GODZILLA 9.33

Examples and Templates

GODZILLA IS RESEARCH CODE!

Please check all estimates generated by the program against experimental results before committing any time or funds to your project as no liability can be accepted by Cyberiad.

Contents

1 INTRODUCTION

GODZILLA example input files are in the godzilla directory. These files are described below and should be used as templates for your own designs.

To use the example files, you must save the indicated example file as in.mlt in the main Michlet directory (i.e. the directory that contains the executable file). This will overwrite the current in.mlt with the example input file.

It is probably a good idea to go through these examples with the appropriate file loaded into your text editor, or to have a printout handy.

Note that the number of θ-intervals, N_θ, used in the calculation of the wave resistance and wave patterns has sometimes been set to small values in these examples. For serious design work, these should be increased to the values recommended in the Michlet manual. Better still, before embarking on a large design project, you should calculate resistances using different values of N_θ to see how many are required for reasonable accuracy. Similar convergence tests should also be performed with the number of hull stations and waterlines.

Also note that some plots will not be displayed until after the population has been initialised.

2 MONOHULLS

2.1 gin1.mlt: Objective Functions

The shape of a hull found during the optimisation process depends on what we are trying to minimise. In this example we will look at several objective functions, namely:

- 0 : Minimum surface area
- 1 : Minimum viscous resistance
- 2 : Minimum wave resistance
- 3 : Minimum total resistance

The first part of the gin1.mlt file contains the dimensions and offsets for a one tonne hull. The offsets in the table are for a Wigley standard parabolic hull, however for the present examples they can be considered as dummy arguments. Note that we have used the parametric form for the offset table. Because the offsets in this section are just dummy arguments, there is no need to use a table of offsets, although we could if we wanted to, but the input file would be longer and difficult to print out.

The length has been set to 10.0m and the draft to 0.25m. The first hull in the population will have these dimensions, as long as the dimensions do not violate the constraints specified later in the input file.

The default random seed is 1. For real design problems, the program should be run multiple times, each time varying only the value of the random seed. If the optimum vessels found after many runs are very similar, then we can claim with a little more confidence that we are close to the (global) optimum solution. Of course, there is no absolute guarantee. It must also be stressed that if this random seed is not changed, and if GODZILLA is run again with the same input file, exactly the same sequence of hulls will be produced. This can be either a blessing or a curse. A boon if you want to reproduce a particular run, or a real nuisance if you remember, six hours into a run, that you forgot to change the seed and that you are going to get exactly the same answer as last time.

The number of ships in the population has been set to 20, which is a reasonable value for a small design problem. For larger problems, i.e. when there are many parameters participating in the optimisation, a larger population will be slower, but less prone to becoming trapped in local minima.

The default objective for this example is 0, i.e. minimising the total surface area. The objective variant is equal to 0.

The design speed is $U = 6.0\text{ms}^{-1}$.

The hull shape function is 1. This function takes three shape parameters and is described elsewhere in the manual. In brief, the first parameter controls the shape of the waterlines, the second controls cross-section shape, and the third determines the shape of the sideviews. A value close to 0.0 means the shape is roughly rectangular, a value of 0.5 indicates an elliptical shape, while values close to 1.0 are roughly parabolic.

The next five parameters limit the overall displacement , overall length, and overall width of the vessel. For monohull design problems, these constraints can be set to cover a very wide range of values. In that case, the constraints for the individual hull (which are to be specified a little further on in the input file) will control the size of the vessel.

The constraints on ship motions, pressure signatures, and other special constraints are not used in this version of Michlet. Leave them set to 0.

The minimum and maximum hull displacement are both set to 1.0 which means that the displacement will not vary during the optimisation. It is rarely wise to allow displacement to vary during the optimisation of monohulls. In most cases, the minimum resistance will occur for the hull with the smallest allowable displacement. This is not necessarily so for multihulls, where we might want to find the optimum size of outriggers.

For the present example, length and draft are allowed to vary over a (deliberately) very wide range for a one tonne vessel.

The ranges for the hull shape parameters are specified next. For the hull shape function we are using in this example, we need to specify constraints on three input parameters and to also specify the seed value for each shape parameter. This is done by supplying one row of three comma-separated values for each shape parameter. In this example, we allow each parameter to take values between 0.0 and 1.0. The seed value for each parameter is 1.0.

Following the shape factor section of the input file, there is a list of constraints on a variety of geometric quantities. The ranges for these various parameters have been made very large for the present examples.

2.1.1 Minimising surface area

- Run godzilla.bat

- The hull defined by the offset table in the input file should be displayed. When GODZILLA is initialised, this hull will be replaced on the screen by the best hull in the population.

- The small panel at the bottom right of the screen should indicate that GODZILLA is UNINITIALISED. The objective function should be Min Area. The number of parameters should be equal to 5 for this example; we are trying to find one length, one draft, and three shape parameters. The number of Evaluations is zero, and the Best Fitness found so far is a huge number. The Shape Function is 1. The three seed shape factors should be the same as specified in the input file.

- Press p to display the population fitnesses. Before GODZILLA is initialised, the fitness of all individuals in the population will be a ridiculously large number indicating a very poor fitness.

- Press g to initialise GODZILLA and create one new generation. When the message GODZILLA IDLE appears in the panel at the bottom right of the screen, you can look at various aspects of the hull, assess its drag for the range of speeds specified in the in.mlt file, calculate the wave pattern etc. Some ships in the population may occasionally be flagged as Violated. This means that the ship violated one or more of the constraints. This is not necessarily a bad thing, and in some circumstances these hulls can help to guide the search near constraint boundaries. Wave resistance is not calculated for hulls that are flagged as violated, so that their appearance does not slow down the optimisation too much.

- To resume optimisation again, press g again. The message GODZILLA STEPPING will appear briefly, and then the GODZILLA IDLE message will again be displayed.

- Now press G to let GODZILLA run continuously. The message GODZILLA RUNNING will appear in the bottom right panel. As the optimisation progresses, various hull shapes will appear on the screen (if you pressed G from one of the hull views) or, if you restarted the optimisation from the population fitness view, the fitnesses of the vessels will change. To stop GODZILLA, press a key (the space bar is best) once, and wait for the idle message to appear.

- To finish this example, press the Esc key to return to the title screen, then press Esc to exit Michlet. You will be asked to confirm that you indeed do want to exit. This screen will also contain a reminder when resistances have not been calculated for the last vessel displayed in the Lines Drawing Screen.

- In a serious design problem, you should now save output files to a safe directory to prevent them being over-written the next time you run Michlet.

For the next part of this example, run the program as before and initialise the population by pressing g from the Lines Drawing screen.

There will be times during an optimisation run when you might want to enter your own guess for the best hull. For example, for the present problem we know the exact answer; to at least 5 figures, the (mathematical) optimum for this problem is a hemisphere of length L = 1.563185m and draft T = 0.781593m. The three shape factors are all equal to 0.5 (i.e. elliptical in all views).

- Press l and enter 1.563185 for the length

- Press t and enter 0.781593 for the draft

- Press n to begin entering new shape factors. (Don't press the Enter key.)

- Enter 0.5 for each of the shape factors. The on-screen cross-sections should now be ellipses. It is important to note that the hull displayed on-screen has not yet been included in the population of ships. If you now pressed g or G, the currently displayed hull would be replaced with the best in the population and your seed vessel would be lost. To include the vessel in the population, first press S (that's uppercase). The fitness of this seed vessel is calculated, and the worst individual in the population is replaced. If the seed vessel is the best in the population, it will remain on-screen. If not, it will be replaced by the best vessel. It is possible, by pressing S repeatedly, to replace all hulls in the population with the best individual. This will ensure that many individuals will search in the vicinity of the best ship, but it does not guarantee that a global optimum will be found more quickly.

- The seed hull that you just entered should have a fitness (i.e. surface area) of 3.87901m^2; the exact value (to 6 places) is $S = 3.838317\text{m}^2$, about 1% lower. If we used more offsets to represent the hull surface, we would get better approximations at the expense of longer computer runs and greater demands on computer memory. On the other hand, Michlet is a program designed to deal with thin ships. The present example of a hemisphere is a very severe test and not necessarily a good indication of the program's ability with thin hulls.

- Press G to let the GODZILLA continue the search. Note that the program finds hulls that have a lower surface area than the exact area. Whether the accuracy is acceptable depends on the specific design problem.

- Press the Space Bar to stop the optimisation, then press Esc to exit.

Sometimes global optima will occur in a very wide shallow basin, and quite large variations in one or more of the parameters will not seem to produce any noticeable effect on the fitness. Other times, small changes in for example, design speed, can lead to large changes in the optimal values of the hull parameters.

2.1.2 Minimising viscous resistance

For this example, change the objective function from 0 to 1 in the in.mlt file and save the file. (There is no need to change the objective variant.) Then run godzilla.bat.

Since viscous drag depends on surface area, we expect that the optimum hull will be similar to the hull of minimum surface area found in the previous example. (It won't be exactly the same because of Reynolds number effects).

2.1.3 Minimising wave resistance

For this example, change the value of the objective function to 2 in the in.mlt file and save the file.

With this objective and with the chosen constraints, the hulls can be seen to get very long and very deep, eventually becoming very thin plates. This objective function is not particularly useful for relatively unconstrained monohull designs; its real value will be seen in the multihull examples where we use it to optimise the placement of hulls with respect to each other.

2.1.4 Minimising total resistance

To run this example, change the value of the objective function to 3 in the in.mlt file and save.

As the optimisation progresses, the best hulls tend towards fine canoe bodies. The best fitness (in this case total resistance) I found was about 0.44kN.

2.2 gkay.mlt: Small Sprint Kayak

For this example, save the gkay.mlt file (which is in the godzilla directory) as in.mlt in the main Michlet directory.

In the introductory examples, many of the hull dimensions were not particularly tightly constrained. As an example of a slightly more realistic problem, we will try to minimise the total resistance of a small sprint kayak of displacement volume $D = 0.098\text{m}^3$ and for a design speed of $U = 5.0\text{ms}^{-1}$.

The length of the kayak is required to lie between 4.0m and 5.2m, the draft must be between 0.01m and 0.3m, and the beam must be between 0.35m and 0.75m. The maximum value of the beam could be set to a much larger value, but there are good reasons to keep it relatively small. During the optimisation, wave resistance is not calculated for hulls that have violated any constraints. Since this is the most time-consuming part of the calculations, we do not want to calculate wave drag for hulls that are unlikely to be competitive. In this case, however, even a beam of 0.75m is probably too wide to be practical.

The hull shape function (number 7) for this example is the seven parameter function described in the Michlet manual. This hull series contains fore and aft asymmetric hulls, and hulls with parallel middle bodies. The minimum value for all seven shape factors is 0.0. The maximum value for the first 5 parameters is 3.0, and for the last two parameters, the maximum value is 0.75. The first five shape parameters define the shapes of the forebody and afterbody. A maximum of 3.0 for these parameters means that we allow cusped hulls. The final two shape parameters determine the relative lengths of the forebody and afterbody. The length of the forebody will be allowed to take any value between 0.0 and 0.75; i.e. the forebody can comprise up to 75% of the total hull length. The length of the afterbody can also vary between 0.0 and 0.75 of the length of the hull with the following proviso. If the sum of the last two parameters exceeds 1.0, then the length of the afterbody is reduced so that the sum of the two parameters is equal to 1.0. In other words, the forebody and afterbody together cannot comprise more than the total hull length; if the forebody comprises 75% of the hull length, the afterbody fraction will not be greater than 25%. If the sum of the two last parameters is less than 1.0, then the remaining fraction of the hull length is parallel middle body.

The LCB of the vessel is required to lie between 0.14m aft of centre and 0.025m aft of centre (i.e. between -0.14 and -0.025). The VCB is required to lie between 0.065m and 0.3m; VCB is not of great interest here, and nor is the location of the LCF which is allowed to take a very wide range of values. The values and limits of these quantities have been chosen for the sake of example only - they are not necessarily the best choices for real kayaks.

- Run godzilla.bat and change to the population fitness screen. The fitness of all individuals will be some large number before initialisation.

- Press g to initialise the population. It is quite possible that most, if not all, individuals in the population will be flagged as violating one or more constraints. This is nothing to worry about at this stage. Since random values are assigned to the hull parameters, it sometimes takes a while to find a hull that satisfies all constraints.

- Press G for continuous operation.

- To interrupt the optimisation, press the space bar and wait for the GODZILLA IDLE message.

If the final hull is of interest, calculate the resistance and then exit from Michlet. Copy the output files to a safe directory before running Michlet again.

2.3 gkaybox.mlt: Sprint Kayak with Box Constraint

In the previous kayak example, we found the major dimensions and shape of a small sprint kayak. Suppose now that we want the hull to also accommodate a rectangular box of dimensions $L_{box} = 1.2$m, $T_{box} = 0.04$m, and $B_{box} = 0.34$m, and that the centre of this box be located $X_{box} = 2.6$m from the bow, and at a vertical distance $Z_{box} = 0.02$m down from the waterplane. The draft of the box and the vertical location of the box centre have been chosen so that the top of the box lies in the waterplane.

The offsets for the default hull are defined using Hull Series 7.

There are many other optimisation runs that could be attempted. What hulls would be found if the LCB limits were relaxed? How does the inclusion of a form factor in the calculation of the total resistance affect the hull shape? What effect do different water densities and viscosities have? What if we optimised over a range of speeds?

If hollows in the hull shape are prohibited by competition rules, the optimisation should be done using limits on some of the shape parameters. Specifically, for the hullform family used in this example, Hull Series 7, the maximum for the first 5 shape factors should be set to 1.0. This will guarantee that the hull does not have hollows in the waterlines or sections.

2.4 gs20.mlt: Hull Series 20 with Shape Constraints

This example shows how to use hull series 20 and, more importantly, how to constrain the shapes of the waterlines, sections and the keelline, to prevent wild, impractical oscillations in the hull lines. The objective is to minimise the total resistance of a one tonne monohull at 5.6ms^{-1}, however the actual details of the objective are unimportant.

2.4.1 Unconstrained

First save the gs20.mlt file from the godzilla directory as in.mlt in the main Michlet directory. The last two lines of the input file should read:
Special Hull Constraints (See manual and GODZILLA examples.)
-1,0,0,0
Box constraints (None for this example)
0

These input lines indicate that no special constraints are made on the waterlines, sections and keelline (apart from those imposed inherently by Series 20), and that no box constraints are imposed. Run the program and note the unusual lines. Change to the horizontal view and note the oscillations in the waterlines.

Start the optimisation, and note how the best hull becomes smoother, but that it has some unusual bumps near the baseline. When you have seen enough of the process, stop the program and exit.

2.4.2 Constrained waterlines

Load file in.mlt into your text editor and change the special hull constraints line to read:
-1,1,0,0

This will constrain the shape of the waterlines by restricting the number of slope sign changes. The sections and keelline remain unconstrained (within the allowable variations of the hull series). Run GODZILLA, change to the horizontal view, and start the optimisation.

As the search progresses, the best hull will have some hollows in the waterlines, but the wild oscillations evident in the previous example are not present. Stop the search and exit.

2.4.3 Convex waterlines

Edit in.mlt so that the special hull constraints line reads:
-1,2,0,0

This constraint requires that waterlines are to be convex.

Run GODZILLA, change to the horizontal view, and start the optimisation. It could take a quite a while to find a hull that doesn't violate all constraints which is understandable given that we are dealing with a 22 parameter problem. Stop the program and exit.

2.4.4 Combining special shape constraints

Edit in.mlt so that the special hull constraints line reads:
-1,2,0,1

This means that waterlines will be convex, and the keelline shape is constrained so that there is, at most, only one slope sign change.

Run GODZILLA and start the optimisation. The hulls found during this run will probably be similar to those found during the previous example.

2.4.5 Special shape constraints and shape parameter constraints

In the previous examples we saw how to implement special constraints on the shapes of waterlines and the keelline. The section shapes were not constrained because they are already convex for hull series 20.

There are many other interesting hull shapes that can be found using Series 20 in combination with limits on the minimum and maximum values for the various shape parameters. For example, the last five lines of the set of constraints beginning with Shape Factors: Minimum, Maximum, Seed are:
0.0, 1.0, 0.5
0.0, 1.0, 0.0
0.0, 1.0, 0.0
0.0, 1.0, 0.0
0.0, 1.0, 0.0

Change the first of those five lines to:
0.5, 0.5, 0.5
and the last line to:
0.5, 0.5, 0.5

The set of shape constraints in the input file should then read:
Shape Factors: Minimum, Maximum, Seed.
0.1, 0.5, 0.25
0.1, 0.5, 0.25
0.1, 0.5, 0.25
0.0, 1.0, 0.1
0.0, 1.0, 0.5
0.1, 1.0, 1.0
0.1, 1.0, 0.1
0.0, 0.5, 0.0
0.0, 1.0, 0.1
0.0, 1.0, 0.5
0.0, 1.0, 1.0
0.0, 0.5, 0.1
0.0, 1.0, 0.5
0.0, 1.0, 0.5
0.0, 1.0, 0.5
0.5, 0.5, 0.5
0.0, 1.0, 0.0
0.0, 1.0, 0.0
0.0, 1.0, 0.0
0.0, 0.0, 0.0

The first line shown in bold forces the transom section to be elliptical in shape, i.e. the shape factor is 0.5 for all candidate hulls created during the search process. The last line in bold forces the sternmost hull section to be perpendicular at the waterplane. That hull section is not raised out of the water and so it is not flared at the waterplane. Other sections can, however, be flared.

Run GODZILLA and start the optimisation. When I ran the program it took about 7,000 evaluations before a viable hull was found, but that was because I also used the hull shape constraint line unchanged from the previous example, i.e. my shape constraint line was:
-1,2,0,1

A viable hull could, no doubt, be found if we changed that line to:
-1,0,0,0
so that no special constraints were imposed, but the hulls might not be very practical.

2.5 gs32.mlt: Hull Series 32

This example shows how to use hull series 32 in an optimisation context. The objective is to minimise the total resistance of a one tonne monohull at 5.0ms^{-1}.

We demand that the longitudinal metacentric height of the vessel should be greater than 120m by using the lines
Minimum GML Overall
120.0
Maximum GML Overall
10000.0
in the Ship Constraints section of the input file.

Although there are many shape parameters to consider, the general principles are exactly the same as for the g20.mlt example; each shape parameter must be initialised, and constraints must be imposed.

2.6 gs42.mlt: Hull Series 42

This example shows how to use hull series 42 in an optimisation context. Similar to the gs32.mlt example, the objective is to minimise the total resistance of a one tonne monohull at 5.0ms^{-1}.

It is interesting to pause Godzilla every so often and look at the waterplane view. Hull waterlines can have some very unusual bumps. When I ran this example, it took about 51,000 evaluations before the waterlines became convex.

Note that a similar shape hull could be achieved by using fewer shape parameters. A worthwhile exercise would be to try the same optimisation using series 7, 9, 20, and 32.

2.7 gapproxm1x.mlt: Approximating real hulls with mathematical series

Sometimes it is very useful to have a reasonable mathematical approximation of real hulls. In this example we will attempt to find a reasonable approximation of the lightweight men's rowing shell described in Michlet example file m1x.mlt.

Objective function 100 attempts to achieve this by matching the section areas and/or waterplane areas of the real hull and the approximation. Variant 0 is used to match section areas, variant 1 matches waterplane areas, and variant 2 matches both.

Users must also supply the section areas and/or waterplane areas. If you run m1x.mlt (after copying to in.mlt in the michlet directory) you will see that the section areas and waterplane areas are saved as, respectively, hsecarea.mlt and hwparea.mlt. The first column in each file is the non-dimensional distance from the bow or, for the waterplane areas, from the keelline. The values in the second column can be pasted into the Godzilla section of the input file after the lines specifying Maximum B/T. It is important that the number of section areas is identical to the number of sections in the approximating hull and, similarly, that the number of waterplane areas is the same as the number of waterlines.

Copy the gapproxm1x.mlt file to in.mlt in the michlet directory. At the end of that file, you will see how the section areas have been placed in the file. Also note how the constraints have been set up in the input file. In this example, we choose to keep several hull parameters the same, and we allow others to vary between acceptable limits as shown in Table 1.

Table 1: Target values, limits, and values after 500,000 evaluations.

Parameter	Target	Low	High	Found
D	0.085	0.085	0.085	0.085
L	7.42	7.42	7.42	7.42
T	0.095	0.095	0.095	0.095
WOA, B	0.226009	0.22	0.24	0.239648
$GMTOA$, GMT	0.015816	0.015	0.025	0.021327
$GMLOA$	52.948460	52.5	53.5	52.504990
S	1.952715	1.9	2.0	1.946648
Awp	1.321113	1.3	1.4	1.357543
LCB	-0.353012	-0.36	-0.30	-0.300001
VCB	0.059285	0.05	0.07	0.059149
LCF	-0.426114	-0.43	-0.33	-0.330152
BMT	0.051531	0.0	99999.9	0.057178
At	0.005948	0.0	0.006	0.003455
Bt	0.159812	0.10	0.16	0.114016
Tt	0.044531	0.04	0.05	0.041563
Cb	0.533538	0.50	0.55	0.503174
Cm	0.784763	0.72	0.80	0.751000
Cp	0.679872	0.67	0.70	0.670005
Cwp	0.787790	0.76	0.79	0.763442
L/B	32.830557	30.0	33.0	30.962100
L/T	78.105263	75.0	80.0	78.105263
B/T	2.379042	2.3	2.6	2.522609
Shape Parameter				
f_0	n/a	0.8	1.0	0.941067
f_1	n/a	0.2	1.0	0.508199
f_2	n/a	0.2	1.0	0.396614
f_3	n/a	0.2	1.0	0.732045
f_4	n/a	0.2	1.0	0.663387
f_5	n/a	0.3	0.7	0.516986
f_6	n/a	0.3	0.7	0.425082
f_7	n/a	0.0	0.5	0.380424
f_8	n/a	0.0	0.5	0.158534

Now run Godzilla as usual. The program screen will change very rapidly because no hydrodynamic calculations are needed for this objective function. After 500,000 evaluations (about 4 minutes on an IBM i7 PC), I found the values in the right-most column of Table 1.

On exit from Michlet, the difference between the current areas and the target areas are written to the out.mlt file. (Look towards the end of out.mlt, after the Hull Constraints section).

It is important to note that Objective 100 is not guaranteed to always find a reasonable approximation. It will do well on some reasonably smooth hulls, like rowing shells and canoes, but it is unlikely to do well with, for example, SWATH-like hulls, bulbous bows, or hulls with large transom sterns. Using a hull series with a larger number of parameters (e.g. series 32 or 42) might help, but it will take a lot longer.

The shape parameters are the most important quantities found using the technique in this example. It would be a very worthwhile exercise to make a copy of the m1x.mlt file and to replace the offset table with

9, 0.941067, 0.508199, 0.396614, 0.732045, 0.663387, 0.516986, 0.425082, 0.380424, 0.158534

Is the drag of the mathematical approximation similar to the drag of the original M1x hull?

Another useful exercise would be to repeat the present example with Series 20, and to change the limits on many of the parameters.

2.8 gs4.mlt: Hull Series 4

This example shows how to use hull series 4 in an optimisation context.

The objective is to minimise the total resistance of the Collins-like Submarine Body described in s4collins.mlt in the Michlet Exampe manual.

The design speed is 10.289ms^{-1} (20 knots) which is very high for the submergence depth of 7.8m.

The constraints on the beam to draft ratio are very tight ($0.99 \leq B/T \leq 1.01$) in order to guarantee near circular cross-sections during the search.

I found an optimum length of 87.3m and a beam (i.e. diameter) of about 8.45m.

3 DIHULLS

3.1 gcat.mlt: Standard Catamaran

In this example, the displacement volume of the demihulls is $D = 1.0\text{m}^3$, the length of the demihulls is $L = 11.537\text{m}$, and the draft is $T = 0.254\text{m}$.

This example uses objective function 2, i.e. we will minimise the wave resistance. The hull shape function has been set to 1 which means that we will use a mathematical hull series. However, later in the file we constrain the shape parameters so that they cannot change during the course of the optimisation.

The only parameter that can change in this example is the lateral hull separation, w, which is required to lie between 2.0 m and 15.0 m inclusive.

My optimum was about $w = 5.5\text{m}$.

For this 1-parameter optimisation we could manually plot the wave drag for various demihull spacings and find the best separation. A more difficult problem would be to optimise for total drag (change the objective function to 3) and to also search for the best shapes from, say, the Series 1 hullform family (change the hull shape function to 1). Remember to allow the shape parameters for both hulls to vary during the optimisation! This will be a 7-parameter optimisation (1 lateral separation distance and 3 shape parameters for each demihull). This can be reduced to a 4-parameter problem (1 lateral separation distance and 3 shape parameters) using the Special Ship Constraint option to force the demihulls to have identical shapes and dimensions.

3.2 gdiwave.mlt: Wave Minimisation

There are three main ways to minimise wave amplitudes with GODZILLA. Minimising the wave resistance is the simplest. The expectation is that a vessel with minimum wave drag will produce waves of small amplitude. The second way is to minimise waves along a specified line behind the ship, e.g. a longitudinal cut. The third way is to minimise the peaks in the free wave spectrum for specified angles of wave propagation. This last method is particularly useful when we wish to minimise waves of particular frequencies. High-speed vessels often create large diverging waves, (those travelling outwards from the vessel ($\theta > 35.267°$), and these can slam moored boats against walls, or cause severe damage to beaches.

For this example, save the gdiwave.mlt file as in.mlt in the main Michlet directory.

The size and shape of demihulls for the vessel are the same as those in the gcat.mlt file, however only 21 stations and 11 waterlines are used in order to speed up execution.

The maximum speed is 6.0 ms^{-1}, the same as the design speed. In GODZILLA mode, the value for the speed in the Wave Analysis menu cannot be changed while the program is running; it is always the maximum speed defined near the start of the input file. Using the same value for the maximum speed and the design speed allows us to look at intermediate results. If a different maximum speed had been specified, we would only be able to see intermediate graphs for that speed and not the design speed.

To illustrate some principles of wave minimisation, the location of Hull 2 is allowed to vary both laterally and longitudinally. Of course, the resultant asymmetric arrangement is not necessarily going to be a practical solution to a real problem, but the methodology is the same when vessels are required to have lateral symmetry.

The objective function has been set to 2, i.e. minimum wave resistance.

3.2.1 Minimum wave resistance

Run godzilla.bat. From the Lines Drawing screen, press 2 to display details of the second hull.

Now press G to initialise and run GODZILLA. After about 600 evaluations, pause GODZILLA by pressing the space bar. My best vessel had a longitudinal stagger of about s = 7.2m, and a lateral stagger of about w = 0.8m. When the GODZILLA IDLE message appears, press W to display the Wave Analysis menu. Note that none of the parameters in this menu can be changed in GODZILLA mode.

Press L to show wave elevations along the longitudinal cut. The minimum wave amplitude is about -0.023m, the maximum is about 0.02m.

Press Esc to return to the Wave Analysis menu, then press F to show the free wave spectrum. For me, the largest peaks in the spectrum were at about $\theta = -80°$ and $\theta = 80°$. Negative θ means waves propagating to starboard, positive θ is for waves on the port side of the vessel. Waves travelling at $\theta = 0$ are those following the vessel, i.e. pure transverse waves. Waves travelling for θ near $\pm 90°$ are those high frequency diverging waves travelling outwards and more parallel to the ship's track.

You can see the wave pattern for the vessel by pressing R from the Wave Analysis menu. For the $301 \times 301 = 90601$ points in the rectangular patch, this takes about 1 minute to calculate on a 2GHz PC. Note that the wave pattern is not truescale: the rectangular patch is 200m long but only 100m wide. Also note the 'streaking' of the wave pattern. This is caused by using too small a value for N_θ. Increasing N_θ to 1024 will eliminate the streaking at the expense of a longer execution time.

3.2.2 Minimising waves along a longitudinal cut.

Change the objective function to 6 in the GODZILLA section of the in.mlt. Also add a line containing the three (comma-separated) values 50.0, 50.0, 250.0 after the variant. Your input file should now look like the following:

```
# Objective Function (integer: see Manual)
6
# Objective Function Variant (integer: see Manual)
0
50.0, 50.0, 250.0
```

Objective function 6 means that we will minimise waves along a longitudinal cut. The line after the objective function means that we will search from $x_1 = 50.0$m and $y_1 = 50.0$m to $x_2 = 250.0$m. The positive y-value means that the longitudinal cut is on the starboard side of the vessel. The number of points along the cut that will be tested during the optimisation is specified in the definition of the rectangular patch earlier in the in.mlt file. In this example, we use 301 points. Therefore, we will search for a hull arrangement that minimises the wave elevations at 301 points from 50.0m aft of the centre of Hull 1 to $x = 250.0$m.

Run GODZILLA and pause the optimisation after about 600 function evaluations. The optimisation proceeds somewhat more slowly than when we were using minimum wave resistance as an objective function. The process could be sped up by using fewer points along the cut, but then we might miss some large peaks or troughs. We could also use a smaller value of N_θ, but then the accuracy of the predicted wave heights might be compromised.

My best ship had $s = 9.3$m, and $w = 1.6$m for the lateral and longitudinal spacings, respectively. The minimum wave amplitude was $z = -0.01$m and the maximum was 0.01m. Of course, we might be able to do better if the program was run for longer.

3.2.3 Minimising high frequency waves

Change the objective function in the input file to 4, then add the following line after the objective variant:
-90.0,-45.0

Your input file should now look like the following:

```
# Objective Function (integer: see Manual)
4
# Objective Function Variant (integer: see Manual)
0
-90.0, -45.0
```

We will now try to minimise the energy shed for wave propagation angles from $\theta = -90.0$ to $\theta = -45.0$. (Note that GODZILLA requires that the first angle be less than the second angle.) Waves travelling between these angles of propagation are high frequency diverging waves shed to starboard.

Run GODZILLA for about 600 function evaluations and then pause the optimisation. My best vessel had $s = 8.39$m and $w = 1.21$m. Examination of the free wave spectrum shows that the largest peaks occur for positive values of θ.

Stop execution, exit, and then change the line after the objective variant to read:
45.0, 90.0

Using positive angles means that we will minimise waves shed on the port side of the ship.

Run GODZILLA for about 3500 function evaluations and then pause the optimisation. My best vessel had $s = -8.1$m and $w = 1.1$m, i.e. Hull 2 is now forward of Hull 1, whereas when we were optimising for negative θ, Hull 1 was forward of Hull 2. Examination of the free wave spectrum shows that the largest peaks now occur for negative values of θ, i.e. the largest waves are now shed to starboard.

3.3 gcatsym.mlt: Using Special Ship Constraints

In this example, the displacement volume of the demihulls is $D = 0.75\text{m}^3$, the lengths of the demihulls are allowed to vary between $L = 1.0$m and 10.0m; the draft is allowed to vary between $T = 0.01$m and 1.0m. The lateral hull separation, w, is required to lie between 2.0m and 10.0m inclusive.

The hull shape function has been set to 1; Hull Series 1 will be used to generate offsets during the optimisation.

The objective function has been set to 3, i.e. minimum total resistance.

As it stands, this is an 11-parameter optimisation problem; we must find the length, draft, and three shape parameters for each demihull, and the lateral separation distance between the hulls.

By using a value of 1 for the Ship Special Constraint, the problem can be reduced to a 9-parameter problem. GODZILLA needs to find one hull length, one hull draft, three shape parameters for each hull, and one lateral separation distance.

The size of the problem can be reduced further by using a value of 2 for the Ship Special Constraint. In this case, GODZILLA needs to find 6 (primary) parameters: one hull length, one hull draft, three shape parameters, and one lateral separation distance.

Note that the number of parameters shown on-screen in the GODZILLA panel is 11 for all three cases even though the program actually needs to find fewer than 11 optimal parameters when using Special Ship Constraints.

4 TRIHULLS

4.1 gtri.mlt: Trihull with Three One-tonne Demihulls

The objective in this example is to find the optimum placement of the two outriggers so as to minimise the wave resistance for a ship speed of 6.0ms^{-1}.

The displacement volume of all demihulls is $D = 1.0\text{m}^3$, the length of the demihulls is $L = 11.537$m, and the draft is $T = 0.254$m.

The hull shape function is 0, i.e. the program will use the offsets defined in the offset data section of the file. For Hull Series 1, the given three parameters define a Wigley hull.

Because we are using Hull Shape Function 0 in the GODZILLA section, the three hulls will retain the Wigley shape throughout the optimisation. It is important to note how we handle the minimum, maximum and seed values for this case. A single (dummy) value, of 999.99 is used instead of the method where we provide three comma-separated to signify the minimum, the maximum and the seed values for each shape parameter. When using Hull Shape Function 0 there are no shape parameters to constrain, but the program expects a value in that part of the input file.

The lateral separation distance, w, for the first outrigger (Hull 2) is constrained to lie between a minimum of 2.0m and a maximum of 15.0m from the central hull (Hull 1); for the second outrigger (Hull 3) the separation distance is constrained to lie between a minimum of -15.0m and a maximum of -2.0m from the central hull. (Note how the minimum and maximum values are defined when negative values are used). For both outriggers, the longitudinal separation distance must lie between 0.0m and 12.0m from the centre of the central hull.

This optimisation problem has four parameters; two longitudinal separation distances and two lateral separation distances. For a trihull with symmetrically-placed outriggers, this can be reduced (effectively) to a 2-parameter problem (one longitudinal and one lateral separation distance) by using a value of 1, 2 or 3 for the Special Ship Constraints field. In this example, any of these three values will result in the same hull arrangement because the three hulls have the same displacement, and the lengths, drafts and shapes are not allowed to vary.

4.2 gtrisym.mlt: Trihull with Small Amas

The objective in this example is to minimise the total resistance of a trihull comprised of a four tonne central hull and two one tonne amas. This is a 19-parameter problem; we need to find three hull lengths, three hull drafts, three shape parameters for each hull, and two longitudinal and two lateral separation distances.

In the first instance, run this example using a value of 0 for the Ship Special Constraint. No symmetries will be maintained during the optimisation.

Now run the example using a value of 1 for the Ship Special Constraint. In this case, the outriggers will be forced to be identical to each other and they will be symmetrically placed around the central hull. Thus the problem has been (effectively) reduced to a 12-parameter problem: we need to find one length, one draft and three shape parameters for the central hull, one length, one draft, and three shape parameters for the amas, and one longitudinal and one lateral separation distance.

Using a value of 2 for the Ship Special Constraint reduces the optimisation to a 9-parameter problem: one length, one draft and three shape parameters for the central hull, one length and one draft for the amas, and one longitudinal and one lateral separation distance.

We cannot use a value of 3 for the Ship Special Constraint in this example: that symmetry condition requires that all demihulls have identical displacements.

5 TETRAHULLS

5.1 gtet6.mlt: Tetrahull

The vessel in this example has four identical Wigley demihulls. We will try to find the location of the demihulls in order to minimise the wave resistance of the vessel. The hull shapes and dimensions are not allowed to vary so that this is a 6-parameter problem: we only need to find the longitudinal and the lateral spacings of three of the demihulls since Hull 1 is always located at the co-ordinate origin.

After about 1200 evaluations, hulls seem to form a diamond pattern.

In the description of the parameter N_θ in the Michlet manual, it was mentioned that some calculations in Michlet are exact for Wigley hulls. We can take advantage of this in the present example where we are only concerned with the wave resistance of Wigley hulls.

To see how this can enormously speed up calculations, change the value of N_θ from 256 to -256 in the in.mlt file. The negative sign means that boundary layer displacement effects will not be included.) Also change the number of stations and waterlines so that both are equal to 5. Now run GODZILLA.

It should be very clear that the search proceeds much more quickly. More importantly, exactly the same sequence of 'best' hulls are found as in the original situation where 21 stations and 11 waterlines were used. The reason for this is that Michlet calculates the Michell P,Q functions exactly, irrespective of the number of stations and waterlines (as long as they are both greater than or equal to 5). Note that using a very small number of stations and waterlines is not appropriate for other hull shapes, and certainly not if we are calculating total resistance where we need an accurate estimate of the hull surface area.

6 PENTAHULLS

6.1 gpenta8.mlt: Wave Cancellation Examination

This example is similar to the previous tetrahull example. Here, however, the vessel has one four-tonne hull, and four one-tonne demihulls. As in the previous example we will attempt to reduce the wave resistance by varying only the locations of the demihulls. This is an 8-parameter problem: we need to find the longitudinal and the lateral spacing of the four small demihulls.

6.2 gpenta58.mlt: Total Resistance Minimisation

This example uses one four-tonne demihull and four one-tonne demihulls. Offsets are defined using Hull Series 8, the 8-parameter shape function. Since all lengths, drafts, hull locations and shape parameters are allowed to vary, this is a 58-parameter problem: we need to find 40 shape parameters (8 for each demihull), 5 lengths, 5 drafts, 4 longitudinal spacings and 4 lateral spacings.

This example is really only useful as a template for your designs.

7 INVERSE PROBLEMS

In these examples we will try to estimate the principal dimensions and the shape of vessels given only some observed wave patterns and a set of constraints. Note that many of these examples use small values for some parameters, such as the number of θ-intervals in wave elevation calculations, to reduce program execution time. For "real" problems you will need to use larger values. It is also important to note that these are "toy" problems designed to illustrate some principles and techniques. Real world inverse problems are far more difficult and complicated.

7.1 gid1.mlt: Monohull Shape from Two Given Wave Cuts

In this example wave elevations are specified along two two parallel cuts, one either side of the hull. We will try to predict the principal dimensions of the hull that created those waves by minimising the RMS of the differences between the specified wave elevations and those predicted for hulls found during the search. In this version of Michlet only two wave cuts can be used.

The ship is assumed to be located at x=0.0, y=0.0. We also assume that it is travelling at a speed of 17.6679ms^{-1}.

Hull Series 1 is used to estimate the shape of the hull. This hull series requires three parameters. We also wish to know the displacement volume, the length, and the draft, making six unknown parameters in total.

To run this example, save the file gid1.mlt as in.mlt in the main Michlet directory (i.e. the directory that contains the executable file). This will overwrite the current in.mlt with the example input file. You must also save the wave elevation file gid1recp.mlt as userrecp.mlt in the same directory.

Run the program. The initial guess for the hull dimensions and shape are specified in the in.mlt file. Here we have taken the midpoint of the ranges allowed to certain parameters. For example, the vessel displacement volume is constrained to lie between 100m^3 and 10000m^3, and the initial guess is 5500m^3. Initial estimates for the length, and draft and the three hull shape parameters are calculated in the same way. If you have worked through the previous examples, you should be able to see that the initial hull is half a spheroid. (The three shape parameters are each equal to 0.5, which, for Hull Series 1, means that the sections, waterplanes and buttocks are all elliptical.)

Note that if you go to the Wave Analysis menu (by pressing the W key), you will not yet be able to view the longitudinal cuts since they have not been calculated for the initial hull. Go back to the main menu by pressing Esc.

Press the g key to initialise the search. The vessel initially shown on-screen will change and then the search will stop. The fitness function for this objective is the RMS of the differences between the "target" wave elevations and the 'found' elevations. Now go to the Wave Analysis menu (press W) and then press L to view the longitudinal wave cuts. The plot shows the "target" wave elevations in red and pink; "found" elevations are in green and yellow. Only the red curve and the green curve will be visible because the two target wave cuts (one at $y = -100$m and $y = 100$m) are identical. Similarly, because the vessel is located at y=0.0m, only one "found" curve will be discernible.

The two curves should already be reasonably similar which is not particularly surprising since we have assumed that the location of the hull, its heading, and its speed are all known. The target wave cuts for this example were produced by running Michlet with a Series 1 hull, so that too would tend to bias the agreement.

Go back to the main menu and press G to continue the search. As the search proceeds the hull should begin to look more and more like the Wigley hull that was used in previous examples in this manual. After about 1000 evaluations, press a key to (eventually) stop the search, and look at the comparison of the wave cuts.

After about 7000 evaluations, the target and the found wave cuts are almost identical. So how well did we do in predicting the dimensions of the unknown ship?

The target ship was a 2777.8t Wigley hull (we estimated 2780.5t) of length 100m (we estimated 97.7m) and draft 6.25m (we found 6.75m). Thus the volume and length were predicted reasonably well, but the draft estimate was fairly poor. This is typical in these sort of inverse ship problems for fairly complicated reasons to do with wave generation and its dependence on depth.

The Hull Series 1 shape parameters for the Wigley hull are (1.0, 1.0, 0.0): we found (0.793, 0.279, 0.276) which, on face-value, is barely acceptable. On the positive side, however, it does seem to confirm that the details of hull shape are not that important in the generation of far-field waves by thin ships: displacement, length, beam and location are more important. In a more formal study, the search should, of course, be repeated using different seeds for the random generator.

Although there are obvious military applications for this type of search, there are also several interesting hydrodynamic aspects. For example, if the target wave elevations are all zero, then the problem can be thought of as a wake minimisation problem, or even as a wave resistance minimisation problem. In these cases, as with the estimation of wave resistance using experimental data in towing tanks, results can be unreliable if the wave cut is too short. There are many other research studies being pursued by the present author and other workers that are unlikely to be resolved soon, or without controversy. What are the effects on the ship wave pattern of viscosity, surface tension, ocean waves, currents, and wind? How do these affect the predicted ship dimensions? Adding noise to the wave elevations in the userrecp.mlt file can give some minor insights.

Clearly we should not expect to be able to predict the exact shape of a vessel from, for example, blurry photographs or radar images of wave wakes, but what is it reasonable to expect? It is unlikely that we will be able to predict the speed and length of a vessel purely from considerations of waves in an inviscid fluid. Would including viscosity and surface tension help?

Note that this example cannot be used to estimate the location of the hull. The first hull in Michlet is always located at $x = 0.0$, $y = 0.0$. To estimate the location we could use two hulls and make the first hull very small so that it does not significantly affect the wave pattern.