GODZILLA 9.33

User's Manual

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 is the evolutionary optimisation module for Michlet. Evolutionary algorithms use principles gleaned from natural genetics and evolution to guide the search for global optima of difficult functions. In a ship design problem, individual vessels are selected from a population of random vessels to create new vessels. Weaker individuals are selected only infrequently and do not survive to subsequent generations. When this process of assessment, selection, reproduction, and replacement is repeated many times, the average fitness of the population improves, and hopefully tends towards a global optimum. For a ship design problem, the fitness could be the total drag, in which case we hope that the drag of the best individual in the population tends towards the lowest drag possible without violating any geometrical (or other) constraints we impose. In Michlet, all optimisation problems are minimisation problems.

In Artificial Life research, it is considered de rigeur to name one's pet program; Genetic Optimisation and Design of Zoomorphs Is Leo Lazauskas's Acronym.

2 INPUT DATA

For GODZILLA to run properly, an appropriate input file must be available. This input file is essentially the same as the in.mlt file used for the normal Michlet runs, however there are a number of extra lines that must be appended. In the first instance, copy the file gin1.mlt from the godzilla directory and save it as in.mlt in the main directory (i.e. the directory containing the executable file). This input file will be used in some examples in this manual.

In GODZILLA mode, some options that are available in normal mode are not available. These restrictions are necessary for the optimisation to proceed correctly. For example, if we are using the three parameter hull series in the optimisation, we cannot access the seven parameter series - we can only change the shape of the hulls using the three parameter series.

Furthermore, if the Ship Special Constraint field is non-zero, (i.e. we want to maintain certain multihull symmetries), other restrictions come into play. Specifically, the lengths, drafts, hull spacings and shape factors cannot be changed on-screen.

Trim and sinkage values are always set to zero in GODZILLA mode. For GODZILLA to work correctly, trim and sinkage would have to be calculated for each candidate vessel (remembering that the variety of possible hull shapes is huge), and this is not possible in current versions of the program.

Note that a file suitable for GODZILLA can also be used for normal Michlet operation. In normal mode, the lines of the input file specific to GODZILLA (i.e. those immediately following the hull offset table and hull dimensions) are not read by the program.

The first few required parameters are described below. Constraints on the ship and the individual hulls are described in separate sections.

2.1 Random Seed

This must be an integer between 0 and 32000 inclusive. The value entered here will determine the random number sequence used to generate new populations of vessels.

2.2 Number of Ships

This parameter determines the size of the population of vessels and must be an (integer) value between 3 and 120, inclusive.

There is no way to determine the optimum population size for evolutionary algorithms. As a guide, very small populations are likely to be trapped by local (i.e. suboptimal) minima. As the optimisation process progresses, the members of the (small) population become very similar to each other and this makes it difficult for most algorithms to search configurations much unlike those already in the population. On the other hand, very large populations can take a long time to evolve towards optimum solutions and are probably best for multihull problems where there are many parameters participating in the optimisation. Running times may then be prohibitive on personal computers. For monohulls, a population of about 20 should give reasonable results. If there are many constraints on the hull, larger populations might be more successful, given enough time.

2.3 Objective Functions and Variants

Each objective function has an associated number of variants that can be used in the optimisation. In the present version of GODZILLA, the available objective functions (and their variants) include:

- 0 : Minimise surface area. Variant = 0.
- 1 : Minimise viscous resistance. Variant = 0.
- 2 : Minimise wave resistance. Variant = 0.
- 3 : Minimise total resistance. Variant = 0.
- 4 : Minimise maximum dR_w/d_θ for a range of wave propagation angles. Variant = 0.
- 5 : Minimise wave height along a radial cut. Variant = 0.
- 6 : Minimise wave height along a longitudinal cut. Variant = 0.
- 13 : Minimise required power (= Total Resistance × Speed). Variant = 1.
- 100 : Minimise difference between hull section areas and/or waterplane areas and user-supplied section and waterplane areas. Variants = 0,1, or 2.
- 10400 : Find vessels that make a user-specified wave pattern. Variant = 0.

See the separate Examples manual for details on how to use these objective functions.

2.4 Design Speeds

2.4.1 Number of Speeds

GODZILLA allows optimisations to be performed over a range of design speeds. The number of design speeds must be an integer between 1 and 50 inclusive. The fitness of an individual is a weighted average of the fitnesses (see below). For objective functions 4,5 and 6 weightings have no effect as I have not yet thought of a meaningful way to use them.

2.4.2 Design speeds and weights

Design speeds and their weighting must be given as pairs of comma-separated decimals. (No trailing comma at the end). There must be as many pairs as specified in the number of design speeds. For example, for a three design speed problem, we might use the list:
3.00, 0.1
5.50, 0.5
13.0, 0.4
to specify that we want to optimise at 3.0 ms^{-1} (with a weighting of 0.1), 5.5 ms^{-1} (with weighting 0.5), and 13.0 ms^{-1} (with weighting 0.4). In this example, if the fitnesses at the three design speeds were calculated as f_1,f_2, and f_3, then the total fitness, f, for the individual would be

$$f = (k_1 f_1 + k_2 f_2 + k_3 f_3)/3 \tag{1}$$

where the k_i are the three weights.

Note that for a one design speed problem, the weighting factor will always be set to a value of 1.0, regardless of what the user has specified for the weight.

Also note that weightings can be equal to zero, but that they will still be "counted" in the averaging process for the fitness. The same vessel will be found and it will have the same resistance, but the reported fitnesses will be different because of the number of design speeds.

2.5 Hull Shape Functions

The hull offset types that can be used during the optimisation include:

- 0 : Use the hull offsets defined earlier in the in.mlt file. The (non-dimensional) value of these offsets will remain the same throughout the optimisation run. Thus only the length, draft, and hull separation distances will be modified. This hull shape function requires no other parameters.
- 1 : Series 1.
- 2 : Series 2. (Crippled).
- 4 : Series 4.
- 7 : Series 7.
- 8 : Series 8.
- 9 : Series 9.
- 20 : Series 20.
- 32 : Series 32.
- 42 : Series 42.

Details of the mathematical series are given in the Michlet manual.

3 SHIP CONSTRAINTS

To constrain the size of the vessel, specify the minimum and maximum overall displacement of the vessel, the minimum and maximum overall length of the vessel, and the minimum and maximum overall width. For monohulls, the ranges can be made very wide since constraints on the displacement, length and beam of the hull will be specified further on in the file. For multihulls, the constraints on LOA and WOA will act to limit the location of the individual hulls and it is important to co-ordinate these constraints with the limits on length, beam, and hull separation for the individual demihulls. Similarly, it is important to co-ordinate the overall displacement with the limits imposed on the individual hull displacements.

$\overline{GM}_{TOA}$ is the overall transverse metacentric height. For monohulls, the value entered here should be the same as that entered below for the hull itself. For multihulls it can be different. In fact, in a multihull arrangement, the $\overline{GM}_T$ of the individual hulls can be negative, while the overall $\overline{GM}_T$ is positive.

$\overline{GM}_{LOA}$ is the overall longitudinal metacentric height. In the present version of GODZILLA, only the overall value can be constrained. The individual values (i.e. $\overline{GM}_L$) of each hull are not constrained (except, of course, for monohulls).

3.1 Ship Pressure Signature Constraints

Leave it set to 0.

3.2 Special Ship Constraints

This field is used primarily to maintain certain hull spacing and other symmetries during the optimisation of multihulls. For monohulls, this field must be 0. Constraints for tetramarans and pentahulls have not been implemented so use 0 for these vessels as well.

For these special constraint options to work, Hull 1 must be the central hull, Hull 2 must be the starboard ama, and Hull 3 must be the port ama.

Also note that if this option is set to a non-zero value, certain options will not be available in the graphic version. Specifically, you will not be able to change on-screen the length, draft, hull spacing or shape factors.

3.2.1 Dihull Constraints

There are three valid constraint options for dihulls, namely:

- 0 : None. The two demihulls can have different displacements, lengths, drafts and shapes, and can be located anywhere (but be reasonable, don't try to put them on the moon!)
- 1 : Proa with similar demihulls.
- 2 : Identical demihulls.

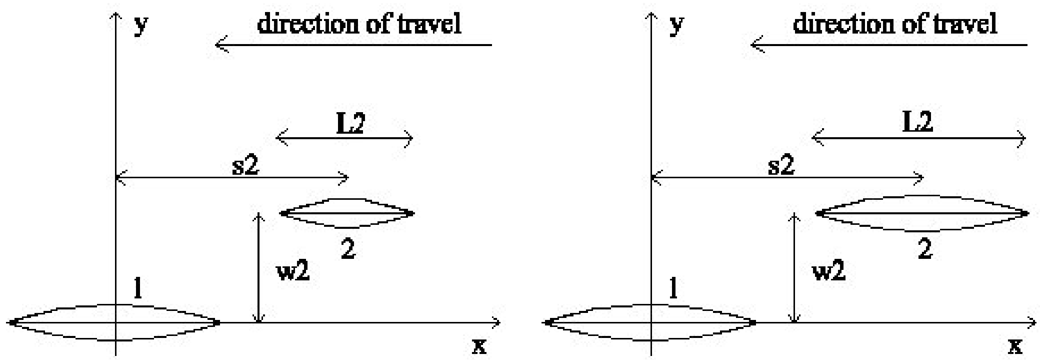

Figure 1: Dihull symmetry condition 1 (left) condition 2 (right).

For option 1, illustrated in the left side plot of Figure 1, the displacements, lengths and drafts of the two demihulls can all be different, but the shapes of the demihulls will be identical, i.e. if Hull 1 offsets are defined by Hull Shape Function 1 with shape parameters 0.5, 0.5 and 0.5, then Hull 2 will have exactly the same values for the shape parameters. The dimensional offsets might be different for the two hulls, but the non-dimensional offsets will be the same for both hulls.

Option 2 is illustrated in the plot at the right of Figure 1. The demihulls are identical in all respects. For a standard (side-by-side) catamaran, set the longitudinal stagger to zero, and do not allow it to vary during the optimisation by setting the minimum and maximum longitudinal stagger to zero.

3.2.2 Trihull Constraints

There are four valid constraint options for trihulls, namely:

- 0 : None. The three demihulls can have different displacements, lengths, drafts and shapes, and can be located anywhere.
- 1 : Trihull with identical amas.
- 2 : Trihull with identical amas of same shape as the central hull.
- 3 : Trihull with three identical demihulls.

Option 1 is illustrated at the top of Figure 2. The amas are identical in all respects, and can have a different shape to the central hull.

Option 2 is illustrated at the bottom left of Figure 2. The outriggers (Hull2 and Hull 3) are identical in all respects, and have the same shape as the central hull (Hull 1) which can have a different displacement, length and draft. The outriggers are placed symmetrically with respect to the central hull.

Option 3 is illustrated at the bottom right of Figure 2. The three demihulls are identical in all respects. The outriggers (Hull 2 and Hull 3) are placed symmetrically with respect to the central hull (Hull 1).

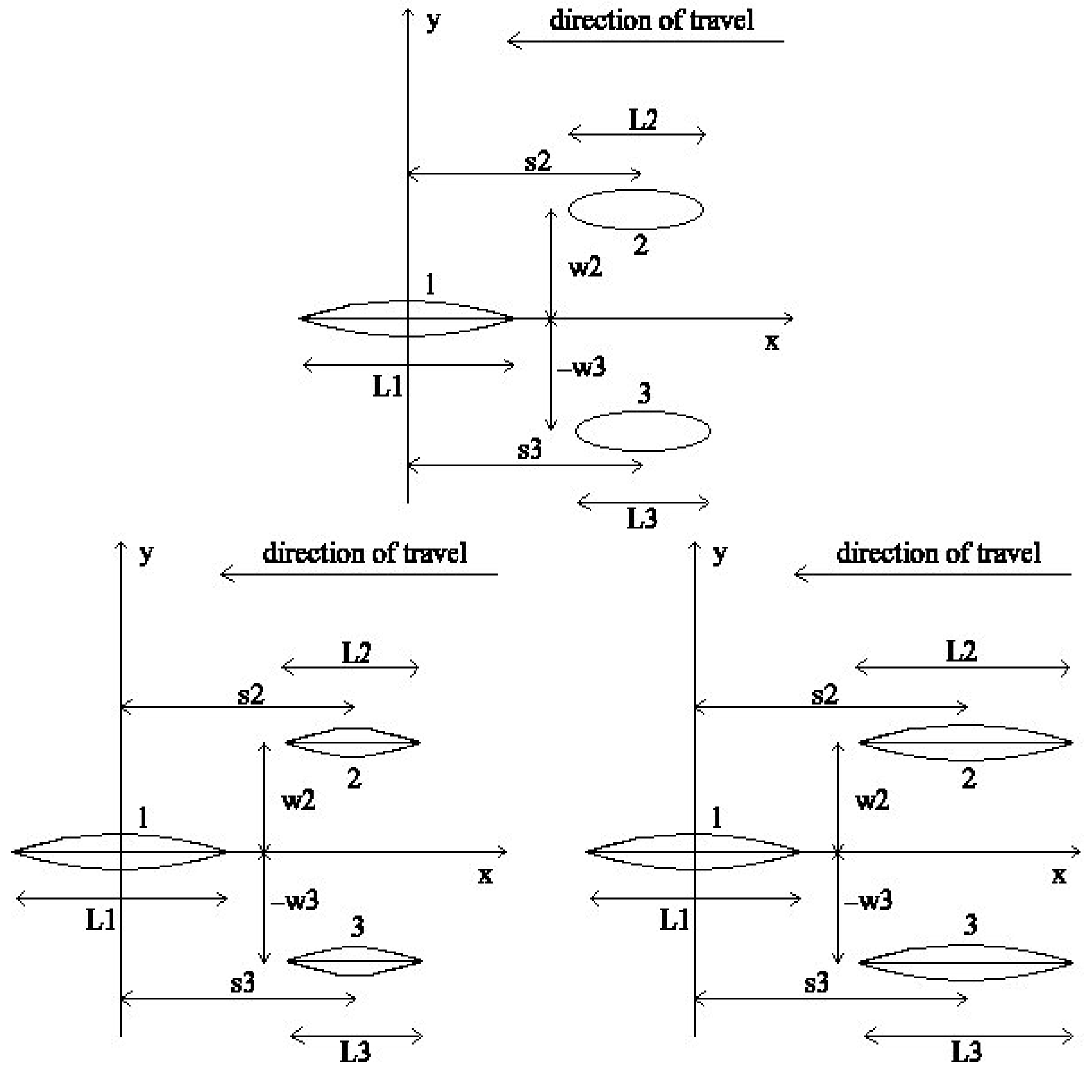

Figure 2: Trihull symmetry condition 1 (top), condition 2 (bottom left), condition 3 (bottom right).

4 HULL CONSTRAINTS

4.1 Primary Constraints

Primary constraints are those that limit the fundamental parameters that define the vessel. These are the displacement, length, draft, and parameters defining the shape of the individual hulls, and for multihulls, the hull separation distances. Dimensional hull offsets and beam are not fundamental parameters because they are calculated from the other hull parameters and the displacement.

The minimum displacement and maximum displacement must be entered as decimals. If, for example, you want the hull displacement to remain constant during the optimisation process, set the minimum displacement and maximum displacement to exactly the same value.

The minimum length and maximum length must be entered as decimals, as must the minimum draft and maximum draft. If you want the hull length and/or draft to remain constant during the optimisation process, set the minimum and maximum to exactly the same value.

The next parameters to specify are the limits on the hull separation distances relative to the first hull, namely, the Minimum Longitudinal Offset, Maximum Longitudinal Offset, Minimum Lateral Offset, and Maximum Lateral Offset. For Hull 1, these must all be equal to zero.

The next parameters to be specified are the limits on the hull shape parameters and 'seed' values.

If the hull shape function specified earlier in the file is 0, then a single (dummy) value must given. Example files use 999.999 to emphasise that it is a dummy value.

For other hull shape functions, three comma-separated decimals must be given for each shape parameter. The first number in the triplet specifies the minimum value for the parameter, the second number is the maximum allowable value for the parameter, and the third parameter is the 'seed' value, i.e. the first individual in the population will be assigned the seed value for that parameter. Maximum shape factors must be greater than or equal to the minimum shape factors. If the minimum and maximum are identical, the value for that parameter will remain constant during the optimisation.

For example, suppose that the hull shape function is 1, and that we have the following list:
0.0, 1.0, 0.5
0.0, 1.0, 1.0
1.0, 1.0, 1.0

The first row specifies that the first shape parameter will be allowed to take values between 0.0 and 1.0, and that the "seed" value will be 0.5. The second row is the same as the first, except that the seed value for the second parameter will be 1.0, i.e. parabolic in Hull Series 1. The third parameter will not change from its initial value of 1.0 because the minimum and the maximum values are identical.

For hull shape function 7, seven triplets must be given on separate lines; hull shape function 8, requires eight triplets, . . . , hull shape function 32 requires thirty-two triplets.

The length and draft of the seed hull will be equal to the values specified in the first part of the in.mlt file. If the vessel with these initial parameters is good (i.e. has a low fitness value) then it will survive in subsequent generations; if it fails in competition with some other vessel during the optimisation process, it will be replaced in the population.

Seeding populations is a tricky subject. It can be used to focus the optimisation on a highly promising region of the search space, but it can also bias the search so that sufficient attention is not given to other, possibly better, regions of the search space.

4.2 Secondary Constraints

Minimum and maximum values must be specified for the following secondary constraints:

- hull beam, B.
- hull surface area, S.
- waterplane area, A_{wp}.
- longitudinal centre of buoyancy, LCB.
- vertical centre of buoyancy, VCB.
- longitudinal centre of flotation, LCF.
- transverse metacentric radius, BM_T.
- transverse metacentric height, $\overline{GM}_T$.
- transom stern area, A_t.
- transom stern beam, B_t.
- transom stern draft, T_t.
- block coefficient, C_B.
- midship area coefficient, C_m.
- prismatic coefficient based on midship area, C_p.
- waterplane area coefficient, C_{wp}.
- length-to-beam ratio, L/B.

- length-to-draft ratio, L/T.
- beam-to-draft ratio, B/T.

If any of these secondary constraints are unimportant for the problem of interest, a broad range can be used for the parameters, but be reasonable. It is important that the range for these values be broad, but not too extensive and not too narrow. For example, if the prismatic coefficient was constrained to lie between 0.5 and 0.501, it might take a very long time to even find one hull with a C_p in that narrow range. On the other hand, if the range is too broad, many hulls that are uncompetitive will be assessed, which might slow down the process considerably. On the third hand, we don't want to miss any unusual, interesting combinations by narrowing the range too much. Even-handed, carbon-based, Naval Architects are probably quite good at this sort of thing!

If, during a run, you notice that many members of the population are flagged as having violated one or more of the constraints, and this persists for what seems an unreasonable time, it is possible that one or more of the secondary constraints is constrained either much too narrowly or allowed too much freedom. Only experience can be your guide here.

4.3 Hull Pressure Signature Constraints

Leave it set to 0.

4.4 Special Hull Constraints

- 0 : None
- 1 : Box at xbox, zbox

This constraint (an idea of L.J. Doctors and A. Day) demands that the hull be able to hold a number of rectangular boxes with dimensions L_{box}, T_{box} and B_{box}, and that the centre of the boxes are to be located X_{box} metres from the bow, and Z_{box} below the waterplane.

For example, the input line
1,1,10.0,2.0,4.0,15.0,3.0
specifies that the hull must be able to hold one rectangular box of length, $L_{box} = 10.0$m, draft $T_{box} = 2.0$m, and width $B_{box} = 4.0$m. The centre of the box will be 15.0m from the bow, and 3.0m below the waterline.

Note that only one box per hull is allowed in the current version.

- 2 : Box at midships

Same as for type 1, but here we only input L_{box}, T_{box} and B_{box}. The program will attempt to fit the box with its centre at midships and with the top of the box at the waterline.

For example, the input line for the previous example would be:
2,1,10.0,2.0,4.0

- 3 : Box at LCB

Same as for type 2 but here the program will attempt to fit the box with its centre at the hull LCB and with the top of the box at the waterline.

For example, the input line for the previous example would be:
3,1,10.0,2.0,4.0
In this case the program would try to fit the box at the hull LCB instead of midships.

- 4 : Box at stern

Same as for types 2 and 3 but here the program will attempt to fit the box with its end at the hull stern.

For example, the input line for the previous example would be:
4,1,10.0,2.0,4.0
In this case the program would try to fit the box at the stern instead of the LCB.

- -1 : Shape constraints and box constraints

If the special hull constraint code is equal to -1, then constraints on the hull shape can be included in addition to box constraints. (The allowable hull shape constraint types will be described in the next subsection.) For example, we could use:
-1,1,1,1
0
to allow hull shape constraints on waterlines, sections and the keel-line, but no box constraints.

To include shape constraints as well as box constraints, use:
-1,1,1,1
2,1,10.0,2.0,4.0

The first line means that hull shape constraints will be used. Here they are all of type 1 for sake of example, but see below for other options. The second line implements the box example described earlier.

4.5 Special Hull Shape Constraints

These codes can be used in conjunction with the box constraints described above. After the -1, on the input line, there must be three comma-separated integers. For example, let the input line be of the form -1,X,Y,Z where X, Y and Z are integers.
X is the type of constraint on waterlines,
Y is the type of constraint on sections, and
X is the type of constraint on the keel-line.

Note that these constraints are not necessary for many of the hull shape series available in Michlet and will only slow down the search process. They are best used with, for example, Hull Series 20 or more complicated hull shape functions. Values of 0 for X,Y or Z means that no constraint check will be performed for waterlines (respectively, sections and the keel-line) during the optimisation.

IMPORTANT NOTE: Although these constraints can help to remove unwanted oscillations in hull lines, they are not guaranteed to work in all cases.

X=1 : Limit waterline slope sign changes.

The input line
-1,1,0,0
means that the number of slope sign changes in the waterlines is constrained.

X=2 : Force convex waterlines.

The input line
-1,2,0,0
specifies that waterlines must be convex.

Y=1 : Limit hull section slope sign changes.

The input line
-1,0,1,0
means that the number of slope sign changes of each section are constrained.

Y=2 : Force convex sections.

The input line
-1,0,2,0
specifies that sections must be convex.

Z=1 : Limit keel-line slope sign changes.

The input line
-1,0,0,1
means that the number of slope sign changes of the keel-line are constrained.

Z=2 : Force convex keel-lines and buttocks.

Not yet implemented.

Note that the constraints can be combined. For example, the input line
-1,2,1,1
means that waterlines must be convex, and that the slope sign changes for sections and the keel-line are constrained. A few examples have been included to illustrate how these constraints are implemented, however, as noted earlier, they are not guaranteed to work in all cases

4.6 Multihull Constraints

The constraints for the other hulls in a multihull arrangement are the same as for the first hull, except that limits on the position of the hulls must also be specified. These (primary) constraints are specified in Minimum longitudinal offset, Maximum longitudinal offset, Minimum lateral offset, and Maximum lateral offset.

Note that these values, in combination with the lengths and widths of the individual hulls should agree sensibly - or at least not conflict catastrophically - with the limits on the vessel's overall length and width specified earlier. For example, it is pointless to set the limit on overall length of the vessel as 10.0m if the individual hulls are allowed to vary between 20.0m and 30.0m. In this case the vessel would always violate the specified constraints.

5 RUNNING GODZILLA

For this section of this manual, I assume that you have copied the file gin1.mlt from the godzilla directory to the in.mlt file in the directory containing the executable file.

- To run GODZILLA, execute the godzilla.bat file.
- The main difference to the lines drawing screen in GODZILLA mode is that the small panel at the bottom right now contains information specific to GODZILLA, rather than hull offset information as in normal Michlet mode. In GODZILLA mode, some Michlet options are not available, some others have been retained or modified. The options available from this screen can be seen by pressing the ? key. Press Esc to return to the Lines Drawing screen.
- Press p to see the current state of the population of hulls. As GODZILLA has not been initialised yet, the fitness of each hull will be a large number, indicating a poor fitness. If the word Violated appears next to one of the hulls, that indicates that the hull dimensions have somehow violated the constraints defined in the in.mlt file. This is not necessarily a problem.
- Press h to see all the hull waterlines.
- Press H to see the current hull's waterlines.
- Press v to see all the hull sections.
- Press V to see the current hull's sections.
- The ship displayed on the screen before GODZILLA is initialised is the one defined in the non-GODZILLA section of the in.mlt file. If the hull shape function has been set to 0 in the GODZILLA section of in.mlt, then the (non-dimensional) offsets of the hull will not change during the optimisation, and thus the lines drawings will remain the same during the optimisation. If the hull shape function is not equal to 0, then the hull shape is determined by the particular hull shape function and not by the offsets specified in the first part of the in.mlt file. In other words, for parametrically-defined hull shape functions, the offsets in the in.mlt file are merely placeholders, or dummy values. When GODZILLA is initialised, the displayed vessel will be replaced with the details of the best vessel in the population, i.e. the vessel with the smallest fitness value.
- Press g once to run GODZILLA. The GODZILLA panel will display the words INITIALISING, then GODZILLA STEPPING, and finally GODZILLA IDLE when it has finished its initialisation sequence.

- Press p to display the population screen. The best hull in the population is shown in green. At this stage, the population is fairly diverse insofar as the individual vessels are not too similar to each other and so display a wide range of fitness values. Some vessels might be tagged as Violated. This means that the combination of parameters for that vessel has violated one or more of the constraints specified in the in.mlt file.

- If you now press G, GODZILLA will continue to run until you press a key again (for example, press the Space Bar). For some problems it can take quite a long time for GODZILLA to finish evaluating a complete population. Thus there can be a considerable delay before pressing a key to interrupt GODZILLA, and the program actually going into GODZILLA IDLE mode. Of course, if it is imperative that the program stop immediately, you can always press Ctrl-Alt-Del to abort the program.

5.1 A Few Concluding Remarks

Automatic hull optimisation is a difficult, computer-intensive process and I do not claim that GODZILLA will find optimum hulls for every situation. (For most practical applications, there is probably no such thing as a unique optimum hull). At most, GODZILLA may evolve vessels that are worthy of more detailed investigation, or that allow some quantification of subsequent naval architectural compromises. At the least, it might just be fun!