # tankfr[i]=tankfrac
    # tanksg[i]=tankspgr
    #put tankfr[i]
    #put tanksg[i]
  #next tank
```

In this block of code, tank properties are obtained from the variables "tankfrac" and "tankspgr" initialised at the "foreach tank" loop start, placed in an array which will be recognised by the Hullstat - Hullform communication module, and sent to Hullform

```
  # ZMsum=startdisp*startzcofm
  # Msum=startdisp
  # XMsum=startdisp*startxcofm
```

These lines determine the mass and moments due to the bare hull ("lightship").

```
  #foreach load
    # ZMsum=ZMsum+Mload*Zload
    # XMsum=XMsum+Mload*Xload
    # Msum=Msum+Mload
  #next load
```

This automatic loop accumulates the moment and mass contributions of all static loads

```
  # xcofm=XMsum/Msum
  # zcofm=ZMsum/Msum
  # displa=Msum
```

Now the centres of mass have been found from the mass and moments.

```
  #put zcofm
  #put xcofm
  #put displa
```

These revised values are sent back to Hullform, for use in statics calculations.

..

Hull state: `state`

A header line for the second page

Below is a table, which receives the column labels. Both text and numeric substitutions have been used. (Some of the 11 columns have been truncated)

		Righting Levers (KN) - \`lenun\`						
Displacement \`masun\`	Draft \`lenun\`	\`%0 heelv[ 1]\`	\`heelv [2]\`	\`heelv [3]\`	\`heelv [4]\`	\`heelv [5]\`	\`heelv [6]\`	

```
# displa=startdisp
#loopuntil displa > enddisp+0.5*dispinc
```

*We now loop through all requested displacements. The "0.5*dispinc" extra at the end is simply to ensure that truncation errors do not result in the last displacement ("enddisp") being missed.*

```
# i=0
#put displa
#decimal 0
#print displa
# heel=0
#put heel
#perform balance
#get draft
```

Above, the required displacement and a heel of zero have both been transmitted from Hullstat to Hullform, and the draft retrieved for insertion into the table.

Now we loop through all hell values, balancing the hull and retrieving the righting level KN:

```
#loopwhile i < 11
   # i=i+1
   # heel=heelv[i]
   #decimal 0
   #print heel
   #put heel
   #perform balance
   #get kn
   # knv[i]=kn
#endwhile
```

Once all KN values have been found, we can insert them into a line of the table (some columns truncated, again)

\`%0 displa\`	\`%3 draft\`	\`KNV[1 ]\`	\`KNV[2 ]\`	\`KNV[3 ]\`	\`KNV[4 ]\`	\`KNV[5 ]\`	\`KNV[6 ]\`

Then we advance to the next displacement

```
# displa=displa+dispinc
#enduntil
```

When the loop is done, we advance to the next operating condition.

```
#next condition
```

5.7.2. Template "Deadweight table, short form"

This generates a deadweight table, summarising the mass and moment (including free surface moment) contributions of the bare hull, all statics loads and all internal tanks.

Deadweight Tables for Design \`hull\`

This is a header line, as in section 5.7.1.

```
#runsummary
```

This line will write the names and file dates of the hull data file, project file and template used to generate the calculations shown in the following report.

```
#Dim zimm[20]
#Dim aimm[20]
#Get tankct
#Get lenun
#Get masun
#Get Displa
#Get Zcofm
#Dim tankmass[tankct]
#Dim tankvl[tankct]
#Dim tankvg[tankct]
#Dim tankfm[tankct]
#Dim tanklg[tankct]
#Dim tankds[tankct]
#Get densty
#Get spgrav
#set massconv=0.01*densty/spgrav
```

The above lines initialise arrays, and retrieve overall parameters for the design from Hullform. The "massconv" variable is used to convert tank contents to mass, later in the template.

```
#foreach condition
#print condition
```

Above, a loop through all operating conditions is initialised, and the text is displayed on-screen, to inform the user of progress.

....

The above line signifies a page break in the template - so that the "runsummary" information remains on a separate page.

\`condition\`

\`state\`

These lines are placed at the top of the page

Deadweight Table

Water S.G.	\`spgrav\`
Fresh Water S.G.	1.000
References	Longitudinal about stemhead
	Vertical about baseline
	All units in metres.
	Moments are \`masun\` metres.

These lines summarise the terminology used in the report. Observe that metric units are presumed - if you wanted to support both metric and British units, you could replace the last line with "Moments are \`masun\` \`lenun\`."/

```
#!
#! Find total displacement for this hull condition,
#! adding load and tank contents
#!
```

Comments like these lines are useful for anyone who may want to alter the template later, to help anyone to understand better what it is doing - or to help you understand what you were thinking about, when you designed this template so long ago!

```
#Set Totdisp=displa
#Foreach load
  #set totdisp=totdisp+Mload
#Next load
```

These lines calculate the total mass of the bare hull plus static loads (Tank contributions come later)

```
#!
#! Set up tank contents
```

```
#!
#Set i=0
#Foreach tank
   #Set i=i+1
   #Set tankfr[i]=ftank
   #Set tanksg[i]=gtank
   #Set tankpm[i]=ptank
   #Put tankfr[i]
   #Put tanksg[i]
   #Put tankpm[i]
#Next tank
```

The purpose of the above section is to extract from Hullstat the variables ftank (fraction filled, in percent), ptank (permeability) and gtank (specific gravity) which are updated automatically when the "foreach tank" loop commences, to copy them to arrays whose names which tell Hullform both the variable concerned and the tank index, and to transfer them to Hullform.

```
#perform balance
```

Once the tank contents have been defined, a "balance" operation is required to make Hullform calculate other tank values - such as water level and free surface moment.

```
#!
#! Get tank load contributions
#!
#decimal 3
#Set i=0
#Foreach tank
   #Set i=i+1
   #Get tankvl[i]
   #Set M=tankvl[i]*gtank*ptank*ftank*massconv
   #Set tankmass[i]=M
   #Set totdisp=totdisp+M
#next tank
```

Now each tank volume content can be retrieved, its mass found and so the total hull mass calculated.

Below is the header for the deadweight table (some right-hand columns truncated for printing on a narrower page):

Deadweight Item	Weight \`masun\`	LCG Metres	Long. Moment	VCG Metres

Now we calculate moments due to the bare hull and static loads, and enter them into the table

The commands below calculate the total longitudinal moment (XMload), transverse moment (YMload) and vertical moment (ZMload) for all loads. The index variable "i" is only used for display purposes, being entered into the first column of the table.

```
#Set i=0
#Set Msum=0
#Set ZMsum=0
#Set YMsum=0
#Set XMsum=0
#Foreach load
   #Set i=i+1
   #Set XMload=Xload*Mload
   #Set YMload=Yload*Mload
   #Set ZMload=Zload*Mload
   #Set XMsum=XMsum+XMload
   #Set YMsum=YMsum+YMload
   #Set ZMsum=ZMsum+ZMload
   #Set Msum=Msum+Mload
```

\`%0 i\`	\`loaddesc\`	\`%3 Mload\`	\`Xload\`	\`XMload\`	\`Zload\`

```
#Next load
```

The tank terms have now been completed. We proceed to find the tank terms:

```
#!
#! Find tank moments
#!
#Set j=0
#Set FSM=0
#foreach tank
   #Set i=i+1
   #Set j=j+1
```

"tanklg[i]" is the lcg value of tank number "i". Similarly, tankvg[i] is its vcg value, and tankfm[i] is its transverse free surface moment.

```
   #Get tanklg[j]
   #Get tankvg[j]
   #Get tankfm[j]
   #Set FSM=FSM+tankfm[j]
   #Set Msum=Msum+tankmass[j]
   #Set XM=tanklg[j]*tankmass[j]
```

(Remember that tankmass[i] was calculated earlier)

```
   #Set XMsum=XMsum+XM
   #Set ZM=tankvg[j]*tankmass[j]
   #Set ZMsum=ZMsum+ZM
   #Set DZ=tankfm[j]/totdisp
```

A common problem in formatting output is that, when a tank is empty, its contributions are all zero, but its lcg and vcg positions are undefined. These values are returned by Hullform as "-999.9" and "999.9" error codes respectively. We can use the dual identity of Hullstat variables (numeric and text) to replace the error codes by blanks, so that an empty cell will occur, rather than a mysterious-looking number.

```
   #if tanklg[j] = -999.9
      # tanklg[j] = " "
   #endif
   #if tankvg[j] = 999.9
      # tankvg[j] = " "
   #endif
```

The calculated values are now inserted into the table. While the substitution strings may overflow a line, the numerical values will not.

`%0i`	`tankds[j]` `ftank` %	`%3 tankmass[j]`	`tanklg[j]`	`XM`	`tankvg[j]`

```
#Next tank
#!
#! Find deadweight contributions
#!
#if Msum > 0
   #Set Xdw=XMsum/Msum
   #Set Zdw=ZMsum/Msum
#else
   #Set Xdw=" "
   #Set Zdw=" "
#endif
#decimal 3
#Get xcofm
```

This command gets the centre of mass position of the hull. Hullform also recognises the term "lcg".

```
#Set XMtot=XMsum+Xcofm*Displa
#Set ZMtot=ZMsum+Zcofm*Displa
#Set VCG=ZMtot/totdisp
```

Summaries of the totals are now added to the bottom of the table:

Deadweight total Lightship	\`Msum\` \`displa\`	\`Xdw\` \`Xcofm\`	\`XMsum\` \`Xcofm*displa\`	\`Zdw\` \`Zcofm\`
Displacement	\`totdisp\`	\`XMtot/totdisp\`	\`XMtot\`	\`VCG\`
Corrected VCG (Fluid)				\`VCG+FSM/totdisp\`

5.7.3. Template (Stability table, vertical format”

This template produces a stability summary, following closely the USL requirements. A graph of GZ against heel is also generated.

We commence with a few header lines, as before, then force a page break:

```
#runsummary
```

....

Informative text is added next

Stability Summary

\`condition\`

\`state\`

All areas under the GZ curve are shown in Metre Degrees

1 Metre Radian = 57.296 Metre Degrees

This template is set for fixed intervals from 0 to 90^{o} by steps of 5^{o}. This could be altered, using the “ask” command.

```
#! Initialise for GZ integration
#Set heelinc=5
#Set heelmax=90
```

The size of the arrays is determined once the number of heel values can be calculated:

```
#Set n=heelmax/heelinc+1
#Dim gzt[n]
#Dim deg[n]
#Dim area[n]
```

Later, we will need to know whether an immersion point has been immersed. When the immersion angle is found for any such points, the corresponding array value is set to a non-negative heel value. We can therefore set all array values negative initially; those negative at the end of the process correspond to immersion points which were never immersed.

```
#!
#! Set immersion angles “undefined"
#!
#Set i=0
#Foreach immersionpoint
  #Set i=i+1
  #Set Aimm[i]=-1
#Next immersionpoint
```

Now we initialise a range of variables which we are trying to define in the loop: the array index "i", the hell value, the area under the GZ curve "a", the maximum GZ value, the angle of the maximum GZ, the maximum GZ value beyond 30^o, and the angle of the maximum GZ beyond 30^o.

```
#Set i=0
#Set heel=0
#Set a=0
#Set maxgz = 0
#Set amaxgz = 0
#Set maxgz30=0
#Set amaxgz30=-999
```

We start at heel angle zero. Due to asymmetrical internal tanks, GZ may be nonzero in this state.

```
#Put heel
#perform balance
#Get gz
```

The first downflooding angle forms a limit to some GZ calculations. A value greater than the maximum heel to be used (90^o) indicates that no downflooding angle has been found.

```
#Set firstDFA=200
#!
#! Integrate areas to 90 Deg., and to the first
#! downflooding angle
#!
```

The test for exiting from the following loop is at the end of the loop, so it just starts "loop".

```
#loop
  #Set i=i+1
```

We save the previous GZ value, because it may be needed later, for one end of a linear interpolation.

```
  #Set gzv=GZ
```

The current heel value is displayed, to show progress of processing:

```
  #print heel
```

The hull is then balanced at the current heel value

```
  #put heel
  #perform balance
  #Set cosh=cos(heel)
  #Set sinh=sin(heel)
```

We extract from Hullform the new values of GZ, of the waterline offset, and the pitch. These allow us to calculate the waterline clearance of any point defined in the hull design coordinates.

```
  #get gz
  #Get Waterl
  #Get pitch
  #Set sinp=sin(pitch)
  #Set cosp=cos(pitch)
```

The lines below calculates the total moment of all static loads.

```
  #Set LM=YMsum*cosh+ZMsum*sinh
```

Now we check to see whether the previous maximum value of GZ, and of GZ beyond 30^o, have been exceeded

```
  #if heel > 29.9 & gz > maxgz30
    #Set maxgz30=gz
    #Set amaxgz30=heel
  #endif
  #if gz > maxgz
    #Set maxgz = gz
    #Set amaxgz = heel
  #endif
```

We save the GZ and heel values for later plotting

```
  #Set gzt[i]=gz
```

```
   #Set deg[i]=heel
#!
#! Check immersion points for the first
#! downflooding angle
#!
   #Set j=0
   #Foreach Immersionpoint
      #Set j=j+1
```

Z is initially the vertical coordinate of the immersion point:

```
      #Set Z=cosp*(Zimme*cosh-Yimme*sinh)-Ximme*sinp
```

Then it becomes the clearance of this point above the waterline

```
      #Set Z=Z-Waterl
```

A point has moved under the waterline if it was above at the last iteration, and is below now:

```
      #If Z < 0
         #If Zimm[j] > 0
#! Point has moved below waterline
```

Set the downflooding angle for this point

```
            #Set DFA=Heel-HeelInc*Z/(Z-Zimm[j])
            #Set Aimm[j]=DFA
```

If a downflooding angle has not yet been found, or a greater one has just been defined, set it now

```
            #If firstDFA > DFA
#! This is the first downflooding angle: Save it
               #Set firstDFA=DFA
#! Set heel to first downflooding angle, balance
#! the hull, get GZ and find total area to this
#! angle
```

The comment says it all ... or most of it ...

```
               #Set OldHeel=Heel
               #Set Heel=DFA
               #Put Heel
               #perform balance
               #Get GZ
               #Set sinh=sin(Heel)
               #Set cosh=cos(Heel)
```

Observe that we correct GZ to include the effects of all static loads:

```
               #Set LM=YMload*cosh+ZMload*sinh
               #Set GZx=GZ-LM/totdisp
               #Set chHeel=Heel-OldHeel+Heelinc
```

"DFarea' is the area to the first downflooding point

```
               #Set DFarea=a+0.5*chHeel*(lastGZx+GZx)
#!
#!   Restore Heel for completion of loop
#!
               #Set Heel=OldHeel
               #Put Heel
               #Set sinh=sin(Heel)
               #Set cosh=cos(Heel)
               #Perform Balance
            #Endif
         #Endif
      #Else
```

If the point is still clear of the water, preserve the height for reference next time

```
         #Set Zimm[j] = Z
      #Endif
```

#Next ImmersionPoint

Below, trapezoidal integration is used to integrate the area under the curve.

#Set a=a+0.5*(gzv+gz)*heelinc
#Set area[i]=a
#Set heel=heel+heelinc

Here we are at the end of the loop. It is exited when heel exceeds the maximum specified.

#until heel > heelmax

We need to know the index from which we can find values at 30 and 40^{O}.

#Set i30=30/heelinc+1
#Set i40=40/heelinc+1

Since the USL requirements are specified to two decimal places, we output the references values for this hull to two decimal places also.

#Decimal 2

The results are tabulated below. Note the formatting used to highlight non-compliant values. We could set the criterion values (e.g., 3.15 for the first) as variables, initialised using the "ask" command, or using the "#set" command at the start of the template.

USL Requirements **Minimum** **Actual**

#set fail=0
#if area[i30] > 3.15
Area under GZ curve between 0 and 30^{O} 3.15 `area[i30]`
#else
Area under GZ curve between 0 and 30^{O} 3.15 `area[i30]`
#set fail=1
#endif
#if area[i40] > 5.16
Area under GZ curve between 0 and 40^{O} 5.16 `area[i40]`
#else
Area under GZ curve between 0 and 40^{O} 5.16 `area[i40]`
#set fail=2
#endif
#Set da=area[i40]-area[i30]
#if da > 1.72
Area under GZ curve between 30 and 40^{O} 1.72 `da`
#else
#set fail=3
Area under GZ curve between 30 and 40^{O} 1.72 `da`
#endif
#if maxgz30 > 0.20
Maximum GZ at or greater than 30^{O} (metres) 0.20 `maxgz30`
#else
#set fail=4
Maximum GZ at or greater than 30^{O} (metres) 0.20 `maxgz30`
#endif
#if amaxgz > 25
Angle of heel at which max. GZ occurs (^{O}) 25 `%1 amaxgz`
#else
#set fail=5
Angle of heel at which max. GZ occurs (^{O}) 25 `%1 amaxgz`
#endif
#Set zm=zmeta-waterl
#if zm > 0.35
Initial metacentric height 0.35 `%2 zm`
#else
#set fail=6
Initial metacentric height 0.35 `%2 zm`

#endif

Compliance:

#if fail=0
Design meets requirements in this condition,

#else
Design does not meet requirements in this condition, because

#if fail=1
area under GZ curve between 0 and 30^{o} is less than 3.15.

#else if fail=2
area under GZ curve between 0 and 40^{o} is less than 5.16.
#else if fail=3
area under GZ curve between 30 and 40^{o} is less than 1.72.
#else if fail=4
maximum GZ at or greater than 30^{o} is less than 0.20 metres.
#else if fail=5
angle of heel at which max. GZ occurs is less than 25$^{o.}$
#else if fail=6
initial metacentric height is less than 0.35 metres.
#endif
#endif

General Information

Static trim in this condition (degrees) `pitch0`

We now plot the GZ curve:

GZ - Righting Lever Curve

The first line initialises graphics output
#Opengraphics

The next plots the graph
#Graph deg,gzt,N,"Heel",0,Heelmax,10,"GZ",0,0,0

Now we annotate the graph with the maximum GZ point. The "colour" command will only have effect if the Windows metafile option is used (otherwise, it will be ignored).
#Colour 7
#move amaxgz,maxgz
#draw 0,maxgz

Observe that text output must be positioned explicitly every time it is used.
#Set y = maxgz*1.03
#move 1,y
#plottext "Maximum GZ"
#move 15,y
#decimal 3
#plotvalue maxgz
#move 21,y
#plottext " at "
#decimal 0
#move 24,y
#plotvalue amaxgz
#move 27,y
#plottext " deg."
#closegraphics

....

`#next condition`

A page break is issued, before the processing of the next operating condition commences.

5.8. Generating a Report

Once all immersion points, static loads and tanks have been set up as required, you can generate a range of stability reports for all conditions. The form of report is determined by the template used.

You have been provided a set of templates, based on standard report format, and user experience since the first release of the program. On your distribution diskette (and installed along with Hullform+Hullstat) is a file directory Template, with subdirectories Txt and Rtf. They contain, respectively, templates for generation of equivalent reports in both text and RTF format.

Following is a brief summary of each template provided at the time of printing of this manual. Others may be added later.

5.8.1. Bending Moments

This produces a plot of bending moments for a hull. The template sets the displacement of the design to zero, before adding each defined load to the hull. The distribution of masses and positions of the static loads therefore defines the distribution of mass of the hull, allowing the distribution of buoyancy found from Hullform to be combined to give the bending moment curve along the hull. The result is tabulated and the hull's immersed areas and bending moments are plotted.

All inputs are obtained from the project's defined static loads, so no user intervention is required prior to processing.

5.8.2. Deadweight table, short form

This is a summary of the hydrostatic contributions of all static loads and tanks, for all operating conditions. The hull is free-trimmed at zero heel for these calculations.

The output includes a deadweight table, USL analysis, and a GZ curve.

5.8.3. Deadweight and Stability

This corresponds to the output of templates described in sections 5.8.1 and 5.8.9, combined. The output includes a deadweight table and a GZ curve.

5.8.4. GZ table, horizontal format

This tabulates GZ values for a range of heel and displacement, both controlled by the user. Inputs you must provide comprise fine heel increment, fine heel increment end and coarse heel increment, as well as start and end displacements, and the displacement increment.

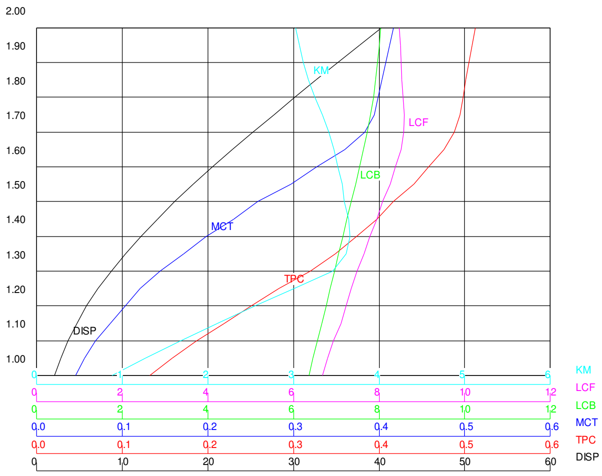
The table in its default form provides for 11 columns, each corresponding to one heel angle. In its default configuration, these angles are 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, and 90^o. These correspond to a fine heel increment of 5^o to an end angle of 20^o, then a coarse increment of 10^o to 90^o. If (for example) you wanted to display values at finer increments, you could select a fine heel increment of 2^o to an end at 10^o, then a coarse increment of 5^o. The 11 columns would then extend to 40^o.

The generated table fits across an A4 page. Upright hull draft is also shown in the table.

5.8.5. Hydrostatic table, level trim, basic

All calculations are performed at zero pitch and heel. Inputs required are the start and end draft, and the draft increment. Displacement, displacement per unit change of draft, moment to change trim, centre of buoyancy and flotation, and righting lever arms are tabulated and plotted.

There also exists an enhanced version of this template, giving a combined colour plot (See the following plot - colours lacking, of course). This template was prepared and paid for by a Hullstat user, but is available from Blue Peter Marine Systems for a small commission charge, if required.



5.8.6. Immersed areas

This outputs a tabulation and plot of the immersed areas for the hull, at hydrostatic balance.

5.8.7. KN table, horizontal format

This produces a table of the righting lever KN, at user-defined heel increments and over a user-defined displacement range.

5.8.8. Righting moment table, horizontal format

This produces a table of the righting moment, at user-defined heel increments and over a user-defined displacement range.

5.8.9. Stability table, vertical format

Analyses of stability characteristics of the hull, using USL requirements. GZ versus heel cross-curves are also plotted. Statements of attainment (or non-attainment) of the USL guidelines are generated automatically.

5.8.10. Tank statics tabulation

This template generates a tabulation (at 5% intervals) of hydrostatic parameters for all tanks in the hull.

5.8.11. Towing lever and plot

This produces a combination of outputs analysing the stability of a tug when towing. It includes a plot of tow rope lever on the hull's righting lever curve, and reports parameters (like areas under the tow level curve) needed by regulatory authorities. When you run this template, you will be asked to provide towing parameters (bollard height, attachment radius and bollard pull).

6. Post-Processing the Report

A Hullstat report should require little editing. If significant routine editing is required, it is likely that enhancements to the report template are needed. For example, the HSL “if” structure allows insertion of cautionary text when any parameter is out of an allowed range.

However, if your word processor is not capable of accepting the RTF file format, some reformatting of the text output will be needed. This could include arranging tabular data into proper columnar format. Insertion or conversion of graphical output may also be possible. To obtain any assistance with these matters, you will need to consult your word processor's help system, or manual.

If you have used RTF output, there is one limitation of the RTF format which your manual is unlikely to mention. To see if you are likely to hit this problem, try producing a simple RTF file, entering a few lines of text and using your word processor's options under “File, Save As...". Then read it in again, and select the “File, Print Setup” menu item. When you select some printers, the text displayed will default back to a basic font like Courier.

If this occurs, you have a simple solution - never print an RTF file. Instead, follow the recommendations of section 1.2, and include the RTF file into another document. Documents saved in the word processor's native format include information which ensures that the font-loss problem does not occur.

7. Configuring Hullstat - User Preferences

There is a limited number of configurable parameters within Hullstat, as follows:

7.1. Progress Display

You can select the amount of information shown in the display window at the bottom of the Hullstat dialog box.

Selection of the first (normal default) option, “None", means that the only data displayed is that explicitly output using the Hullstat “print” command.

The second option, “loops", provides some extra tracking, showing all looping statements executed.

The third, “enacted commands", shows all HSL commands as they are executed, while the fourth, “all commands", also shows commands skipped (e.g., those in blocks of code skipped because they correspond to a non-satisfied “if” command). Selection of either of these options will slow template processing markedly, but will help if changes to a template result in an error message.

7.2. Use RTF Format for Templates and Output

This setting helps Hullstat identify the format of the files your want to use. If this checkbox is set, the program will look for files with extension RTF when searching for templates and output files. If not, it will look for extension TXT.

The validity of an RTF template file is tested when it is first read. If the output file proves to be invalid, or is generated as a text file only, check the RTF setting - if it has been un-set in the processing of the template, it is likely that the template file has been saved in the wrong format.

7.3. Graphic Format

Hullstat can generate output in either Hewlett Packard Graphics Language ("HPGL“) or Windows metafile format. The former is widely accepted as an image format, by word processors and CAD systems - as well as by many plotters. Windows metafiles are less widely accepted, but allow better image control - particularly line and font handling.

7.4. Text Data Size

The text component of each variable name has a default size of 40 characters, but this may be extended (to a limit of 128 characters) if longer text information strings are preferred.

Even with the maximum size of 128 characters, the 400 variable limit to the symbol table means only about 50K of memory is used. This is not a significant demand on most modern systems.

7.5. Symbol Table Size

While the 400 variable limit is more than adequate for templates developed so far, really big ones may reach this limit. The symbol table can be expanded to any size required, by changing the value in this edit box.

7.6. Project Directory

Hullstat project files are the initial information source, which identify hull configurations and file organisation for report output. The project directory is defined here (external to any project) for this reason.

8. Getting Help

If this manual or the Hullstat help system does not show you how to do a particular task, you should get in touch with Blue Peter Marine Systems. You can do this by mail (92 Dyson St, Kensington, W.A. 6151, Australia), by fax (+61 9 474 1288) or by e-mail (support@www.hullform.com).

If you do not find a template which covers your needs, remember that you are entitled (workload at Blue Peter Marine Systems permitting) to one free ustomised template, as part of your purchase. The sole conditions are:

1. You must provide, on a diskette or via e-mail, a text or RTF formatted document in the format of your required template. The hydrostatic values required at each point of the template must be clearly evident.
2. The layouts of tables and graphs are not guaranteed to be identical, being governed by page limits and the existing format of Hullstat graphs.
3. Parameters beyond the existing list available from Hullstat will not be included in calculations or reports.
5. You must agree to the template's being made available to other Hullstat buyers.
5. Provision of a sample report, and the design on which it was based, will assist checking of the report's validity.
6. The provided template is equivalent to a new item of software, and will be checked as far as possible before being transmitted. No guarantee can be given that it will work as transmitted for all designs, but a three-month warranty - during which time any reported faults will be traced with top priority - is included in the offer.

Index