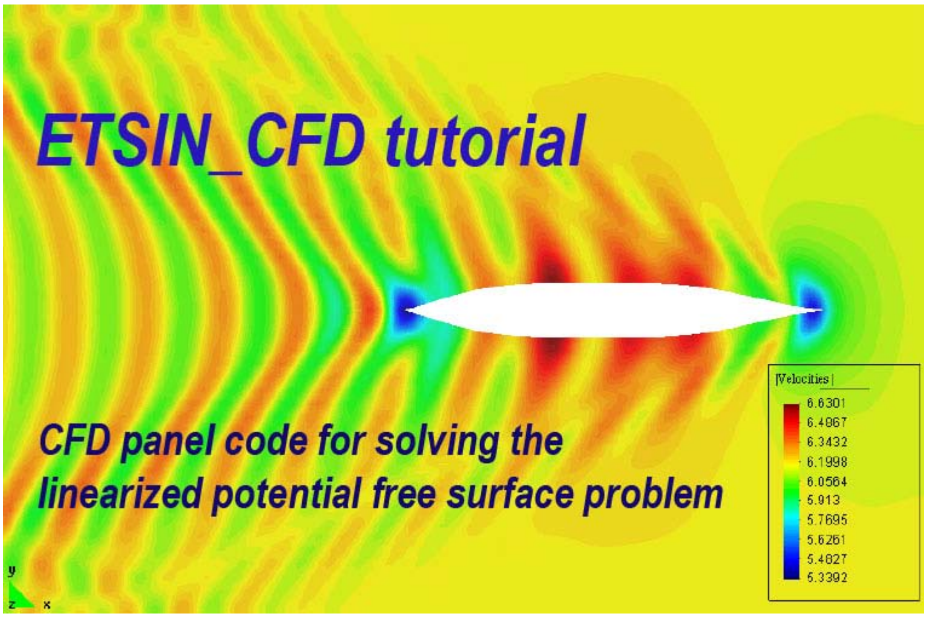
UPM

canal.etsin.upm.es

ETSIN



Coordinator: Antonio Souto Iglesias.

ETSIN-UPM Model Basin Research Group (CEHINAV)

Contents

Introduction

ETSIN_CFD is a computer code aimed at the optimisation of ship hull forms based on the calculations of potential flows with free surface [3] [9].

Nowadays, the numerical simulations carried out on computers have become an irreplaceable tool for the optimisation of forms. The commonly called CFD (that stands for Computational Fluid Dynamics) are used in the industry to predict the behaviour of fluids without having to resort to expensive scale models that simulate a process.

These tools allow us to test a greater number of models in a shorter time and with reduced costs compared with traditional tests. Thus, it is logical to use them in the first stages of the design in order to predict concept failures as soon as possible [6].

One of the main applications for the naval industry is the optimization of ship forms and the calculation of forward resistance. Although the accuracy of the calculation of resistance is quite low, the potential flow methods are the most common in improving ship forms. The method we have conceived is of this type.

Numerical results are very interesting because one can visualize the fluid’s behaviour at each point of the hull, whereas towing tests only provide overall results such as the resistance or the trim angle. For this reason it is much easier to devise modifications aimed at reducing the resistance, without relying on the investigator’s intuition and experience.

Nevertheless, towing tests in tanks are still necessary. With the present state-of-the-art techniques it is not possible to compute the total resistance within the precision levels required by the shipyards and some important phenomena (for example the interaction between the propeller and the hull) do not yet have a satisfactory model. However, numerical tests allow for the reduction of the number of scale tests by only needing the construction of one or two models. With these definite tests, it is possible to determine the propeller and the propulsive plant.

The towing tank of the Naval Architecture Department of Universidad Politécnica of Madrid has been investing concerted efforts in the development of software for numerous fields, such as manoeuvrability, seakeeping and, of course, forward resistance optimisation [8].

The works in the field of forward resistance began in the frame of the Spanish Challenge for the 1992 America’s Cup. Since this event has been described on numerous occasions as the "formula 1" of sailing, the basin group had to commit itself completely to the world of the computer simulations [9] in order to meet expectations. The codes designed at that time, and improved later, have been put under numerous processes of validation, comparing our results with those obtained from commercial programs of recognized prestige and with the available experimental data [4]. At the moment, new programs are still being developed with the purpose of analyzing more intricate processes like the turbulent flows with formation of vortices, and many efforts are dedicated to the use of the software that has already been validated with practical aims.

Using this Manual

The analysis of a problem using this code consists of a three stage process: First, the *preprocess* that consists in the treatment of geometric forms and grid; Second, the calculation itself; and third, the postprocess which is the visualization and interpretation of the results.

In agreement with this, we have divided the manual into several different parts. The first one is a summary of the numerical and mathematical foundations of the calculation module. In the second part, the setup process of the code is described. The third part consists of a tutorial. Three practical examples the three necessary steps that we have described previously (preprocess, calculation, postprocess) are explained.

Foundations

The panel method

The physical laws that govern the behaviour of fluids (water inside a glass as well as gases at supersonic speeds) can be summarized in relationships between the variation of fluid speed and the causes that create this variation. Among these causes are the pressure differences, the viscosity between particles and finally the action of gravity. In the problems of naval hydrodynamics (that this software can deal with), viscosity has a residual importance and will not be considered. Wave generation is basically a non-viscous phenomenon albeit very difficult to solve.

Since the problem is perfectly well known, one could think it is easy to solve, but this is far from being true. Except in very simple cases (those without free surface) an analytical solution does not exist and it is necessary to resort to approximate numerical methods. The tricky part is that an infinite number of numerical methods exist. A great family is formed by the ones that distribute the control points across the whole volume of the fluid and later impose on these points discrete forms of the differential equations that govern the phenomenon. The method used in this report belongs to the other great family, characterised by having control points on the boundary surfaces of the domain. Particular solutions are used and are combined to obtain a global flow that fulfils the boundary conditions of the problem governed by the Laplace equation which refers to the velocity potential.

As the Laplace equation is linear, we can combine its particular solutions to obtain a new one that is still a valid solution. The elementary solutions are sources located on the boundaries (discretized in panels) whose intensities are adjusted in order to have the flow verify the boundary conditions: impenetrability on the hull and the free surface being a streamline under atmospheric pressure...

The potential speed field produced by a flat panel, on which a uniform distribution of sources has been placed, is known. Thus, if one considers any panel, and the knowledge of the intensity of the sources, the speed induced at any point in space can be directly calculated. In the developed method, both hull and free surface of the water near the boat are represented as two surfaces formed by panels. One assumes that for each panel there is a uniform distribution of sources, whose intensities are unknown. Moreover, a control point is located on each panel’s centroid. The control points that lay on the hull have to fulfil one rule: the speed vector has to be tangent to the panel. The speed vectors located on the free surface of the water must fulfil two conditions:

The cinematic condition that indicates that the speed of the liquid must be tangent to the free surface

The dynamic condition that requires the pressure to be equal to the atmospheric pressure.

We can combine these two equations in a single equation. The problem consists in figuring out the value of the sources on each one of the panels so that the conditions are fulfilled for all the control points.

As opposed to the methods of discrete volumes (finite elements and finite differences), the panel method has the great advantage of allowing us to obtain more accurate results for the speed field on the hull surface using a much smaller number of control points. This allows us to do a quick and relatively precise calculation of the free surface deformation. The optimisation of forward resistance due to wave generation requires the knowledge of the pressure distribution along the hull and the deformation of the free surface. This is why the panel method is a good choice in engineering.

Technical Notes

It is very important for a correct operation of the program thatthe decimal separator symbol is assigned to the point and not to the comma. The default option in Windows is the comma. In order to change this option it is necessary to modify the settings in the regional configuration menu of the Control Panel.

Example 1: Liner without bulbous bow: “S60”

In this and the following two chapters, 3 examples are presented whose guided execution will let us explore the system’s possibilities. The first example deals with a ship without bulbous bow. The second consists in creating the mesh of a ship’s bulbous bow with the corresponding particularities due to the existence of the bulbous bow. The third example corresponds to a racing yacht with its corresponding difficulties due to the existence of several appendages. In the first two cases, the geometric definition is based on a set of lines. In the third case we directly import a definition of the hull through surface entities.

Pre-processing

The calculations we are going to do are based upon the geometric form of the ship we want to study. These forms are defined in an IGES file which contains a CAD definition of the hull to mesh. With this definition, we will try to obtain a set of lines in three dimensions that will let us generate, by means of surfaces, a computational representation of the submerged part of the hull.

In order to make the preprocess easier, the geometric representation has to meet with a few requirements.

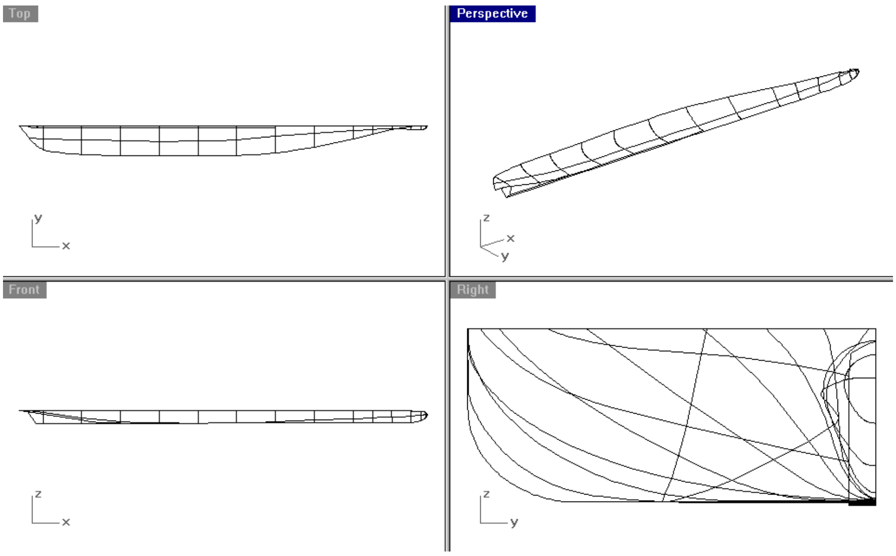


Figure 1

The shape shown in figure 3 is ready to start the pre-process. The lines defining this hull are all connected at the different vertex nodes of the draft.

After the preprocess, the shapes have to become surfaces that cover the entire hull. Keeping this in mind, it is advisable that the initial drawing be the sort of mesh that facilitates the later performance of grid generation. Thus, we will try to ensure that this geometric design, formed by lines, will be the base for those surfaces. Therefore, we will use, as a basis, some frames and longitudinal lines that define the future surface patches, trying to set them as square as possible.

We use Rhino© as the CAD software but any other package capable of exporting IGES files (that will be used by GiD) is just as good. In order to illustrate the handling we have used a Series 60 model, available in the examples (S60_0.3DM).

The reason for using a CAD software before generating the mesh is to define the hull with enough lines for GiD to be able to generate surfaces from them. These surfaces will define the submerged part of the hull and once processed will be the basis for the mesh generated with GiD.

Meshing with GiD

Using GiD, the first step is to import the IGES file of the form (*IGES read*) and to save it (*save as*) as a GiD file[1].

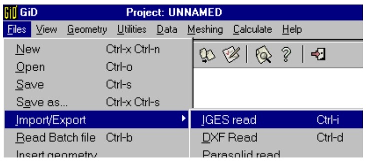


Figure 2

In order to avoid problems that could occur while using GiD, it is safer to copy all the imported lines to an auxiliary layer. The use of several layers simplifies the meshing work and in some cases you may have to erase a few empty layers created when reading the IGES file.

With the imported lines, we can generate surfaces using the command *Geometry* (*Create* - *NURBS Surface* - *Automatic*). We want them to be exactly defined by 4 lines.

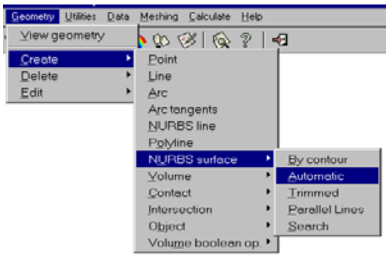


Figure 3

We notice that all 4 sided-surfaces have not been created. Occasionally the line system is modified during the importing process of the IGES file. For instance, once imported, a single segment could be split into two. Thus making it impossible to create surfaces from such lines. To solve this problem you can use *Geometry>Edit>Join Lines end Points* and select the 2 lines to be joined. To rebuild a line we create a polyline from the fragments, *geometry>Create Polyline*.

<u>Possible errors[2] in the surface creation phase:</u>

[1] Exemple file : S60_0.gid

[2] Example file: S60_1.gid

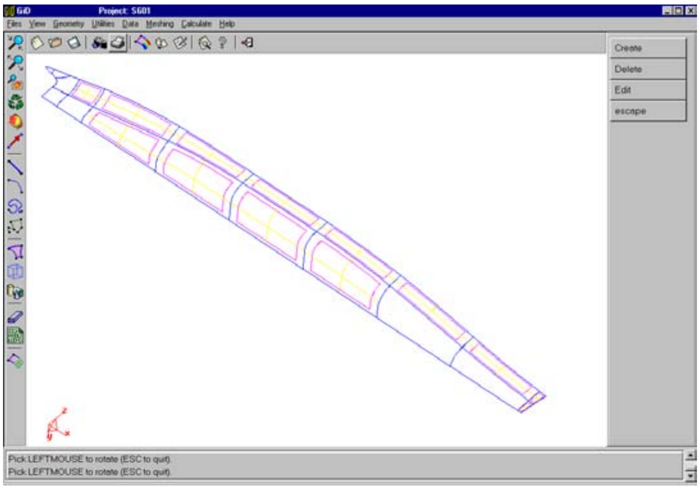


Figure 4

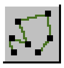
Error I: 3-sided surface has been created in the stem region.

Cause: The surface that was created has 4 sides but because one is very short, we can hardly see it. This occurs when we are not careful enough in the previous step.

Solution: Since this surface is not valid, we remove it.

Figure 5

From the two lines that form the side with the small segment we create a polyline with *Create Polyline*.



Once created, it will constitute a single entity that has to be converted into a NURBS line to ease its handling. Moreover, each polyline can only be used to bound a single patch which produces several small shortcomings. We use the *Convert to NURBS* command to perform this conversion.

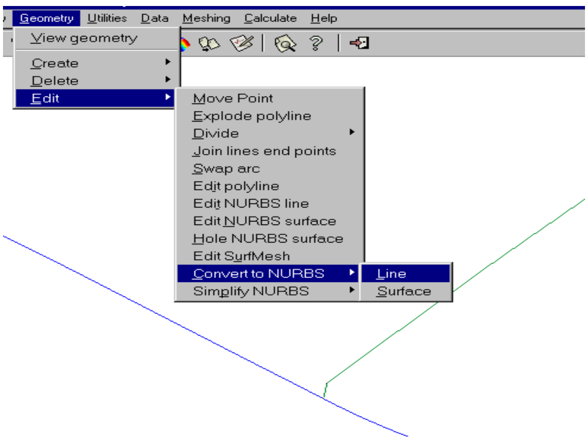


Figure 6

<u>Error II:</u> A few panels have not been created.

Causes: it might be that the surface is defined by 5 sides instead of 4. As in *error I*, there are incomplete lines. A gap between 2 lines preventing the surface from being created could also exist.

Solution: using a polyline and a NURBS Line, we create a single line from the 2 segments as in error 1. If the problem is due to the existence of a gap, we have to join the line ends. Since there are 4 corners for each panel and we do not know which one is the bad one, we have to test them one by one.

In the example we are studying, the problem is due to the existence of divided lines. We can also mention the fact that even after joining the lines or creating the polyline and transforming it into NURBS, we should finally be able to create the surface but this is not always the case. A few nodes can be misplaced and thus, it is never superfluous to check the connections for these points through the *Join Lines end Points* function.

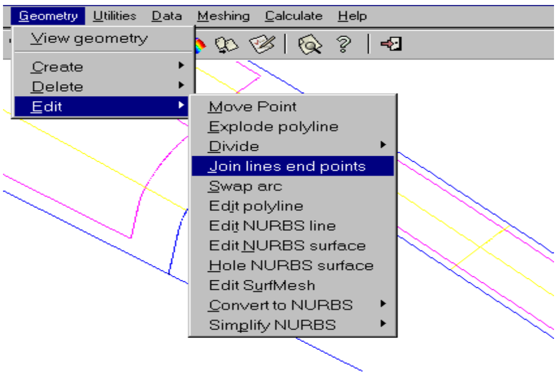


Figure 7

After fixing these errors and creating all the missing 4-sided surfaces, the model should look like what is shown in figure *8*[3].

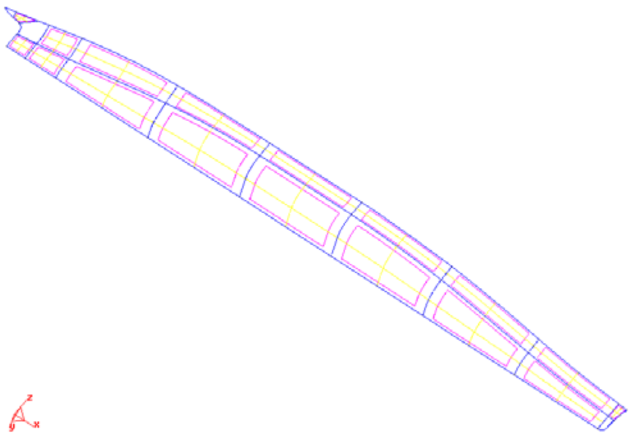


Figure 8

[3] Example file : S60_2.gid

There are still 3 surfaces to be created: one for the stem region and 2 for the stern region. The surface at the stern end and the one at the stem end, have only 3 sides. The third patch from the left of the stern region is defined by 5 sides.

Error III: in some areas, 5-sided patches are the most natural.

Causes: in a few areas, especially for the stern, it is not always easy to only obtain four sided surfaces.

Solution: by joining two lines we can define these surfaces with only four sides/lines. This will result in a new problem for the adjacent panel that will not be able to use this new line as an edge. To solve this, we have to duplicate the original line in order for the nearby panel to use it.

We will now analyse in detail the S60 example where its stern presents an area with such a problem:

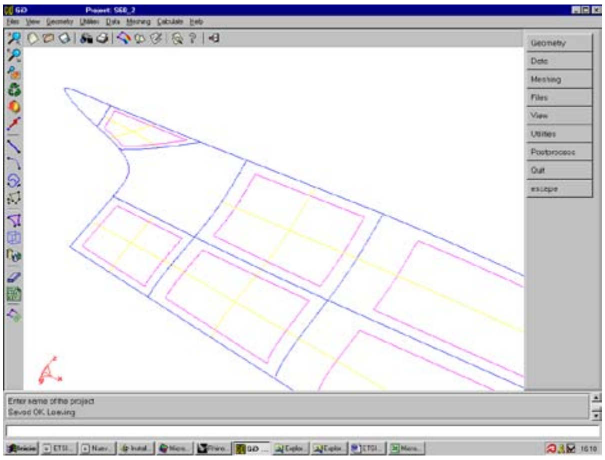


Figure 9

We are going to apply the previously described method to this surface and more precisely to the two left segments. Because it is used by a neighbouring panel, we have to duplicate the upper segment. For this, we use the *Copy* command.

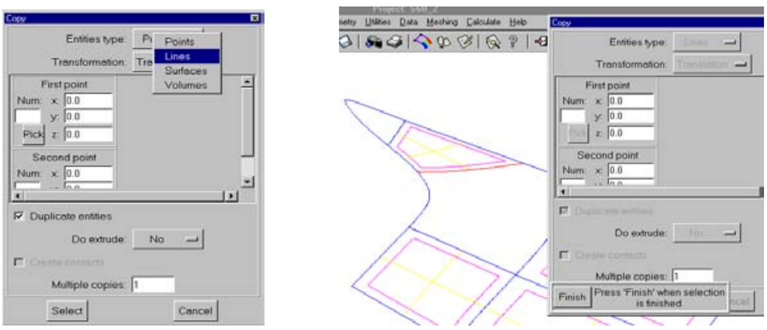


Figure 10

Once the line has been chosen (in red), press *Finish*. Since the two lines are superimposed, we will not notice any change. Before joining it, the new line has to be identified by the function *List*.

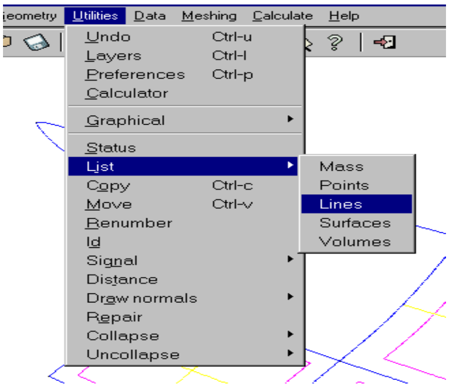


Figure 13

We "encircle" the area containing the elements that we want to identify. In this case we will choose a small zone that includes part of the original line and the duplicated one.

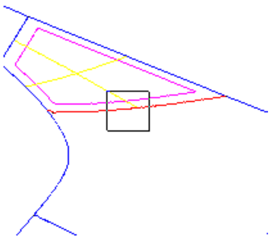


Figure 11

The following box should appear:

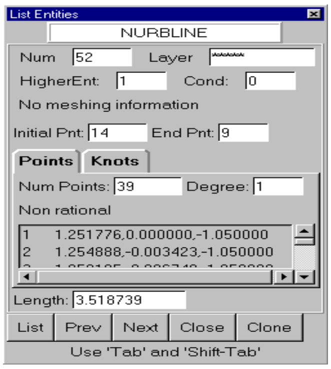


Figure 12

Where *Num* identifies the line. We have to also look at *HigherEnt* which reports the number of patches that lie on this line. In this case there is only one. We can therefore deduce that line 52 is the one bounding the stern panel. If we take the next line (copy 53) using Next, we see that no patch lies on it.

Lets create the line joining line 53 with the lower one. To do this, the best method is to convert both lines into a polyline by pressing the *polyline* button . We select with the mouse the lower line and with the command *line* we introduce the number corresponding to the upper one (53).

If a message appears saying *not correct polyline*, the probable cause is that when doing the copy, both lines were not in contact at their ends. This happens often and is solved with the *Join Lines* *end Point* command(as explained above). For this reason you should be especially careful when selecting the lines to join. The lower one can be selected with the mouse, but the upper one has to be selected with the command line with its corresponding number (53). If the previously mentioned window appears, it means that the only problem was that they were separated. Therefore, we must press "*yes*" and make the conversion to "polyline"

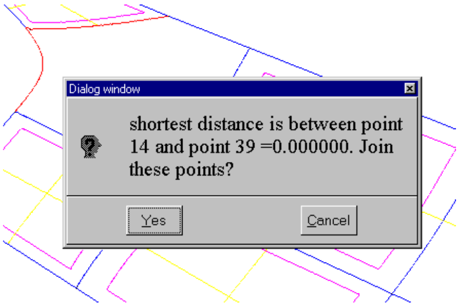


Figure 13

Once we have the polyline we can create the surface. Again, problems can appear with the lines whose endings are not in contact. This is solved once again with the *Join Lines end Point* command.

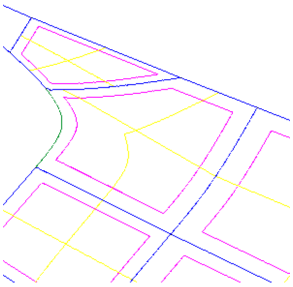


Figure 14

Error IV: Panels are limited by three lines, instead of the four needed.

Cause: Due to an especially complicated geometry (bow, stern, etc.), some zones are difficult to define with four lines.

Solution: We must divide one of the lines to solve the problem. This way, we are able to create four sided panels. It does not matter if the line to be divided is not curved or has something similar to a vertex.

In our example, this happens for the aft panel and in one of the stems. Let us study with detail the fore one.

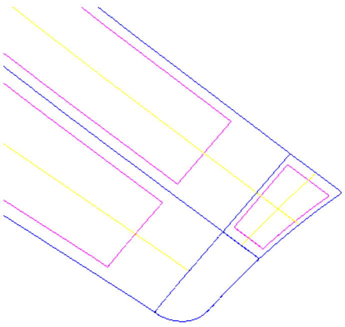


Figure 15

The line that we have to divide is the only one not bounding any surfaces (defined by red lines). Lines that bound several surfaces should not be divided as they form the boundary between them. We use *Divide>Near* Point function to divide the line.

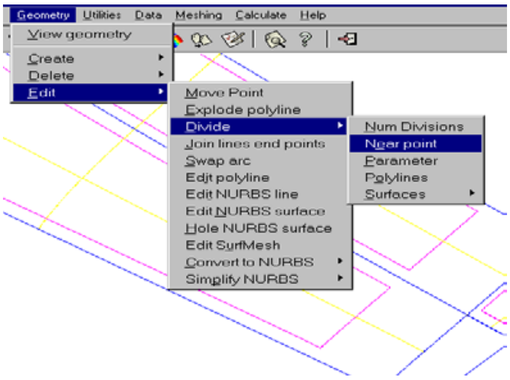


Figure 16

To define the chosen point for the division, the easiest thing to do is to select the line that we want to divide, right click the mouse and in the contextual menu choose *Point in Line*. Thus, if we do not indicate the zone of division very accurately, the program will choose the point on the line closest to the centre of the mouse pointer.

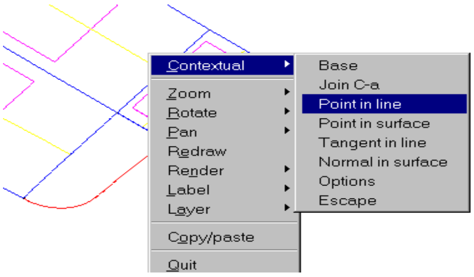


Figure 17

We should choose the point that makes the panel as quadrangular as possible. This way, the mesh will have a more uniform look. In the following figure, the segment where the line was divided is indicated in red.

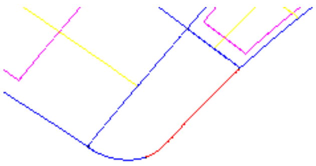


Figure 18

Doing the same thing with the stern one, we finish the process of covering the hull with surfaces.

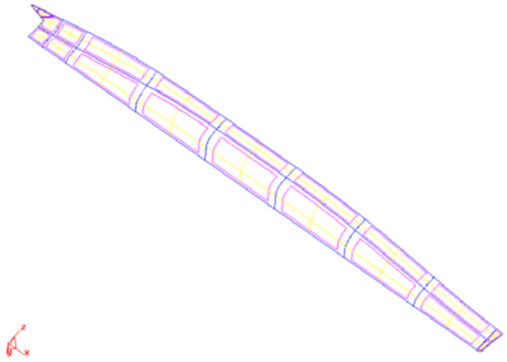


Figure 19

Let us now proceed with the meshing of the hull.

In the *Meshing* menu, go to *Structured*→*Surface* then select the whole hull.

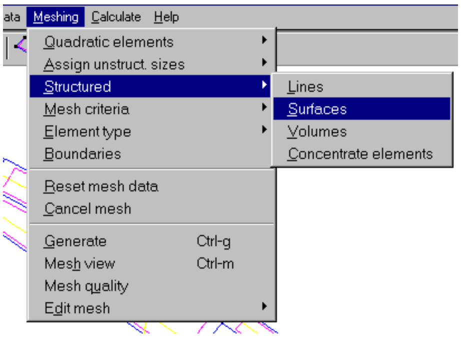


Figure 20

Press *ESC* and a window should appear to request a number of divisions for the surface edges. We will try first with 5. We press *OK* and then select the line for these 5 divisions. For example, we can select the edge of any surface and see what happens. Here, we have selected one of the waterlines in the central section of the hull. Since the mesh has to be continuous, we can see the horizontal lines of the surface get automatically highlighted. A surface will have the same number of cells in the horizontal/vertical direction as its neighbour. Although this is the usual way of creating the mesh, it has to be modified in some instances, as will be seen later.

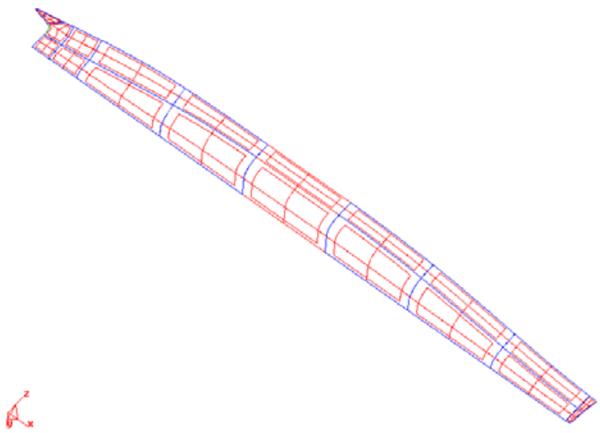


Figure 21

Let us press *ESC* and Cancel the operation. We have now assigned to these surfaces the proper number of cells in the desired direction. We can foresee that with the others the same thing will have to be done. First, we press *Generate* within the *Meshing* menu. In the window that appears there is no need to change anything since the size should be assigned automatically until the surface is complete.

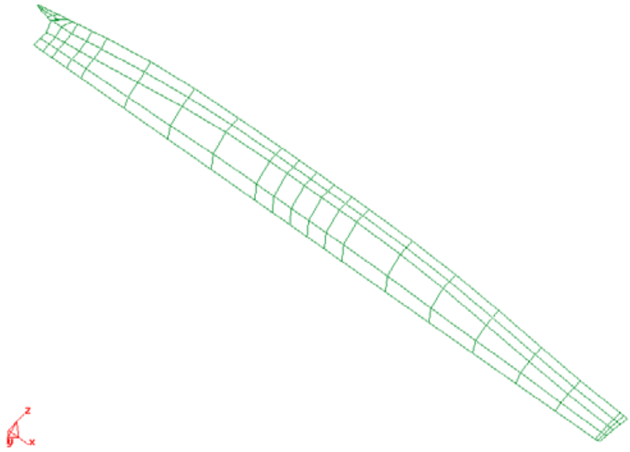


Figure 22

We already have our first mesh but the definition is not good enough. In this mesh it is possible to see how in the central area (the five panels previously assigned) appear. Repeating the previous operation with all the panels we are able to refine the mesh.

For example we select again the whole hull with *Structured>Surface* in the *Meshing* menu. Pressing *ESC*, we introduce 7 as the number of segments per line, then *Ok* again and finally, highlighting all the horizontal lines of the hull to mean that for each horizontal line there must be 7 panels in the mesh. We then press *ESC, Cancel, Generate* and verify the results.

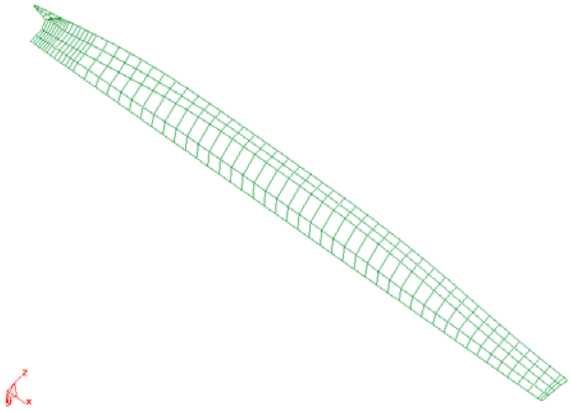


Figure 23

We can see that for the smallest panels (stern no seria bow ???) the concentration of panels is greater, as expected. Once this mechanism is properly understood, we can refine the mesh to a convenient level and start varying the number of rows of cells in the vertical direction until we consider it sufficient. Again we select the whole hull and assign, for example, the value of 9 panels per line. In this case vertical lines have to be selected – by highlighting one, the contiguous ones are highlighted automatically -. The new mesh should look better:

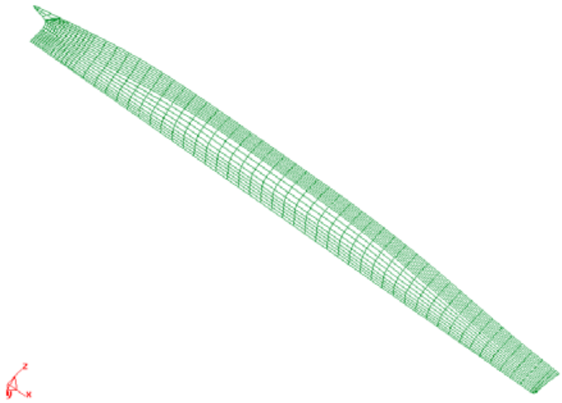


Figure 24

In this initial mesh, in comparison with the upper part of the hull, the bottom part is less defined as far as number of cells in the vertical direction is concerned. In order to homogenize the mesh it is necessary to change the number of cells in the vertical direction of the lower vertical lines. We increase it, from 9 to 14 and regenerate the mesh.

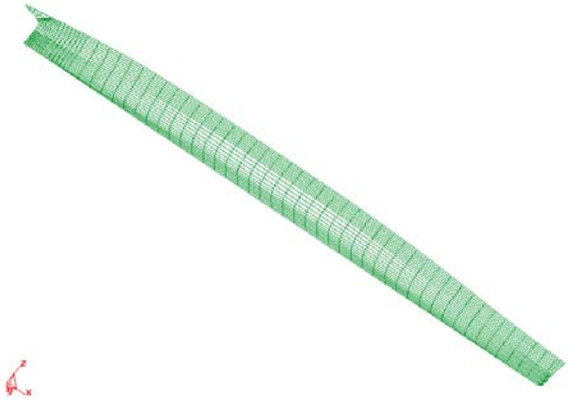


Figure 25

The mesh now looks more homogenous and we are getting closer to a decent result.

We should keep in mind that our mesh must meet certain requirements: It has to have a high density zone close to the bow since it is in this zone where the potential flow is best calculated from the non viscous hypothesis. For this reason, we will pay special attention to this area. When we have a bulbous bow, we have to be more careful since acceptable results highly depend on a good mesh.

In the intermediate and stern areas, the convergence of the calculated potential flow with the real one is usually poorer, there is therefore no need for the mesh to be so dense. Having a very dense mesh makes calculation times higher and memory restrictions apply. If we exceed the capacity of the ram memory on our computer it will be impossible to make the calculation. A rough estimation of the necessary ram memory needed for the calculations can be obtained with the formula RAM $\cong 1.2\ 10^{-5}\ N^2$ (MB). With *N* being the number of unknowns (nº of panels of half hull and half free surface panelization). As an example, computers with 256 Mb RAM can solve 5000 panel problems.

With these principles in mind, we can efficiently refine the mesh. First we are going to improve the mesh in the aft part. With the *Structured>Surface* function we select the surfaces in that zone and change the number of panels in the horizontal and vertical directions until an acceptable mesh is obtained. One of them could be the following - 5 in the X direction, bow-stern, and 4 in the Z direction.

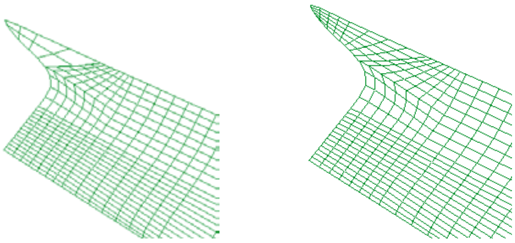


Figure 26

We now repeat the same optimisation operation for the bow zone, and reduce the concentration of panels for the stern and central zones. Once the density of panels in the bow zone is properly increased, we should have a mesh ready for computations.

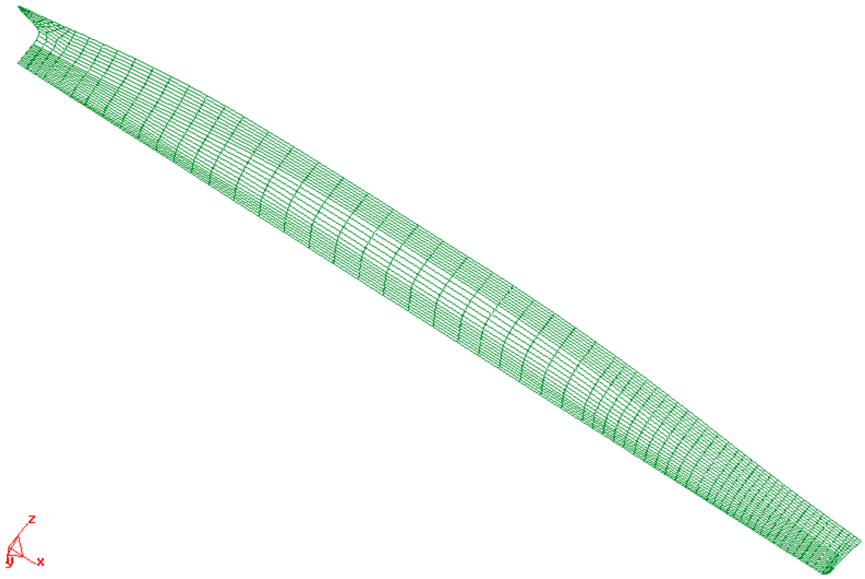


Figure 27

Once the hull mesh is completed[4] we go on to create the free surface one. To do this, we will create another layer in which we will draw the free surface. Once the layer is created and activated for use *View>Layer>To use>*name of the layer - we copy the waterline to the free surface layer.

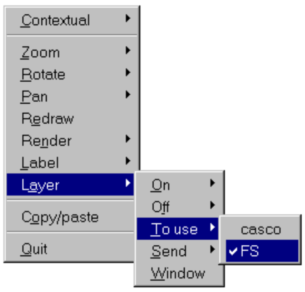


Figure 28

[4] Example file: S60_3.gid

In order to copy the waterline we must have the two layers in on state and the layer of the free surface activated - *To use*. The next step is to copy the waterline to the new layer using the *copy*  command. Within this command context, select lines as the entity type. We then click *select* and select the lines that form the waterline of our hull within the drawing. We do this in order to copy the waterline to the layer in use, the free surface one.

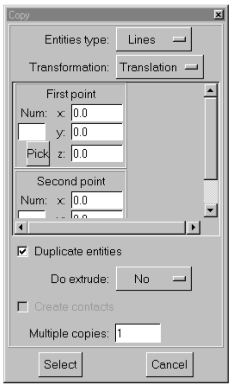


Figure 29

If we now deactivate the layer where the hull drawing is, only the waterline should appear on the screen.

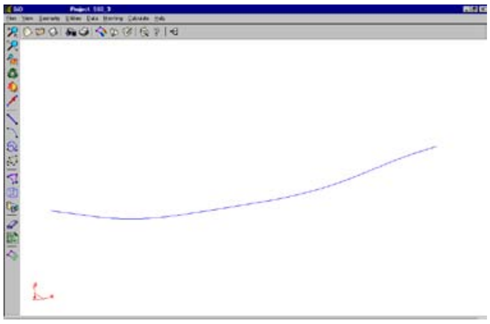


Figure 30

The following step is to create the limits of the free surface (the line we have just created will form part of this limit or contour). It should look something like *Figure 31*:

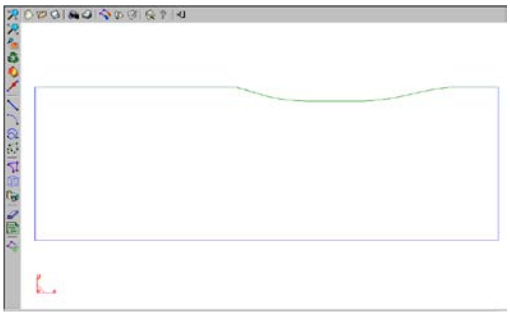


Figure 31

Usually, the fore line is ¼ LBP forward of the bow and the aft line is LBP aft of the stern. Therefore, the x length of the free surface mesh is approximately 2.25 times LBP. In the transverse direction, the line should be around ¾LBP away from the centre line. Remember that all this data is a general estimation and only experience with the peculiarities of each case can suggest appropriate dimensions.

We will now create a line  defined with the command line-from the point (0,0,0) that passes through the vertexes of the free surface which we have calculated with the previous measures. This way we are defining the starboard part of the free surface line.

In this particular case the vertexes are: (0,0,0) (-121,0,0) (-121,-90,0) (151,-90,0) (151,0,0) (50,0,0).

Figure 32

With the command *Geometry>Edit>Divide>Near point* we split the horizontal line that goes inside the forward waterline end.

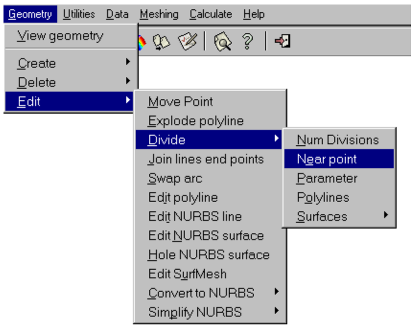


Figure 33

Once we have selected the line that we want to divide (red coloured in figure 37), we then indicate the division point by right clicking the mouse (the *Figure 34* context menu should appear).

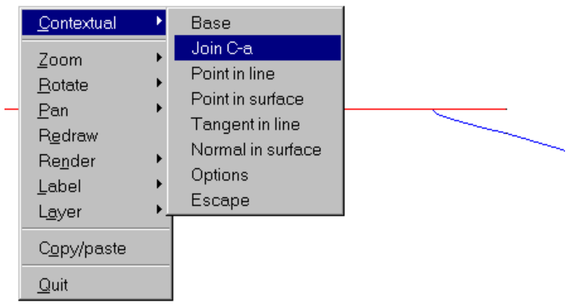


Figure 34

Select *Join C-a* is with the mouse and then select a zone close to the intersection of both lines. Once the division is made, we only have to eliminate the line piece that is outside our free surface.

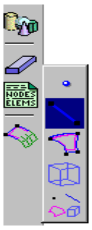


Figure 35

With the *erase line* command activated (*Figure 37*) we select the one that we want to erase and we press Esc. The same operation in the bow zone is repeated and we verify that both lines are joined in the division vertex with the *Join lines end point* command used previously. The appearance at this point should be something like:

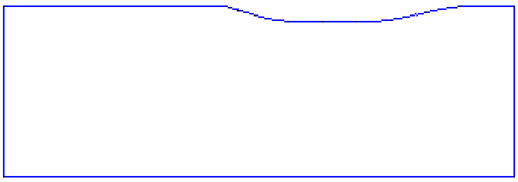


Figure 36

The last operation before creating the free surface is to convert all these line edges into one single polyline. From this polyline, a four sided surface will be created in which one of the sides will be the waterline polyline.

In order to create this polyline we use the icon and with the mouse drawa box that includes most of the lines we want to convert. The rest of them can then be selected with the mouse.

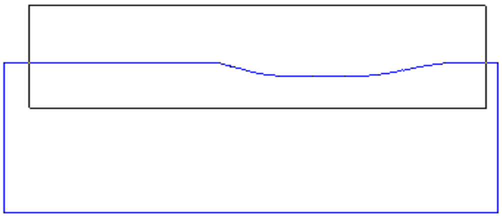


Figure 37

After these operations, a new green polyline should appear. If this is not the case, the most probable reason is that some of the lines are not properly joined and we should check them one by one. The ones that are separated have to be joined with the *Join lines end point* command.

In the end we should have four lines that form a closed perimeter, one of which contains the starboard part of the waterline. With these lines, we create a four sided surface like we did for the hull.

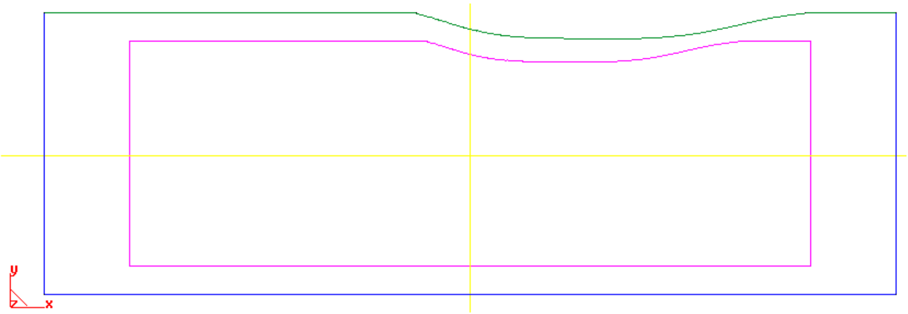


Figure 38

The mesh criteria is stricter here than for the hull, mainly regarding the minimum number of panels.

Number of panels in the longitudinal direction:

To calculate this number, we have to know the ship's speed "V" for this case. The characteristic wave length for this speed is obtained with the formula ($=2\cdot\pi\cdot V^2/g$). The minimum number of panels for each complete wave is around 15. Since we know the total length of the free surface, we can calculate the minimum number of necessary panels in the X direction. In this case, the length of the free surface is 272 m. If we suppose that the speed is 12 kn. = 6.1728 m/s this produces 24m long waves. Since we need 15 panels for every 24 meters, we need about 170 panels in the X direction.

Number of panels in the transverse direction:

To adjust the number in the Y direction we have to get an aspect ratio of the panels that is as close to 1 as possible. This is not always viable since it could lead to a too great a number of panels for our equipment's RAM memory. Therefore, this aspect ratio is considered as ideal and the real limit will always be imposed by the calculation power of our equipment.

As a final consideration, it is important to point out that in order to not exceed the memory limit in the calculations, we should reduce the number of panels on the hull. We can vary the size of the free surface, and even experiment with the mesh density in the transverse direction of the free surface. But we cannot reduce the 15 panels per wavelength for the mesh in the direction of the free surface.

For S60 case we have used 170X23 panels in the free surface. It looks like this[5] (Figure 39).

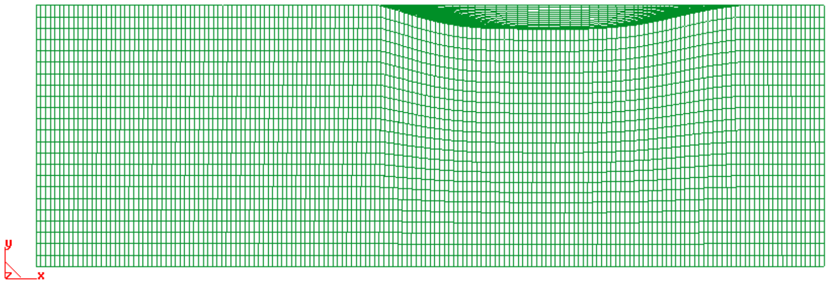


Figure 39

Special Mesh in the stern region.

[5] Example file: S60_4.gid

As we have already commented, ETSIN_CFD calculates the potential flow around symmetrical hulls and since this is a potential calculation, there is a difference between the calculation and the experimental results in the stern region. This is because in these zones, viscous effects generally dominate at low speeds and they are not considered here. This fact means that the results diverge strongly in the stern region for transom stern ships.

For dry transom stern ships, a specific module of ETSIN_CFD exists in which a different mesh type is used for the free surface and specific conditions are applied for the stern region.

In our case, we are dealing with ships without transom stern or with a wet transom stern. The latter type and some others with thicker lines in the stern region can induce bad results unless a particular mesh is used. This is due to the huge curvature of the panel lines in that region which is needed to accurately follow the transom waterline. In order to avoid divergences, we will have to mesh this region of the free surface in a special way. The criterion is to get a smooth union between the waterline and the centre line a little aft of the stern.

Some examples of these cases are shown in the following figures.

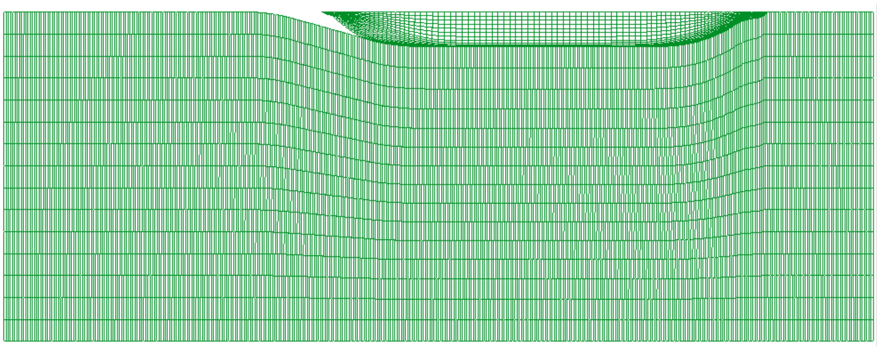


Figure 40

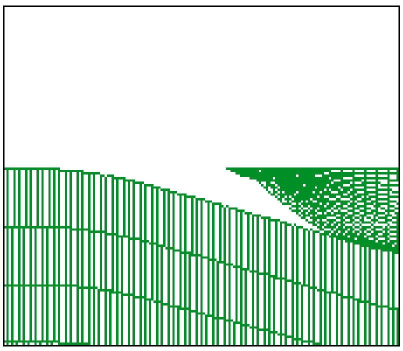


Figure 41

The mesh of figures 42 and 43 corresponds to a bulk-carrier 185 m long sailing at 9 kts. Due to such a big length and the low speed of calculation, the mesh had to be very dense. We had to shorten the free surface at the stern so as not to exceed the computer's calculation limit. As we can see, the stern zone has been smoothed at the free surface in order to avoid the divergences we have just commented on.

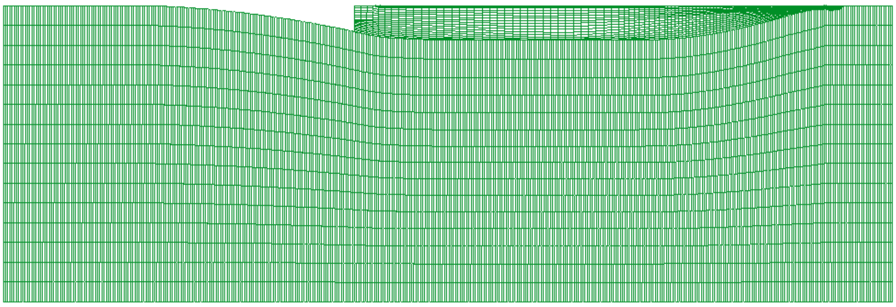


Figure 42

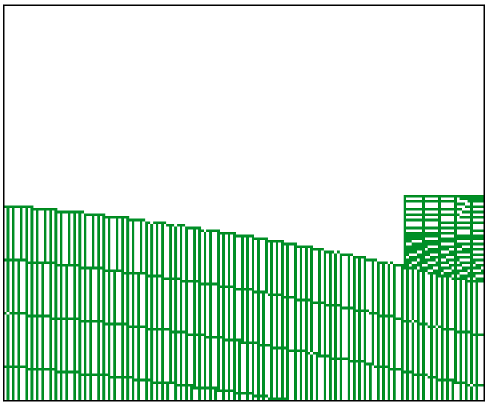


Figure 43

In figures 44 and 45, the mesh was made for a container vessel 200 meters long sailing at 9 kts. The transom stern is wet. This area was smoothed in the free surface panelization. In order to do this, the connection of the waterline was made tangential to the centre line.

Process phase: ETSIN_CFD

Once the hull has been meshed, we must proceed with the proper flow calculations. We first have to define the kind of problem that we want to solve. In our case, the *Problem Type* will be the one that we have called *ETSIN_CFD*. In the *Data* menu we choose *Problem Type*. Within this menu *ETSIN_CFD* should appear as an option. We select this type of problem and introduce the required data for the calculation. (For more details on the calculation basis see [9]).

Within the *Conditions* section of the *Data* menu, we must indicate which part is the hull and which part is the free surface so that they are separated during the calculation process. To obtain this, the first thing we must do is to deactivate one of the layers. For example, we deactivate the free surface layer and we work with the hull layer. We choose the condition *hull* (default) in the *Conditions* window. We leave *Value* as 0.0 and press the *Assign* button highlighting the entire hull. This same operation has to be done with the free surface that we will activate after deactivating the hull layer. The *Starboard flotation condition* has to be assigned to the free surface.

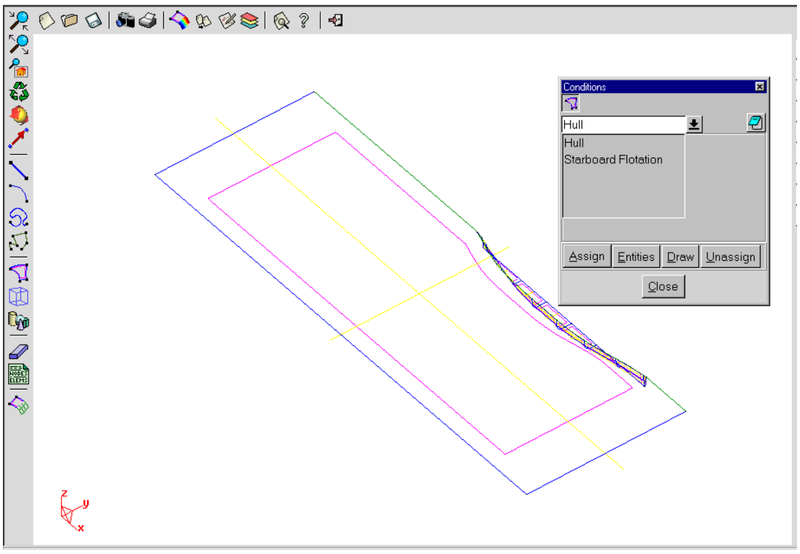


Figure 44

The next operation to be performed is to introduce the problem data. This is done with the *Problem Data* submenu within the *Data* menu. We must fill in the information required in the menu boxes.

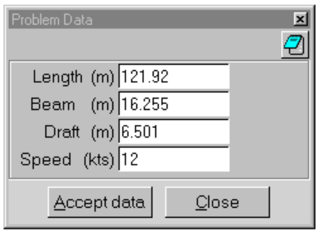
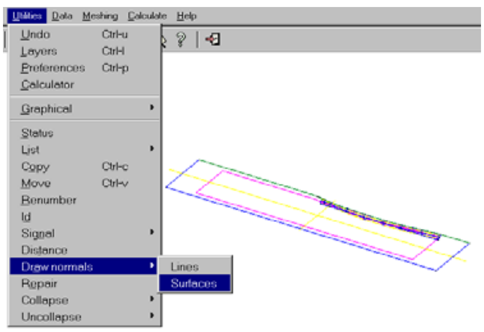


Figure 45

Another very important step is to define the direction of the normal vectors of each surface. To check these directions we select *Draw normals>Surface* within the *Utilities* menu. The normal lines of the surface will be drawn and thus we will be able to check their orientation. They should point to the inside of the hull for the hull surfaces, and they have to point down for the free surface. If any of the lines do not fulfil this condition (normally it is a random condition) we have to swap its direction. To swap it, the option SwapSome situated in the menu at the right hand side of the screen must be used.



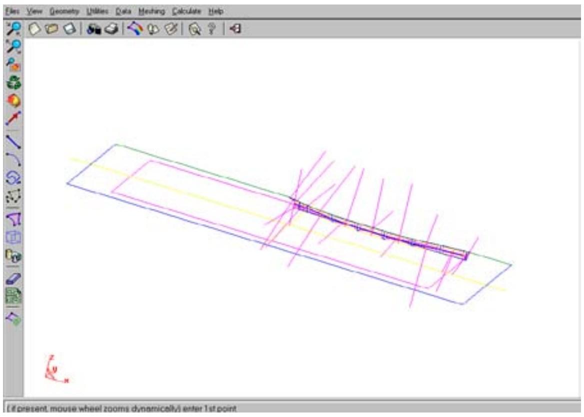


Figure 46

With this option activated we will change the normal direction of all the selected surfaces. The final distribution should now be as it appears in Figure *47*.

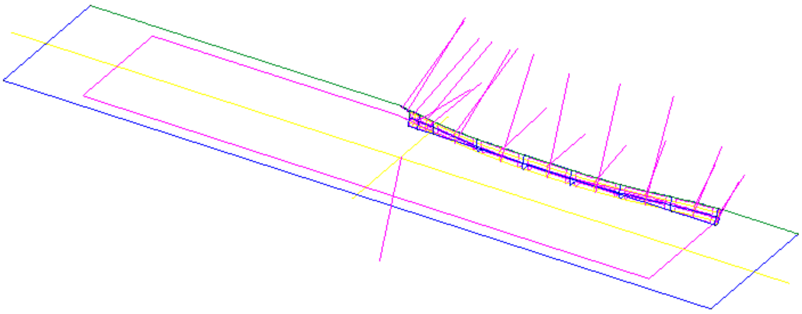


Figure 47

The last thing to do is to regenerate the mesh so that this data and the changes made can be reflected in the final calculation mesh. The *Generate* option from the *Meshing* menu asks for the size of the generated panels but it is automatic and the value we set is disregarded. The mesh will then be rebuilt with the correct orientation of the normal vectors. Now the mesh is ready for the process phase.

In order to start the calculation we must enter the *Calculate* menu. Inside we have several options. We can directly press *Calculate*. This will start the calculation and give us a message when finished. The other option is to open the window *Calculate Windows*. Within this window we will be able to launch the calculation with the *Start* button. This is the recommended procedure and the one we will follow here.

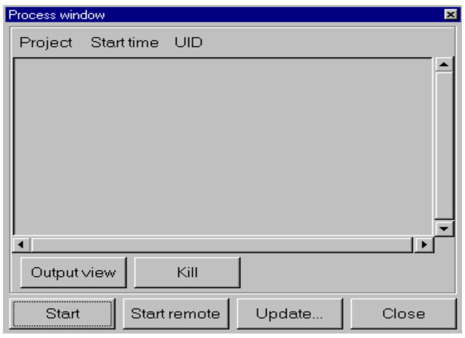


Figure 48

After starting the calculations the screen will change and the name of the current case will appear:

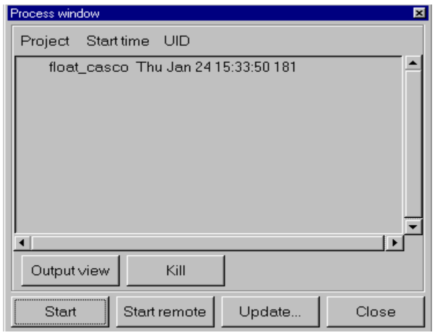


Figure 49

We have already commented on the memory necessities for the calculations. These calculations also require resources from the system and we recommend running them on a dedicated computer so that it does not slow down any other work.

Once the calculations have been performed a warning window will appear on the screen, offering us the possibility of directly accessing the postprocess phase which we should accept.

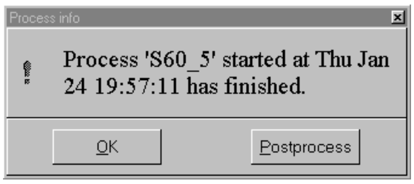


Figure 50

Postprocess

Once the calculation is finished, we can get into the postprocess context of GiD. This can be done when requested after the calculations or directly through the *Files>Postprocess* command. We have to set certain values for a correct interpretation of the results. *View Style* and *View Results* menus have to be active. For more details, consult GiD manual [2].

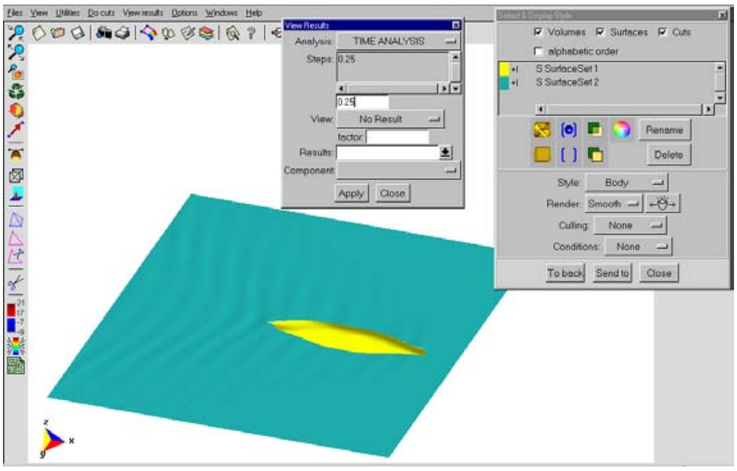


Figure 51

With *ETSIN_CFD* we have the possibility of visualising different results in GiD. The speed field can be visualised on both the hull and the free surface. Pressure Coefficients can also be visualised on the hull and the wave pattern on the free surface. In order to select one of the visualisations we will use the *View Results* menu.

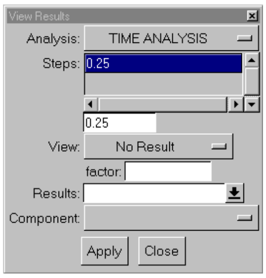


Figure 52

First we have to select the type of view. Within *view* menu, we activate *Contour Fill*.

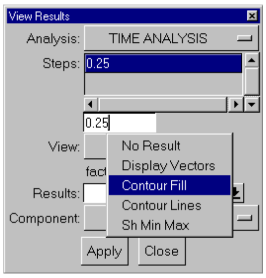


Figure 53

Within the *Results* box, pressing , we can see all the variables that can be visualised:

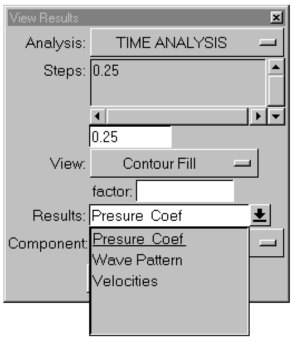


Figure 54

Visualisation of Pressure Coefficient

Pressure Coefficient (Cp): This coefficient is a non dimensional way of representing the pressure on the hull.

$$Cp = 1 - \left(\frac{V}{V_\infty} \right)^2$$

In order to visualise the pressure coefficient function on the hull we have to select it in the previous menu. After selecting it, we press *Apply*. Pressure coefficient distribution appears only on the hull, since it makes no sense to draw this pressure coefficient on the free surface, where the pressure is atmospheric. For a better visualisation we will deactivate the layer that contains the free surface. We use the *Select & Display Style* menu. *S SurfaceSet 1* corresponds to the hull mesh and *S SurfaceSet 2* to the free surface mesh. By deactivating the free surface one [ ] and pressing *Apply* in the *View Results* menu, only the hull will remain visible.

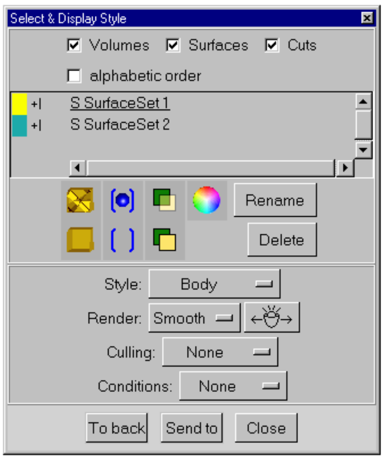


Figure 55

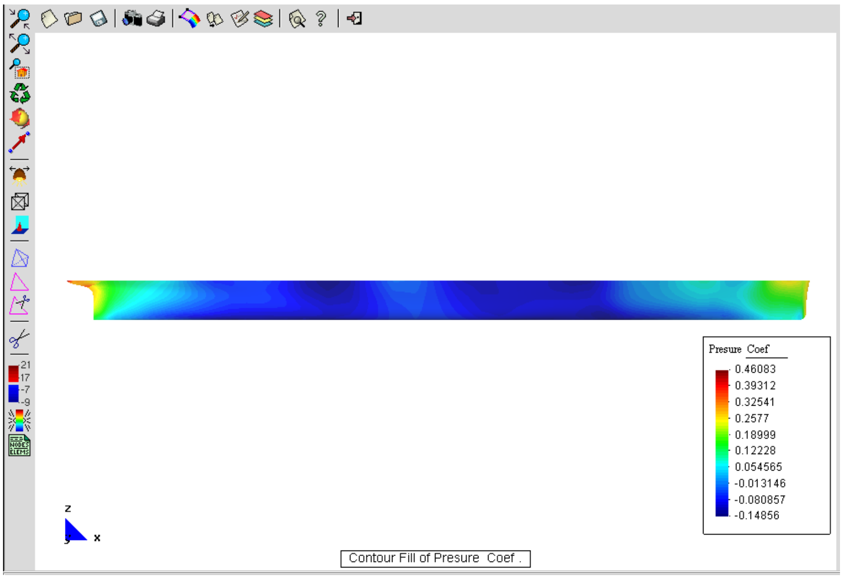


Figure 56

Visualisation of the wave pattern.

Another feasible visualisation is the wave pattern on the free surface.

Again in the *View Results* menu, with this function: we select the visualisation that we want, in this case the *wave pattern*. To visualise only the free surface we enter it in the *Select & Display Style* menu. In this menu, deactivating *S SurfaceSet1 means* only the free surface will remain visible. Do not forget to click *Apply* after every change.

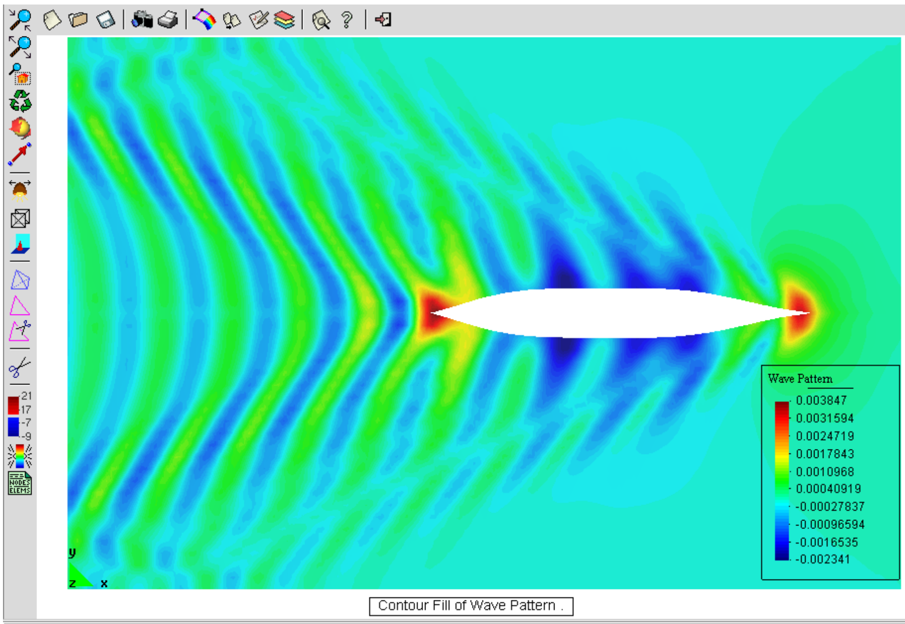


Figure 57

Velocities field visualisation.

The last visualisation option is the velocity field. This magnitude is full of physical meaning, both on the hull and on the free surface. The flow is potential and so the speed on the hull surface is not the ship's speed itself. To correctly interpret this output data, it is necessary to consider the ship still and that the water advances towards it at the ship's forward speed.

Again, we follow the same steps and visualise *Velocities.* There is also the possibility of choosing any Velocity component as well as the module.

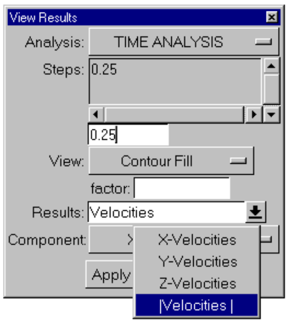


Figure 58

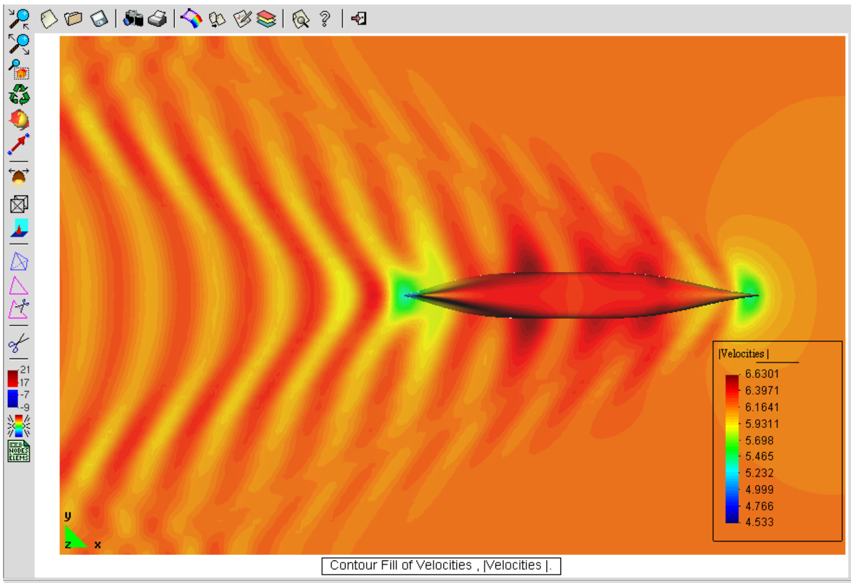


Figure 59

This representation of the free surface together with the hull is not the best one. GiD distributes the colour range automatically and this causes (because the data value limits are always on the hull) the free surface gradients to be ill defined. It is better to deactivate the hull layer when studying the free surface information.

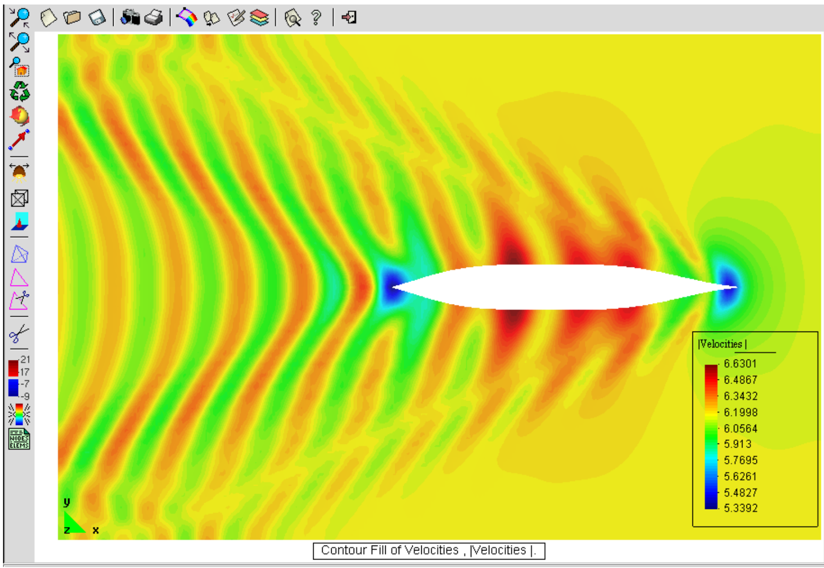


Figure 60

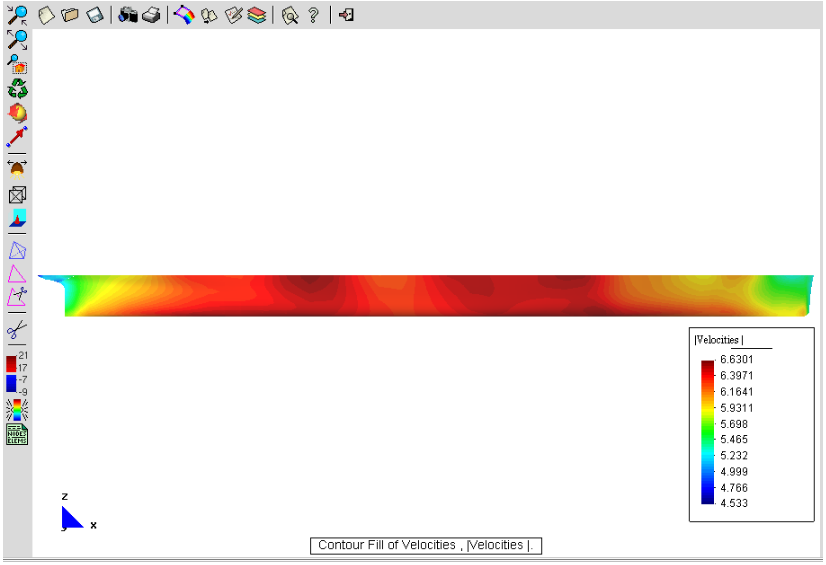


Figure 61

Longitudinal cuts.

The result files are kept in folder *results* within the folder *casename.gid* where the mesh used for the calculation was saved. This folder also contains the ascii files long_cut_1.5508B.dat, long_cut_1.0587B.dat and long_cut_0.5665B.dat*.* In these files there are the curves corresponding to longitudinal sections of the free surface at 0.5665B[6], 1.0587B and 1.5508B. With these graphs we can visualise the height of the generated waves and other characteristics. This will be one of the most useful tools when optimising the ship's hull, as will be discussed later. They can easily be imported from Excel, for instance.

For the S60 example used in the tutorial, the graphs are shown in the following figures.

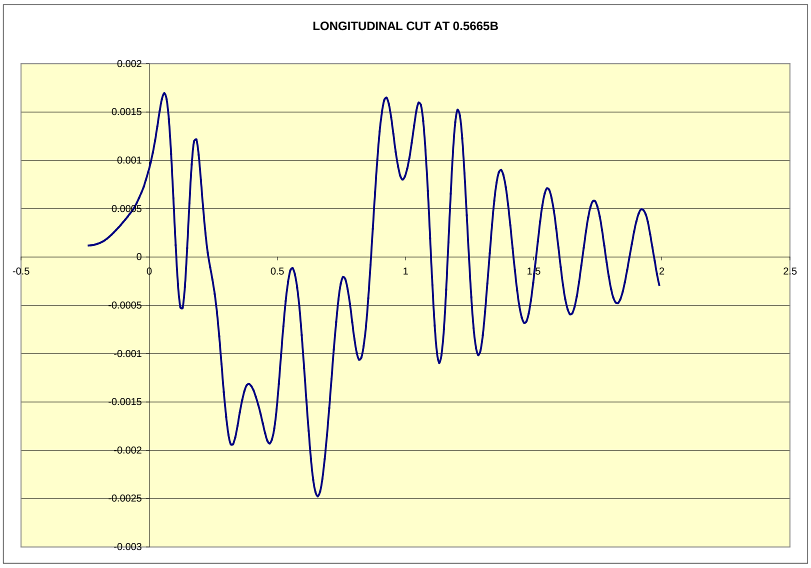


Figure 62

[6] B= ship's beam

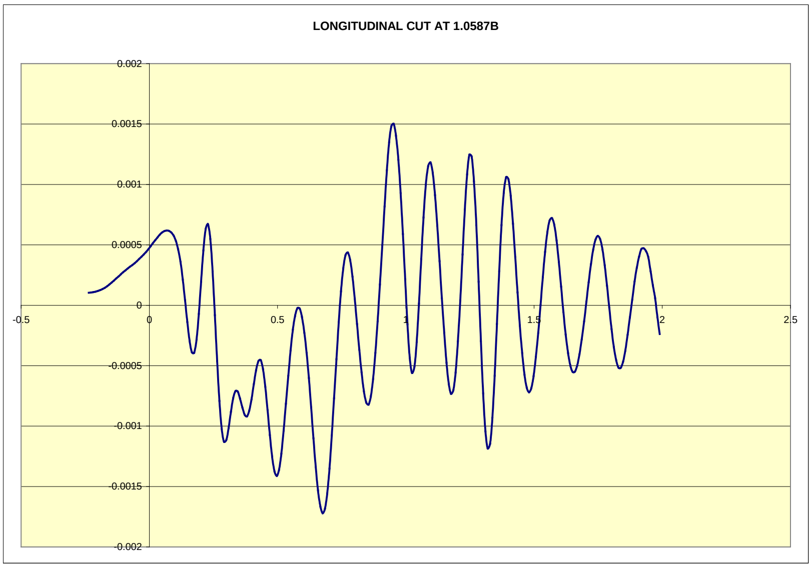


Figure 63

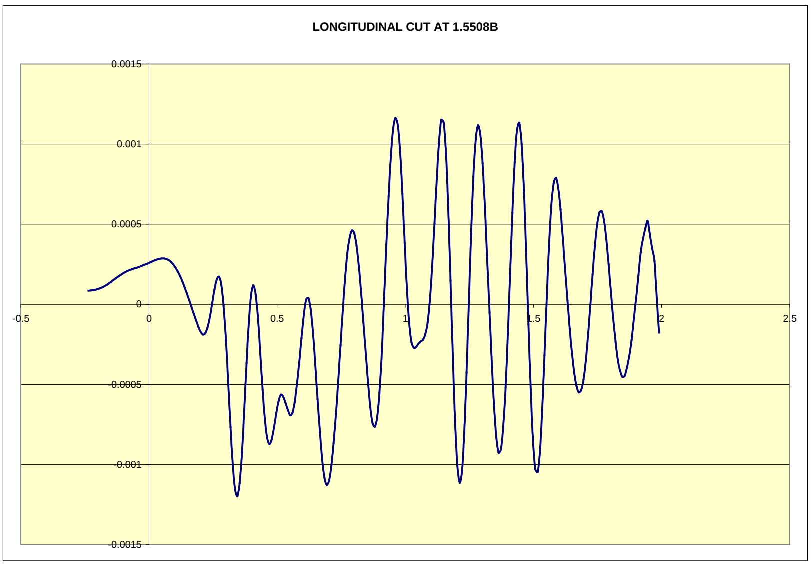


Figure 64

These are non dimensional graphs. Along the X-axis, the free surface X-component is represented. It is in a non dimensional form with LBP as the reference length. The orientation is the common one. Therefore, x=0 corresponds to the ship's bow and x=1 corresponds to the stern. Along the vertical axis we have the free surface elevation in non dimensional form, taking again LBP as the reference length. So, if we want to know the proper value of the height of some of the waves, we must multiply its graphed value by LBP.

Tutorial. Example 2: Fishing Vessel with bulbous bow.

In order to illustrate the calculation process of a ship with bulbous bow, we are going to use an optimised fishing boat. This optimisation process is described in one of the following sections.

We have already explained the first example, commenting on almost every detail. We will now use that example to concentrate on the new features that a case with bulbous bow implies.

Preprocess

Again we need a CAD definition – usually in IGES or DXF format – describing the starboard side of the submerged part of the hull.

Grid generation with GiD

Once we have imported the IGES[7] file we must save it in casename.gid. We are now ready to start the grid generation process.

There is no difference between the phase of preparation of the hull panels and the S60 example. The same errors usually appear and they are overcome in an analogous way. We will just comment on the specific problems that arise from the existence of a bulbous bow.

Stern region:

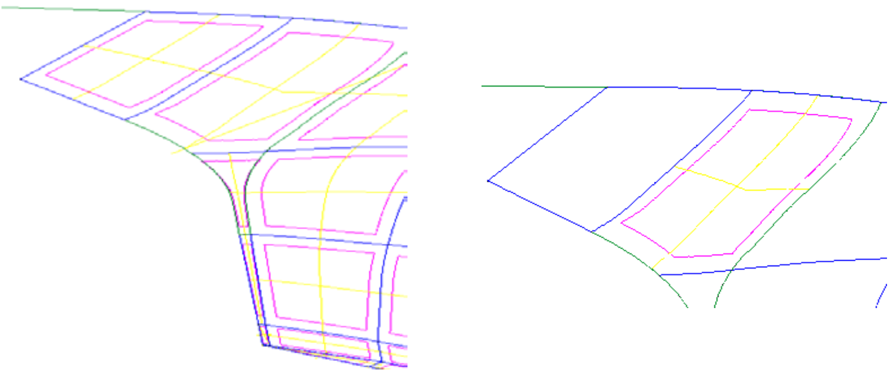


Figure 65

Several surface patches can be problematic. The second patch from the left in figure *65* seems to be bounded by five sides. This is not really the case since the solution was to duplicate the lower line shared with the lower panel. This duplicated line together with the following segment was

[7] Example file: MOTOP.igs

made into a polyline to bind the patch that we wanted to form. Thus, the lower edge line of this patch is a polyline and therefore the panel is bounded by four entities.

Another complicated patch is the one right at the beginning of the sternpost - the one that links the keel with the beginning of the sternpost.

Figure 66

This patch falls under the same category as the one bounded by three lines. We saw that the solution was to split one of them in two and to create the surface with those four lines.

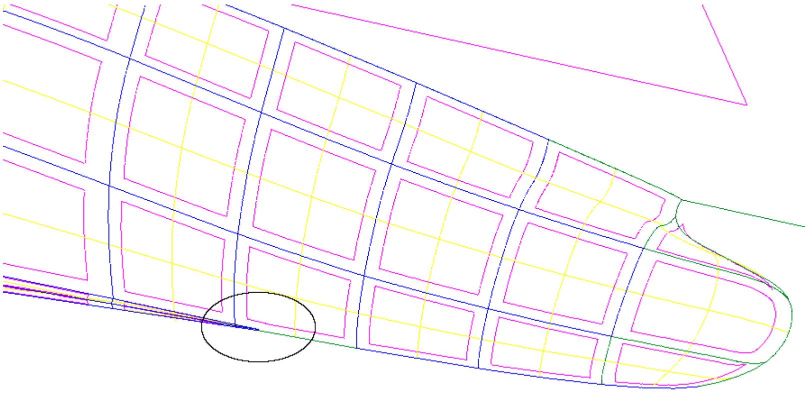


Figure 67

Other problematic patches are those at the end of the keel. Some of them are defined by three lines instead of the four necessary. Again this is solved by dividing one of the lines and generating those patches from four lines.

The transom stern region in the free surface panelization has already been commented on in the previous example. As a reminder, we said that the solution for the cases of wet transom was to model a free surface that does not consider the zone of the transom stern. What has to be done in this case is to extend the waterline in order to smoothly join it to the centre line.

In order to create the free surface patch, some of the ideas taken into account in the previous section were used. A possible free surface would be formed by the waterline and the segments between the points (LBP,0,0) and (5LBP/4,0,0), (5LBP/4,0,0) and (5LBP/4,-LBP,0), (5LBP/4,-LBP,0) and (LBP,-LBP,0), (-LBP,-LBP,0). The last segment will be made between (-LBP,0,0) and the end of the segment we created to join the water line with the centre line, skipping the transom.

For the creation of the line that skips the transom, we will work in the layer where we have previously copied the waterline. In this layer we will create a line that joins the water line to the centre line.

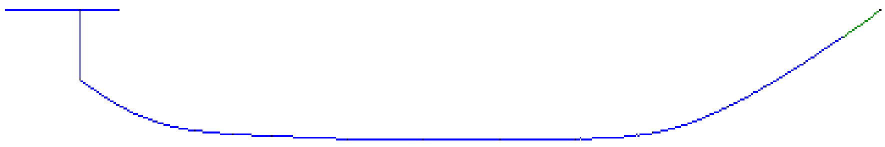


Figure 68

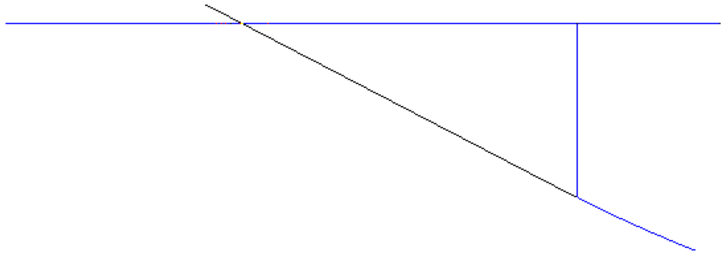


Figure 69

The following step will be to split both lines and to join them at the intersection point. To do this we use the *divide* command as already explained. We will also eliminate the unnecessary transom line.

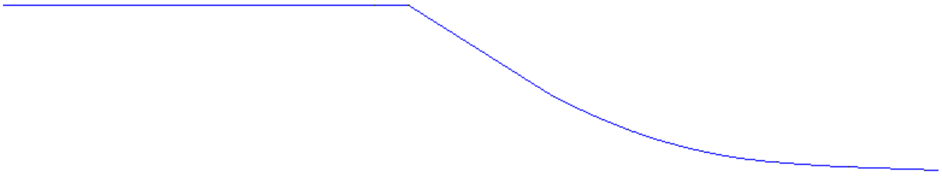


Figure 70

In order to obtain a smooth junction with the centre line we will use the command *Geometry>Create>Arc tangents*. This command asks for the radius of the connecting arc. We will continue trying until obtaining a smooth join. In this case a value of 2 was used.

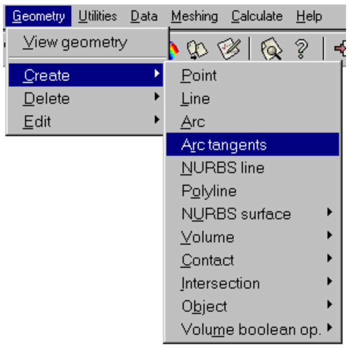


Figure 71

The final result has to be similar to what is shown in the following figure.

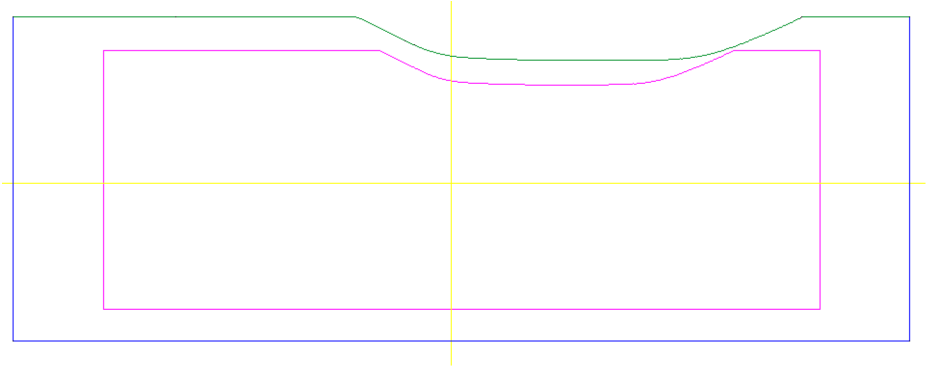


Figure 72

In order to define the panel number we will calculate the reference wavelength ($\lambda = 2\pi v^2/g$). With this value we can calculate the number of panels in the longitudinal direction. In order to decide the adequate number of panels in the transverse direction we will have to take into account the RAM memory of our computer. We set the panel number as to not surpass the RAM capabilities.

In the hull mesh the bulbous bow region has to be carefully considered. In this region a dense and uniform grid has to be created like the rest of the forward region of the hull.

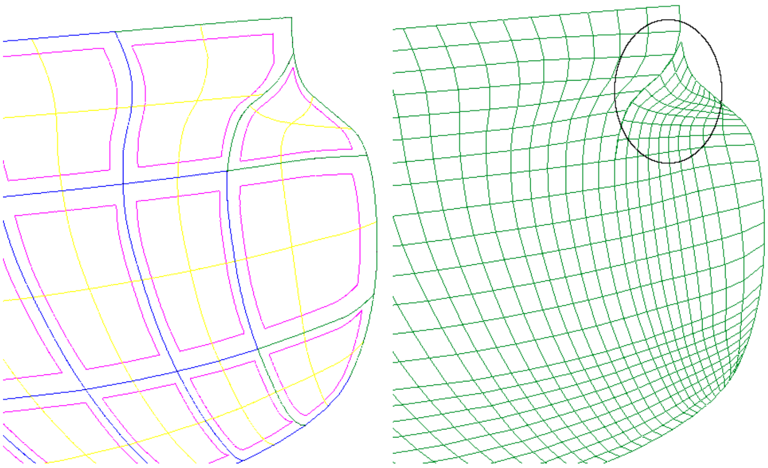


Figure 73

Considering the bulbous bow, an especially interesting area to mesh is the one inside the circle in figure 73. In this region, the boundary of the two neighbouring panels is not the same. Each one is bounded by different lines. The problem is that the upper patch of the bulbous bow does not contain the same number of divisions in the vertical direction because it is shorter than the one to the left, regarding the vertical direction. The problem arises when selecting the row of upper patches to mesh. A solution to avoid these situations is to consider these lines during the previous modelling with the CAD program. We can select the section that starts the bulbous bow as the one corresponding to the forward point of the waterline. Since this is not always feasible we explain the solution used in this example. Through trial and error, we mesh the bulb patch until there is sufficient continuity with the neighbouring patches. We are referring to the number of divisions per unit length.

We will now show different views of the mesh in order to appreciate the details of some of the complicated regions.

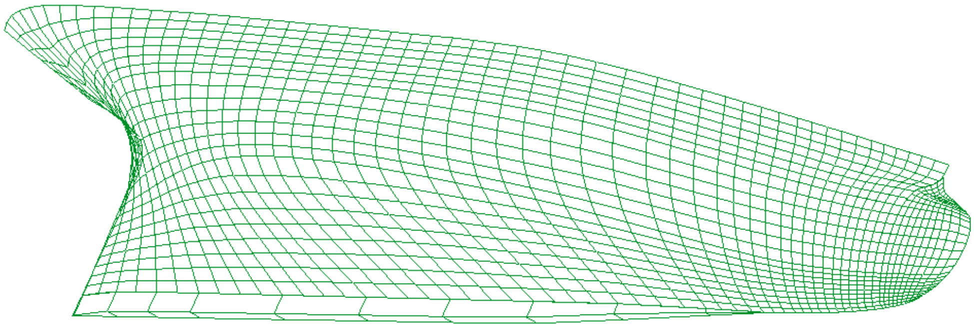


Figure 74

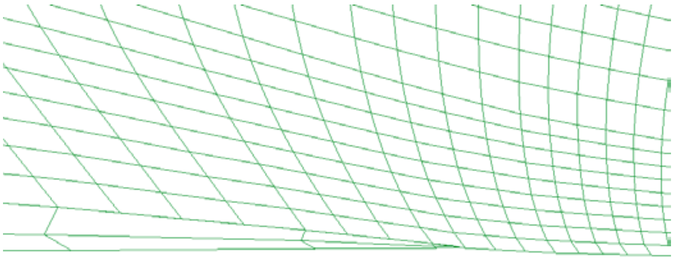


Figure 75

In Figures 74 and 75 we observe the intersection of the keel with the hull.

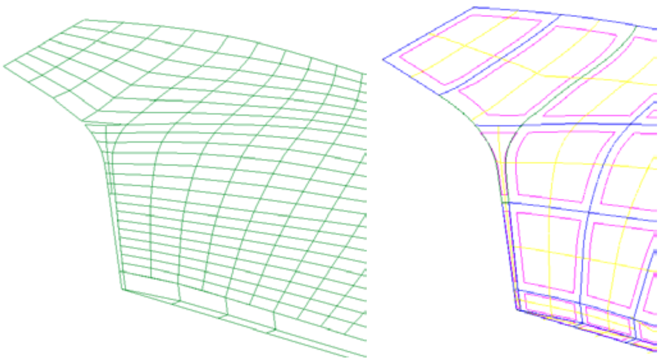


Figure 76

In the stern region special attention must be given to the sternpost. The situation is similar to the bulbous bow with neighbouring patches bounded by different length edges. We proceed in the same way as in the bulbous bow. In this case, to refine the mesh, we have sharpened the sternpost's upper patch in order to improve the smoothness of that region.

Another interesting aspect is that patches that are flat or almost flat, like the ones on the keel, and some of the sternpost, can be meshed with a lower density than those of the hull.

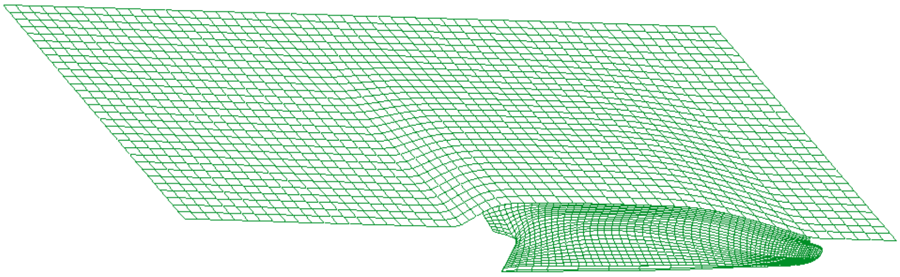


Figure 77

This is one of the many possible meshes[8] to make the calculations that we describe next.

[8] Example file: MOTOP.gid

Calculation Process: ETSIN_CFD.

The first thing to do, as previously discussed, is to define the kind of problem, *Problem Type*, within the *Data* menu. In our case we will select *ETSIN_CFD*.

The following operation will be to assign conditions to the different elements of the mesh, basically hull and free surface. Within the *Data* menu the *Conditions* submenu is opened and the *Hull* Condition is assigned to the hull (condition *Hull* if we show conditions with the *Draw* command). The Starboard Flotation condition is assigned to the free surface (condition *Starboard Flotation* if we show conditions with the *Draw* command.)

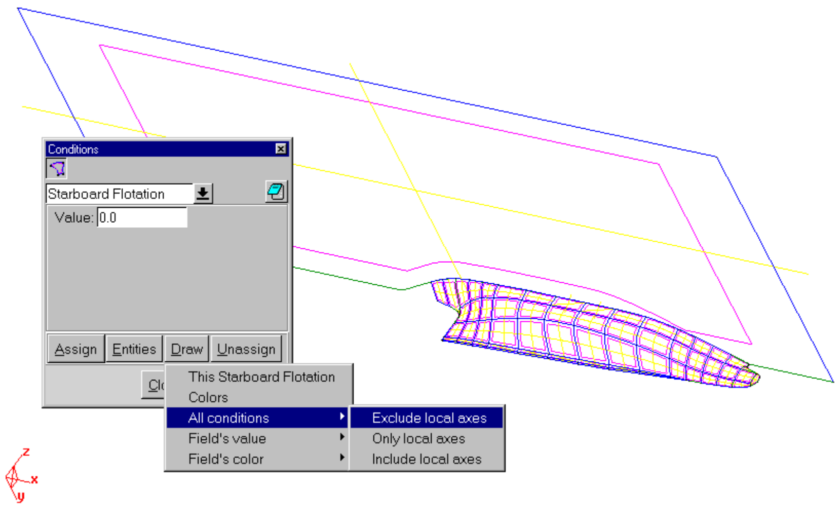


Figure 78

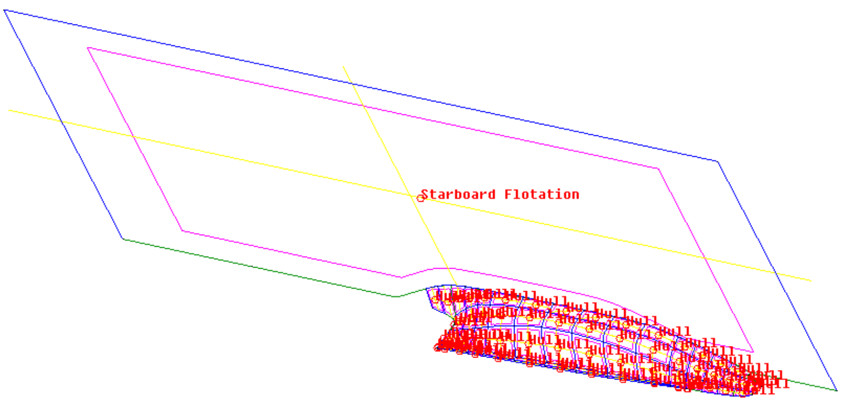


Figure 79

Another form to fill within the *Data* menu will be the one corresponding to the principal dimensions of the problem, *Problem Data*.

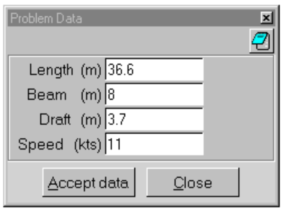


Figure 80

We now draw the normal vectors of the patches and swap with the *SwapSome* command those that are not oriented towards the inside part of the hull, and set the normal vector of the free surface oriented downwards.

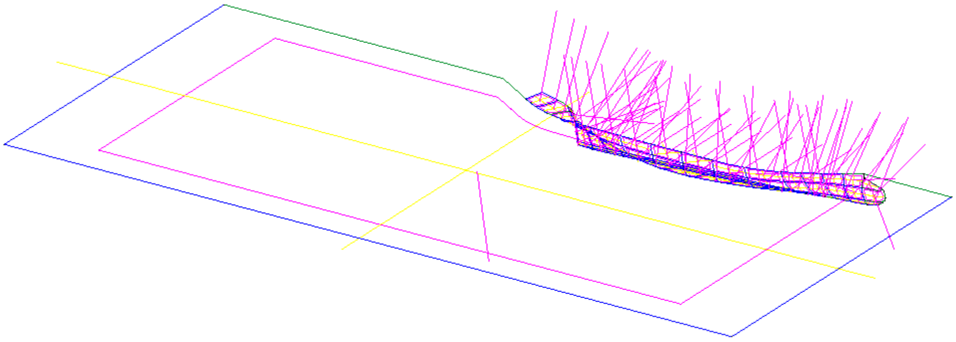


Figure 81

We will regenerate the mesh (*Generate* within the *Meshing* menu) so that the new mesh reflects the changes done so far.

We are now ready to launch the calculation process. Within the *Calculate* menu we unfold the *Calculate Window* submenu and press *Start* to begin the calculation.

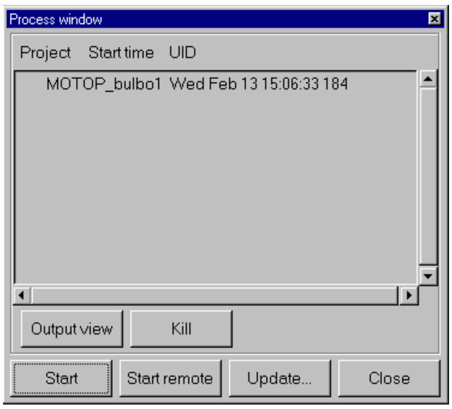


Figure 82

Once the calculation is finished we select Postprocess to start the

postprocessing phase.

Postprocess

We can start the postprocess phase directly from the final screen of calculation or with the Postprocess icon.

If we do it this way, we will have to open the file where the results are automatically saved after the calculation. This file will be in the directory *casename.gid* where *casename* is the name that we gave at the beginning of the case study. Within this folder we will find a file called *casename.flavia.res* containing the data for the visualization of the results.

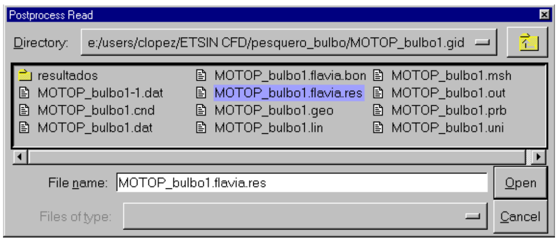


Figure 83

Once opened we can visualize the different results offered by the code.

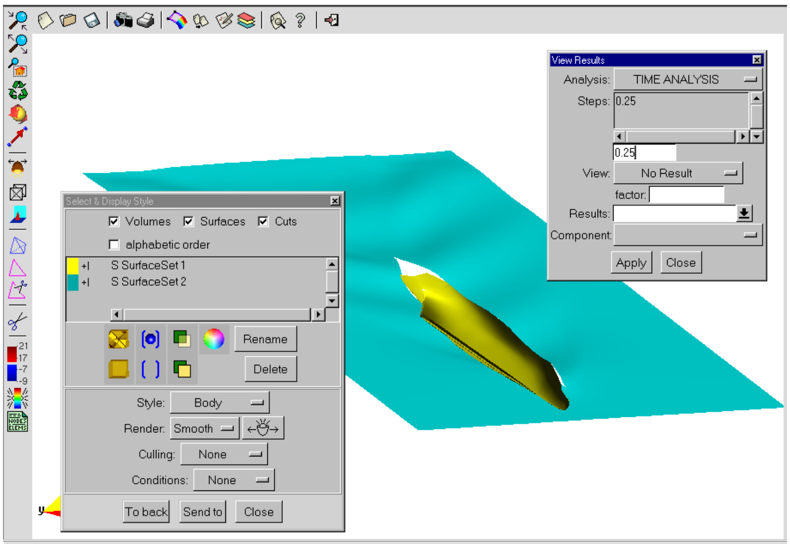


Figure 84

To start, we select *Contour Fill* in the window indicated in the following figure.

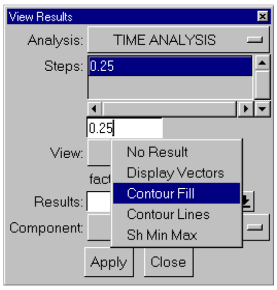


Figure 85

Pressure Coefficients Visualization.

We select the corresponding option in the *View Results* window and deactivate the free surface layer in the *Select & Display Style* window.

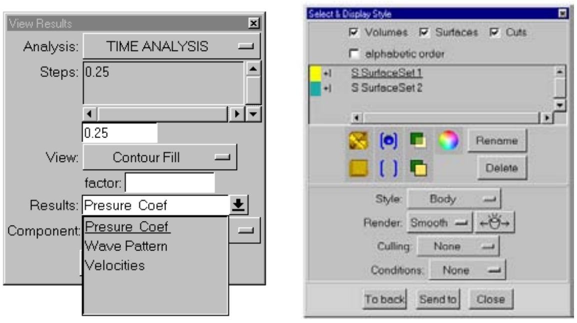


Figure 86

The pressure coefficients graph should look like this:

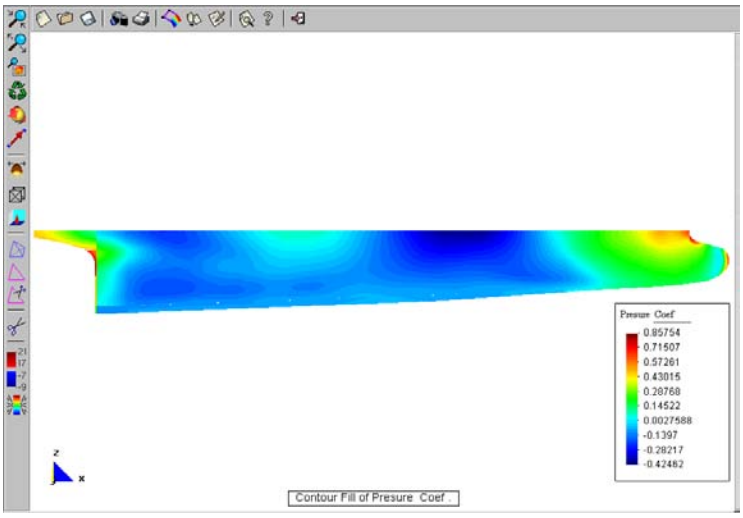


Figure 87

Wave pattern visualization.

We follow the same steps by choosing *Wave pattern*, deactivating the hull layer and activating the free surface layer.

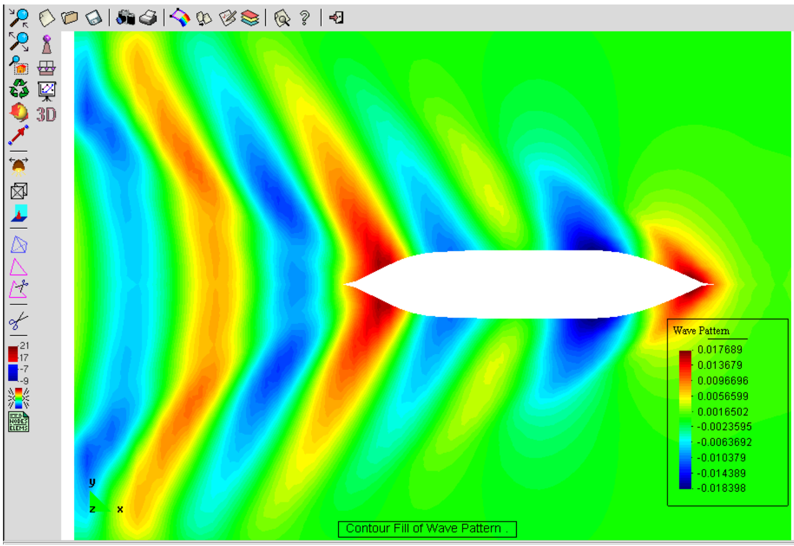


Figure 88

Velocity Field Visualization.

We choose *Velocities* in *Results* box and *[Velocities]* in the *Component* combo box. With the hull and free surface we get the following graph.

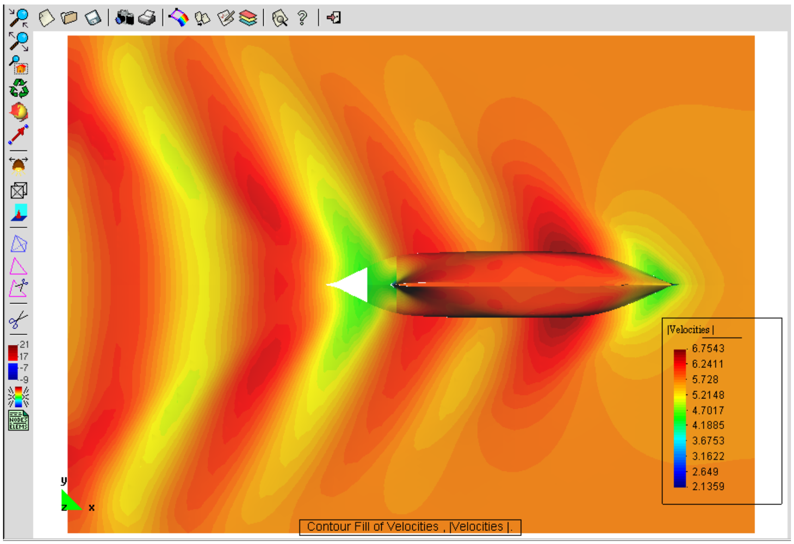


Figure 89

As with the S60, we separate the visualization to see both the hull and the free surface. We get this with the *Select & Display Style* window activating and deactivating *S Surfaceset 1* and *S Surface 2* corresponding to the hull and free surface. If we select *S Surface 1* and press *Off* only the free surface will remain visible. We must not forget to click the *Apply* button after any change to update all the colour scales.

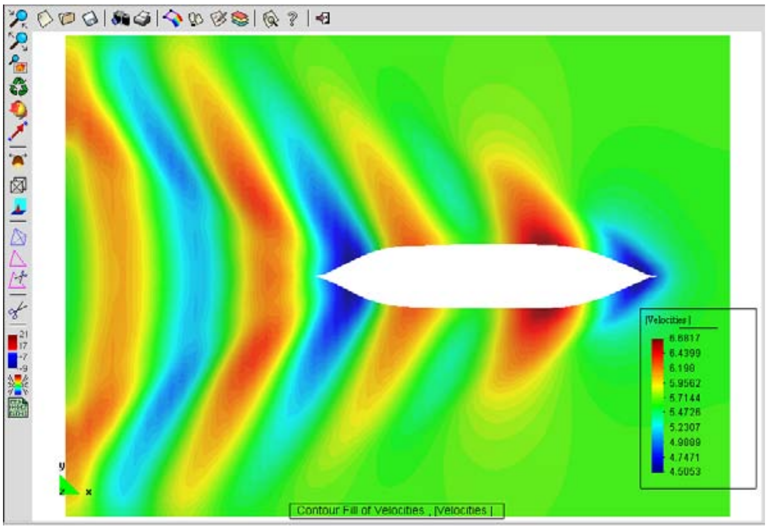


Figure 90

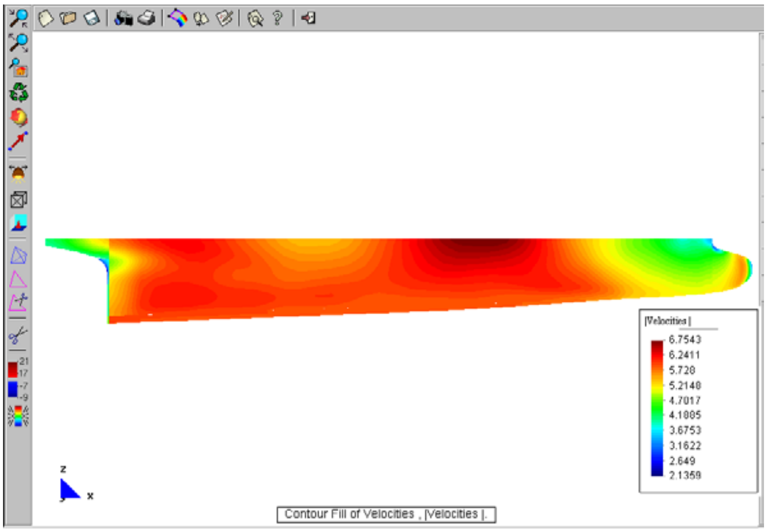


Figure 91

Longitudinal Cuts.

The results files are kept in the folder *results* within the folder *casename.gid* where the mesh used for the calculation was saved. This folder also contains the ascii files long_cut_1.5508B.dat, long_cut_1.0587B.dat and long_cut_0.5665B.dat. Inside these files are the curves corresponding to the longitudinal sections of the free surface at 0.5665B[9], 1.0587B and 1.5508B. With these graphs we can visualise the height of the generated waves and their other characteristics. This is one of the most useful tools when optimising the ship's hull, as will be discussed later. They(other characteristics??) can easily be imported from Excel, for instance. They are represented in the following figures:

[9] B= ship's beam

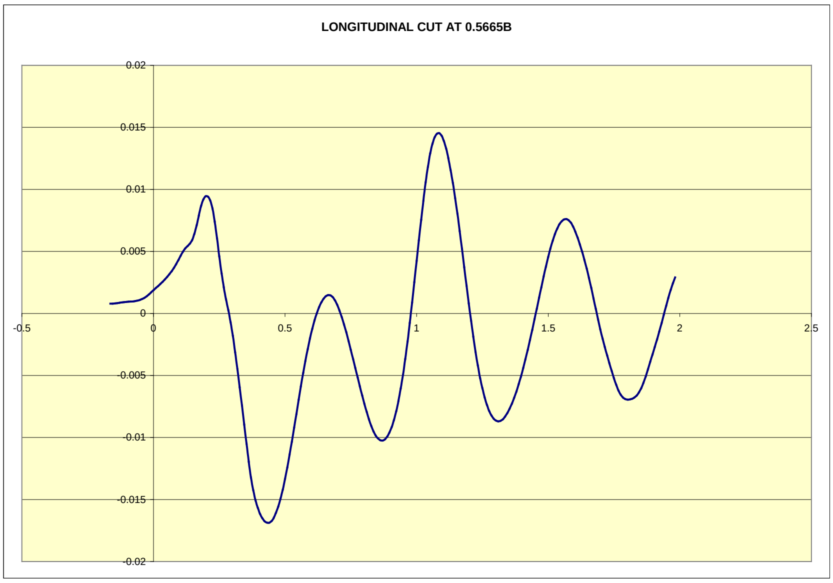


Figure 92

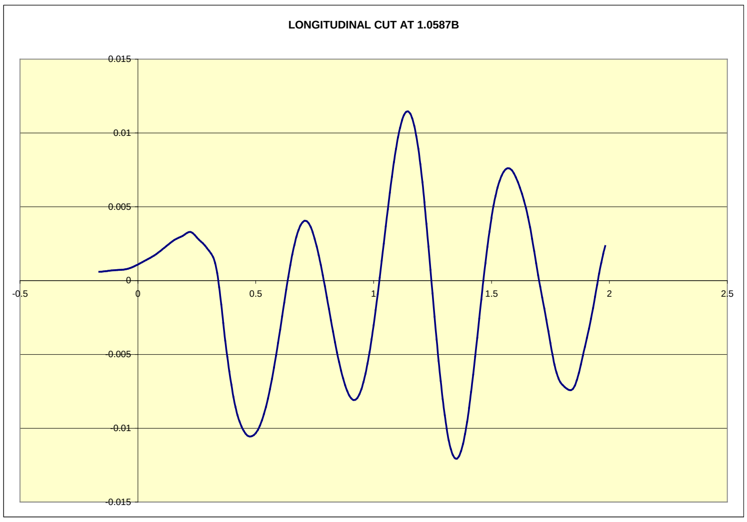


Figure 93

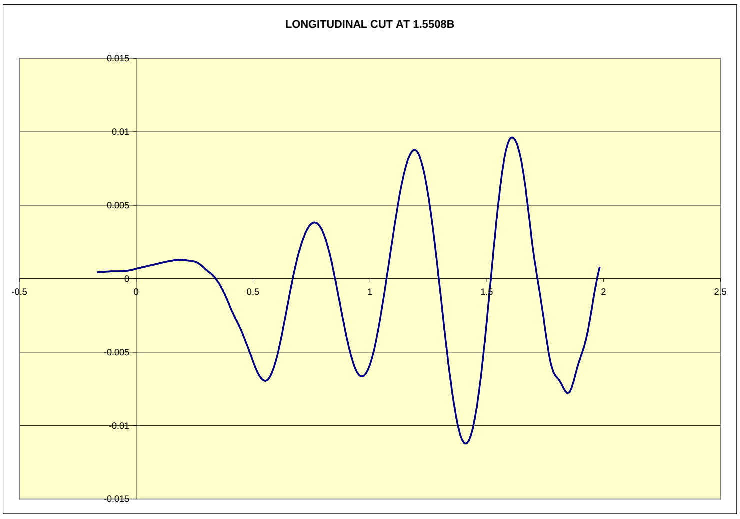


Figure 94

We have put these graphs in a non-dimensional format in the same way that we did for the S60 example. We refer to the explanation of that example to fully understand these longitudinal cut graphs.

Tutorial. Example 3: IACC Class Yacht.

This example is aimed at illustrating an alternative form to proceed with the hull mesh generation. In previous examples the hull is defined by lines which bound the surface patches forming the hull in the grid generator GiD. An alternative method would be to have the case study's geometry as surface patches directly in IGES format. This is the possibility we are now going to study in more detail.

We will use an America's Cup type hull. This geometry is defined by means of an IGES file. This file has been generated with a commercial program for ship design. It was later exported as NURBS surfaces in a 3D IGES[10] format.

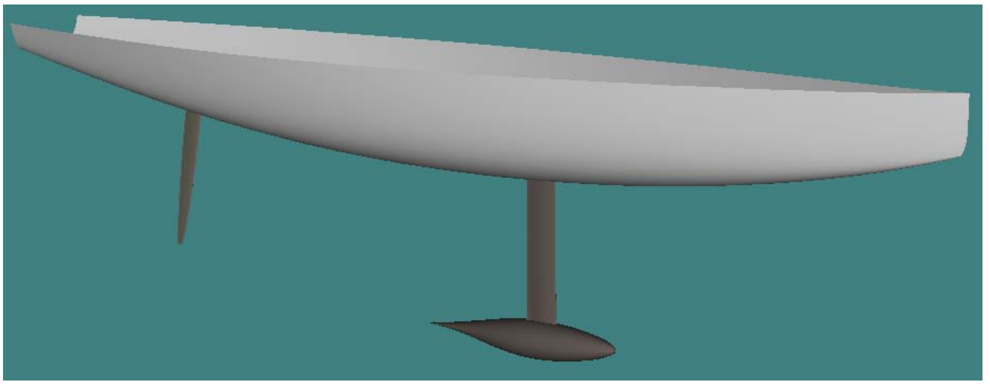


Figure 95

Preprocess

Once we have the geometric form in a GiD compatible format, we can start with the preprocess phase following the same steps as in the previous cases.

First, we import the IGES file to GiD, using the commands: *Files – Import- Iges..*

[10] IACC_0.igs

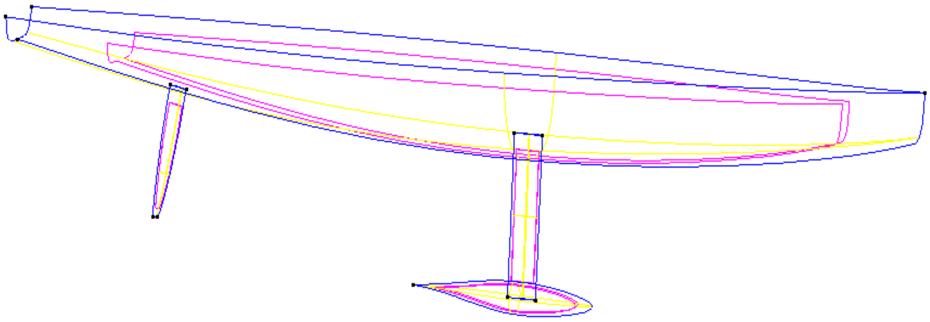


Figure 96

The first thing to do will be to eliminate the port section of the hull, including the rudder, the keel and the bulb[11].

The following step stems from the necessity of having no surfaces that get into the hull or fit the definition of dry submerged areas. Due to this, we need to trim the hull in its intersection with the keel and the rudder, and trim both rudder and keel in their intersection with the hull. This operation is performed with the option: *Geometry-Create-Intersection-Multiples Surfaces*. Once we have selected the surfaces to trim we should press *Esc*. This operation performed on the surfaces of the keel and the bulb should give rise to figure *96*. We see that after creating the intersections there are some new inner surfaces that we have to eliminate because they are not going to form part of our final geometric shape. The previous example of the bulb and the keel would finally end up as shown in figure 99.

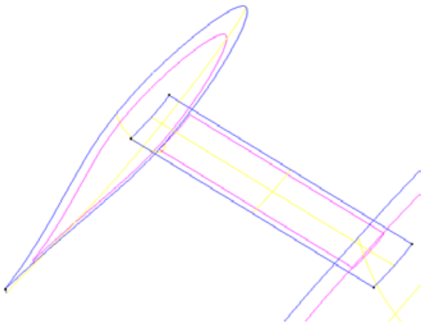


Figure 97

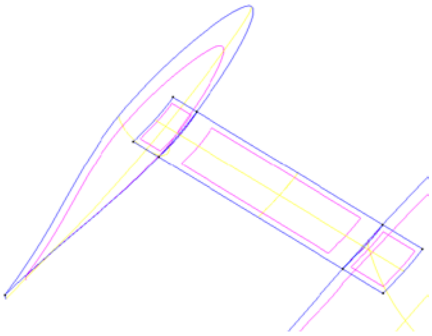
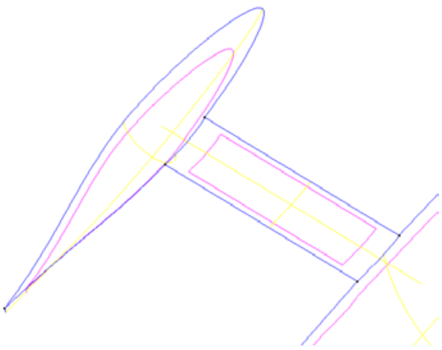


Figure 98



[11] Example file: IACC_1.gid

Figure 99

The surface trimming operation has to be performed between the rudder and the hull as well.

An important recommendation is to put each element in a specific layer in order to work with them more efficiently. A good proposal is the one shown in figure 100.

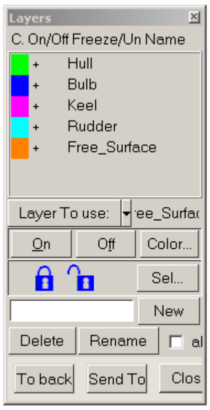


Figure 100

Another important step in the preprocess is to trim the hull (with??)at the free surface. First, we will define the free surface patch that we are going to use. Following the parameters defined in the previous tutorials, we will draw the edges of the free surface. We must take into account that the forward end of the free surface will be 1/4 LBP forward of the hull. The lateral side of the free surface patch will be 3/4 LBP away from the centerline and the back side of will be at LBP from the stern. We will have defined a quadrilateral like the one in figure 101.

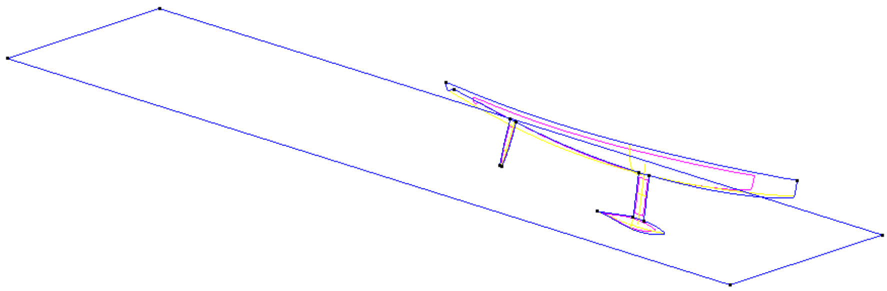


Figure 101

Since we have not defined the flotation line, we will create it from the intersection of the free surface and the hull. First, we will create the free surface patch from the quadrilateral defined in the previous step. Later on, we must intersect the free surface quadrilateral with the hull, using the command sequence *Geometry-Create-Intersection-Surface Surfaces*, and selecting the hull and the quadrilateral. This will produce the main water line. The problem arises when dealing with surfaces that are not so well defined, mainly the hull. In this case we can create a copy of both surfaces in another layer and extend to portside the free surface centerline so that the free surface encircles the hull. Once the new line has been defined, we can put the new elements back in the free surface layer, as can be seen in figure 102.

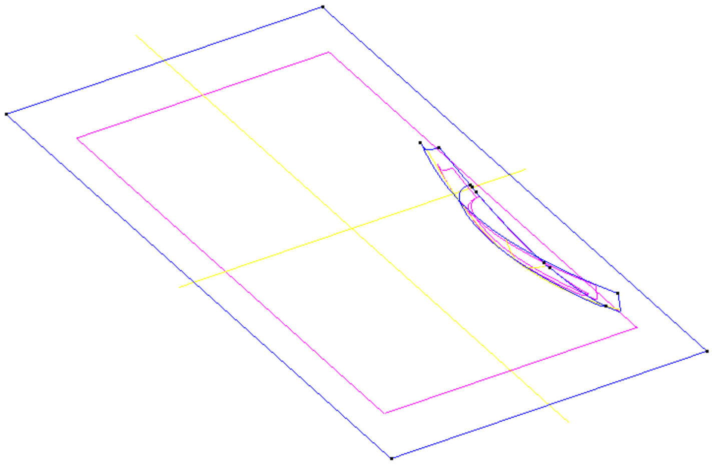


Figure 102

Again we have to eliminate all those surfaces that are to be discarded in our calculations. This includes the dry part of the hull and the auxiliary free surface patch that we created in the previous step to define the water line. Also, we have to verify that the lines defining the free surface patch are in the layer corresponding to the free surface. After this process, the layout should be similar to the one of figure 103.

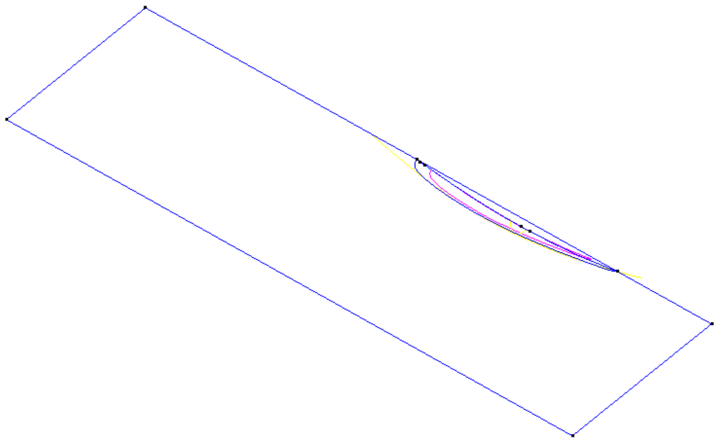
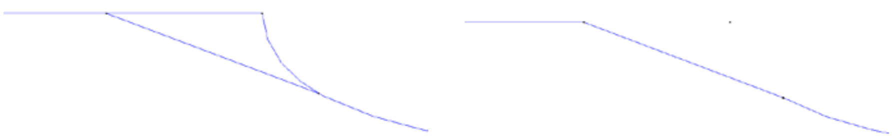


Figure 103

We will reconstruct the centerline of the free surface patch so that it contains the water line we have just generated. It might eventually be necessary to copy this line into the free surface layer as it could have ended up in the hull layer after this process.

In order to avoid stability problems in the stern region calculations, the connection between the waterline (and the hull??) must be as smooth as possible. We do this by drawing a straight line from the stern part of the waterline to the centerline. Later on, we smooth the junction with an arc using the command *Geometry-Create-Arc Tangents* (radio 5). A possible solution is the one shown in figure 104.



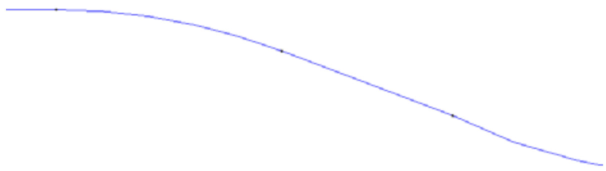


Figure 104

With the obtained waterline we trim the centerline to form the inner edge of the free surface patch. With this line and the other three segments we can define our final surface patch[12].

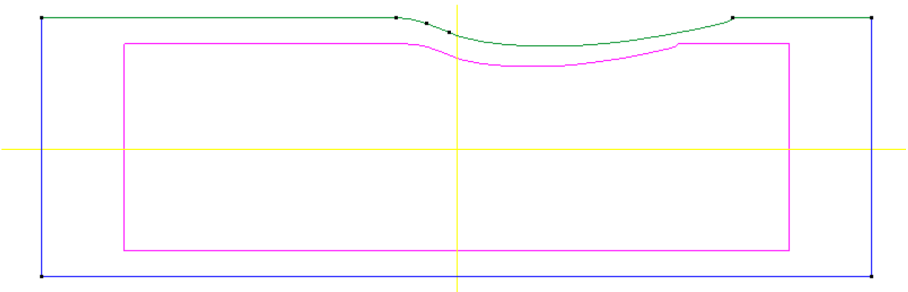


Figure 105

In order to be able to mesh a surface patch, the patch has to be a four sided one. If the patch is bounded by three lines, we have to split one of the lines to get the fourth side, as has repeatedly been shown. If the surface patch is bounded by 5 lines or more, the only option will be to split the patch. Therefore, we will have 2 four sided surfaces. We encounter this type of problem both for the hull and for the bulb.

The hull has been sectioned by the keel, the rudder and the free surface patch. It is bounded by 6 lines. Each line is limited by the points that we show in figure 106.

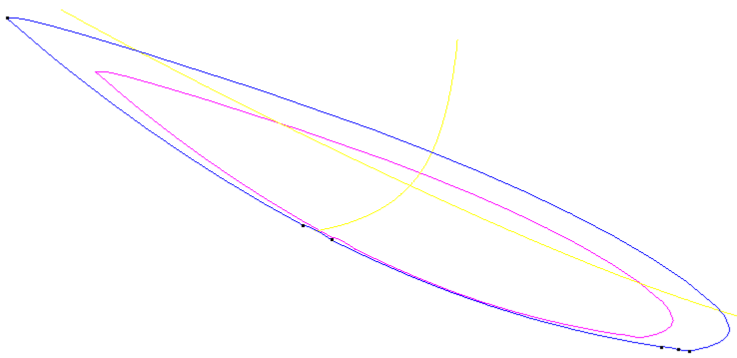


Figure 106

We are now going to split the hull in order to define it entirely with four sided patches. To divide the surface patches we will use the command *Geometry-Edit-Divide-Surfaces-Near Point.* This operation will offer us two directions in order to split the surface. We will select the appropriate direction in accordance with the criteria we have chosen for the whole splitting operation.

[12] Example file: IACC_2.gid

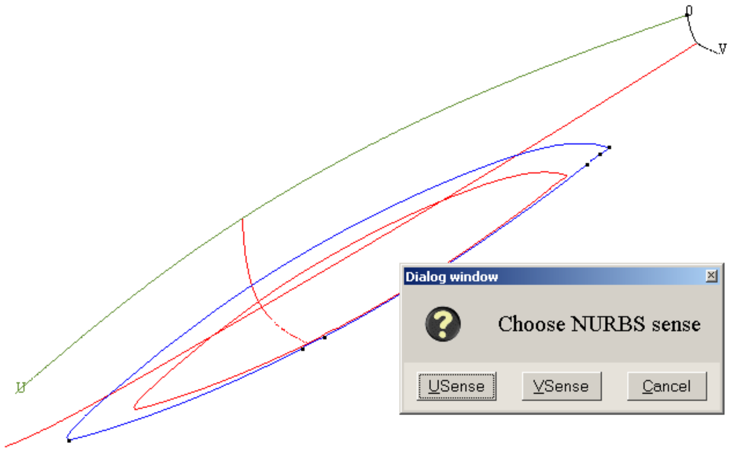


Figure 107

In the case shown in figure 107 we indicate the forward point on the bow and choose the direction U.

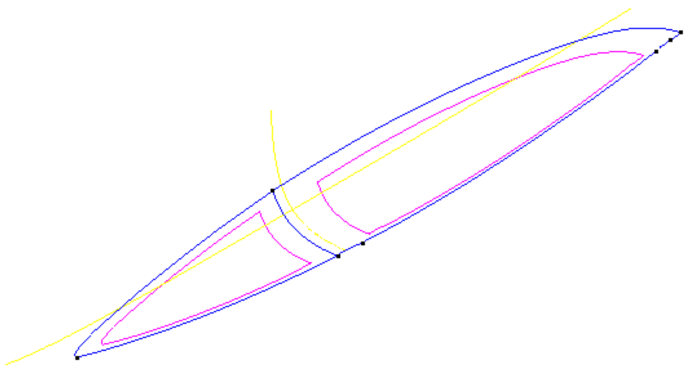


Figure 108

In figure 108 we see how the forward surface is bounded by 3 lines. The aft surface is bounded by 6 lines. If we divide the lines by the points shown in figure 108, we obtain a patch distribution with which we can proceed with the meshing operation, as can be seen in the next figure.

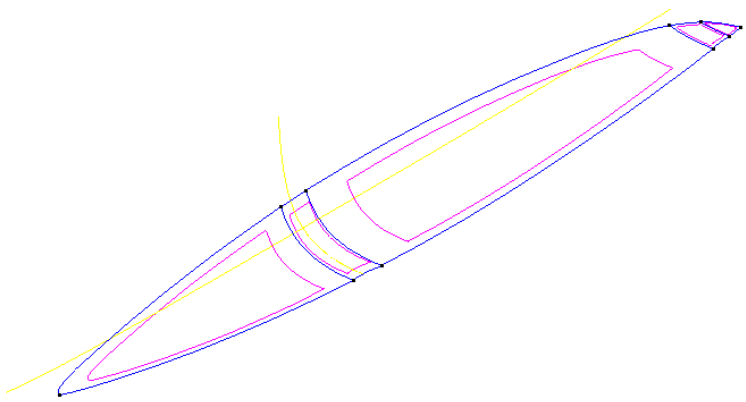


Figure 109

A similar process has to be undertaken with the complicated geometric shape of the bulb (figure 110). In this case we have to separate the forward region of the bulb in order to obtain a better definition. These types of decisions are usually better made after designing a preliminary mesh and trying out different possibilities and then checking whether the grid is satisfactory.

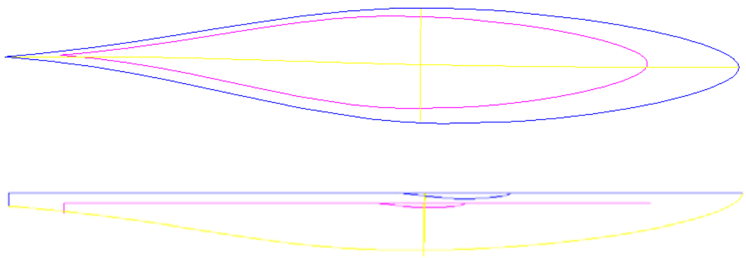


Figure 110

After some tests, the most reasonable option seems to be to divide the bulb in two halves, upper and lower. We take a horizontal plane passing through the trailing edge at the bulb stern as the section plane. We write down the coordinates of the points that define the straight line of the trailing edge and then, from this segment, we define the edges of the patch that will be used for the section.

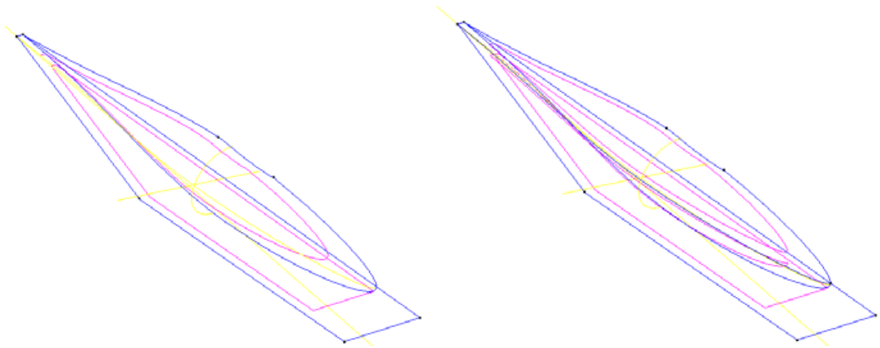


Figure 111

We will start by dividing the bulb surface with this section plane. Now, our experience says that it is usually necessary to create zones in the bow and stern with smaller surface patches. The final form should be similar to figure 112.

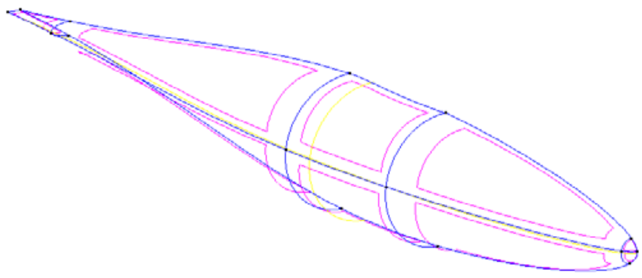


Figure 112

The forward region is formed by 3-sided patches. We have to split some of the edge segments to get 4-sided patches, as shown in figure 113.

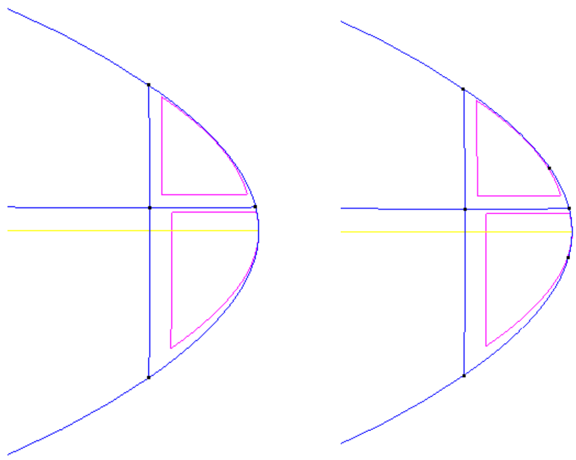


Figure 113

The following step will be to study the viability of the stern patches of the bulb. We must try meshing this region and if the mesh quality is not good enough we will have to create new lines or rebuild the surface patches.

During the visualization of the geometric shapes, it sometimes seems that some of the edges of the shapes do no exist. The reason is that these edges also bound other patches in a deactivated layer. This means that during the meshing process both lines will have the same number of divisions. To avoid this problem and to have the freedom to choose the mesh density in every surface, we must ungroup both surfaces. This operation is performed with the command *Utilities - Uncollapse – Surfaces* indicating the surface, in this case, the keel, as shown in figure 114. In this way we create a new edge line, and both surfaces have different patches. To visualize this border we put the surface patch, in this case the keel, in the right layer(right clicking *Layer –Send – Surfaces* and selecting the keel patch*)*.

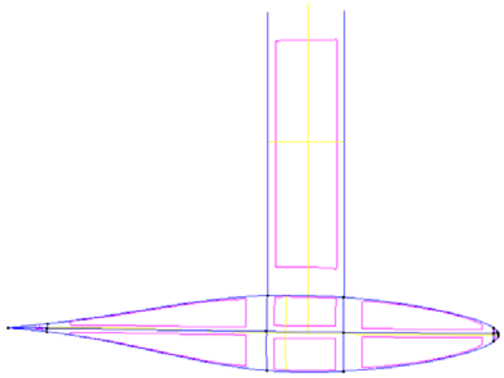


Figure 114

Let us proceed with the meshing of all the entities. Again through trial and error we try to obtain a smooth mesh on the hull and a density on the free surface that accomplishes the requirements explained in the previous chapters.

The density of panels on the free surface is mainly defined by the speed and length of the free surface patch. This case study will be tested at a speed of 8 knots. In the first estimation it will generate waves of a wavelength determined by the following equation:

$$\lambda = 2 \cdot \pi \frac{V^2}{g}$$

Where V and g are dimensionally coherent.

The wavelength for this case study is 10,86 m. Since our free surface is 48m long, we see that we can house 48/10,86 = 4,42 complete waves within our domain. Taking into account that the representation of a wave requires approximately 15 panels, we see that we will need at least 67 panels in the lengthwise direction. For the transverse direction we choose 30 panels, so that the panels have an aspect ratio close to 1. The free surface will have a layout similar to figure 115.

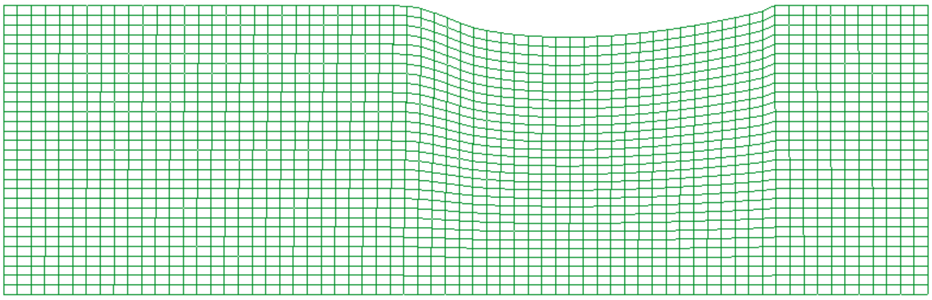


Figure 115

There are no strict rules for the hull mesh and the appendages. We will mesh with a density sufficient to be sensitive to the form and the details. We will try to ensure that the forward third has a denser grid than the rest. We will also pay greater attention to the definition of the leading edges of the different appendages. We use the command *Meshing – Structured - Concentrate elements* to get this. We must select the edges of the side where we want a greater density. We select a weight value that has to be fine tuned by trial and error.

In figure 22, we see the keel after selecting the top and bottom edges as the direction for the longitudinal variation of the panel density. Next, a box will appear to set the weight for the mesh in that direction.

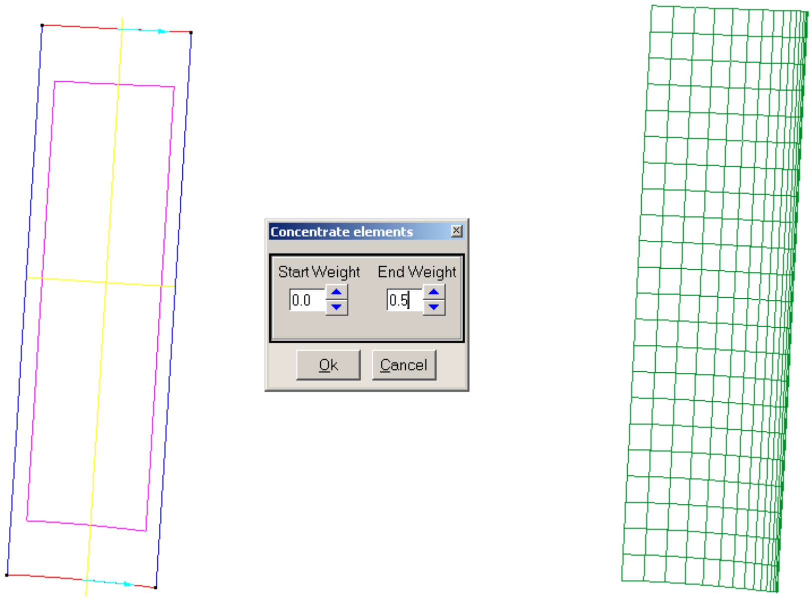


Figure 116

Before proceeding with the hull meshing, forward and stern patches must be modified since they are three sided surfaces. We have chosen the division points shown in figure 117.

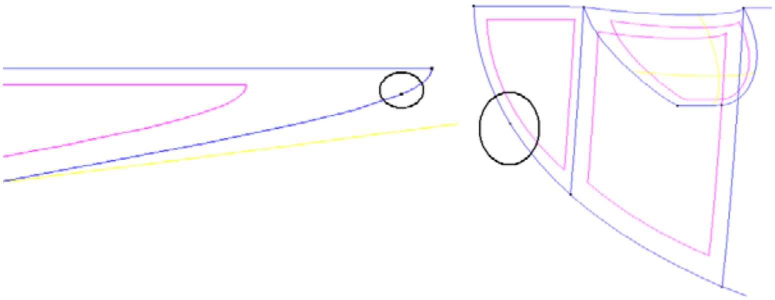


Figure 117

Next we show the different elements of the hull with the proposed mesh[13].

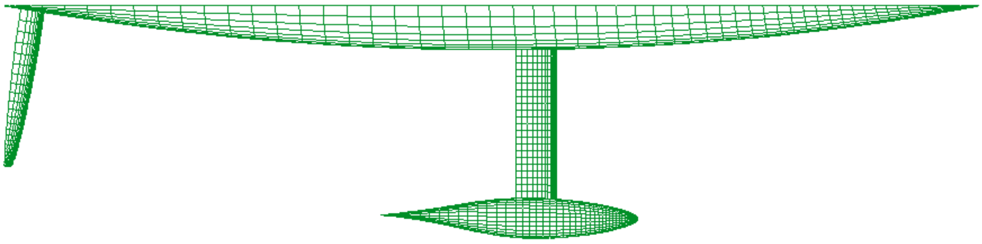


Figure 118

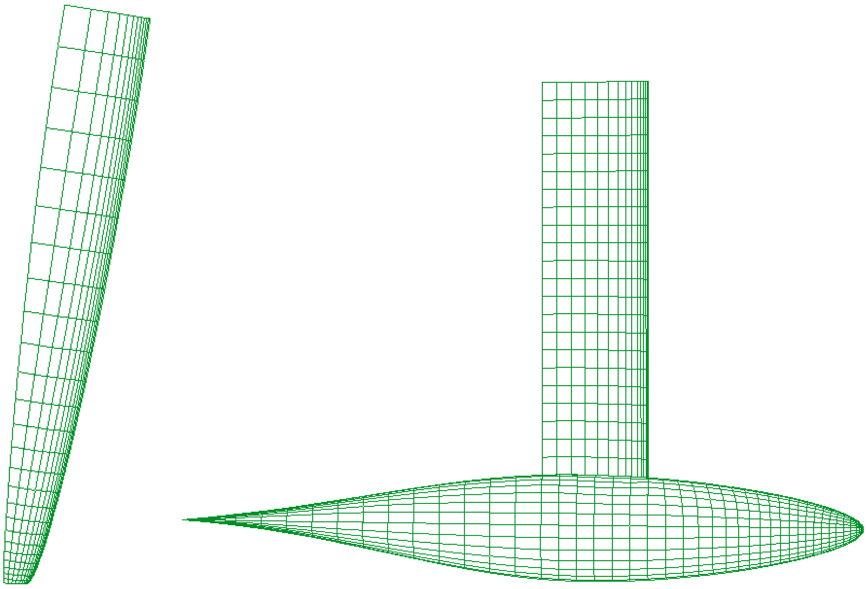


Figure 119

[13] Example file: IACC_3.gid

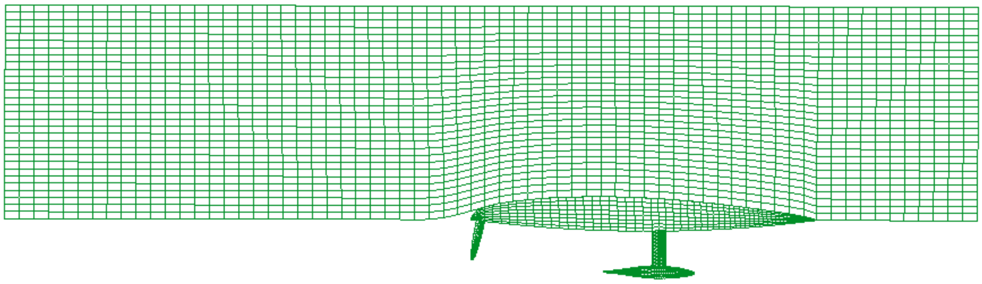


Figure 120

Once we have a mesh, we have to make some verifications prior to the calculation. First, we have to ensure that the normal vector's orientation, points towards the inside of the hull. We must also verify that the free surface's normal vector, points downwards. For the visualization of the normal vectors, the command *Utilities - Draw Normals - Surfaces* is used. In order to change the orientation we will right click and then *Contextual - Swap Some.*

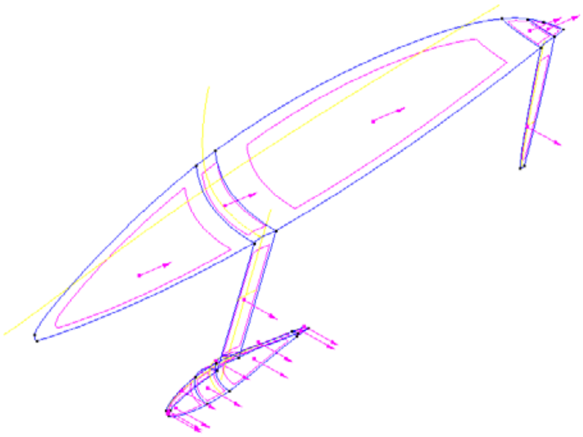


Figure 121

Calculation process: ETSIN_CFD.

The last step before calculating is to define (inside GiD context), the type of problem we are solving. Since this is a generic case of symmetrical flow, without transom, without drift and list, we will indicate *Data - problem Type - ETSIN_CFD*

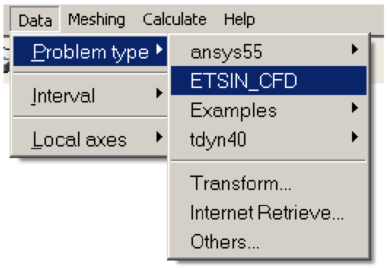


Figure 122

The following operation will be to assign conditions to the different elements of the mesh, basically hull and free surface. Within the *Data* menu the *Conditions* submenu is opened and the *Hull Condition* is assigned to the hull and appendages (*condition 1* if we show conditions with the *Draw* command). The *Free Surface* condition is assigned to the free surface (*condition 2* if we show conditions with the *Draw* command.)

Another form that must be filled in within the *Data* menu will be the one corresponding to the principal dimensions of the problem, *Problem Data*, figure 123.

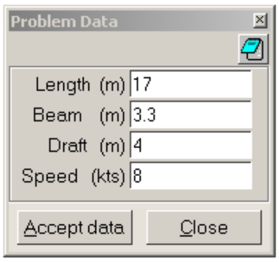


Figure 123

We will regenerate the mesh (*Generate* option in the *Meshing* menu) so that the new mesh reflects the changes done so far. In this case, we have a 3721 panel mesh.

We are now ready to launch the calculation process. Within the *Calculate* menu we open the *Calculate Window* submenu and press *Start* to begin the calculation.

Postprocess

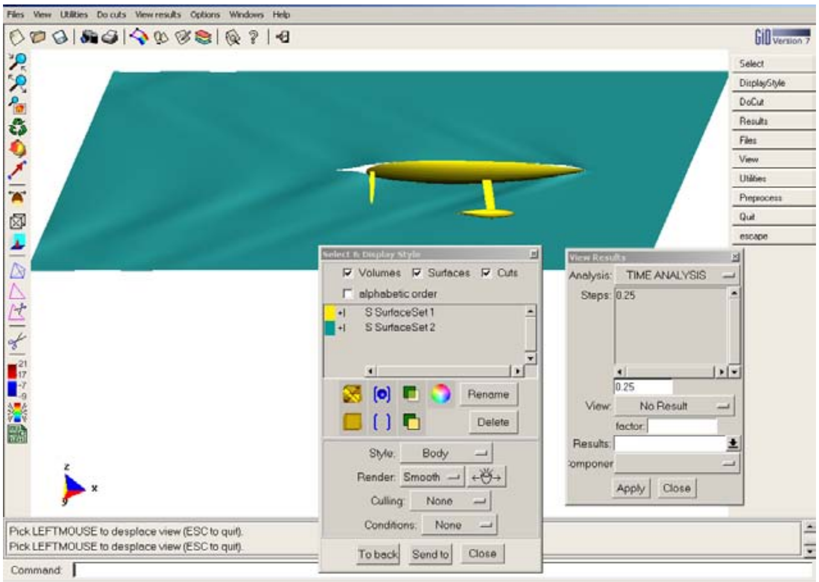


Figure 124

Once the calculation is finished, we have to get into the postprocess context of GiD. This can be done simply when requested after the calculations or with the *Files>Postprocess* command. We have to set certain values for a correct interpretation of the results. *View Style* and *View Results* menus have to be active. First we have to select the type of view. Within *view* menu, we select *Contour Fill*.

With *ETSIN_CFD* we have the possibility to visualize different results in GiD. Speed field can be visualized in both the hull and the free surface. Both pressure coefficients on the hull and the wave pattern on the free surface can also be visualized. In order to select one of the visualizations we will use the *View Results* menu.

Visualization of the Pressure Coefficient.

Pressure Coefficient (Cp): This coefficient is a non dimensional way of representing the pressure on the hull.

$$Cp = 1 - \left(\frac{V}{V_\infty} \right)^2$$

In order to visualize the pressure coefficient function on the hull, we have to select it on the *View Results* menu (figure 125 left hand side). After selecting it, we click *Apply*. The pressure coefficient distribution appears only on the hull since it makes no sense to draw this pressure coefficient on the free surface where the pressure is atmospheric. For a better visualization we can deactivate the layer that contains the free surface. In order to do this, we use the *Select & Display Style* menu (figure 125 right hand side). *S Surface Set 1* corresponds to the hull mesh and *S Surface Set 2* to the free surface mesh. By deactivating the free surface layer with the icon and by pressing *Apply* in the *View Results* menu, only the hull will remain visible.

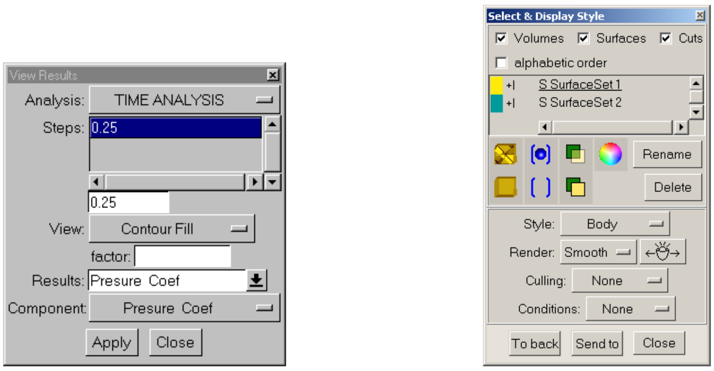


Figure 125

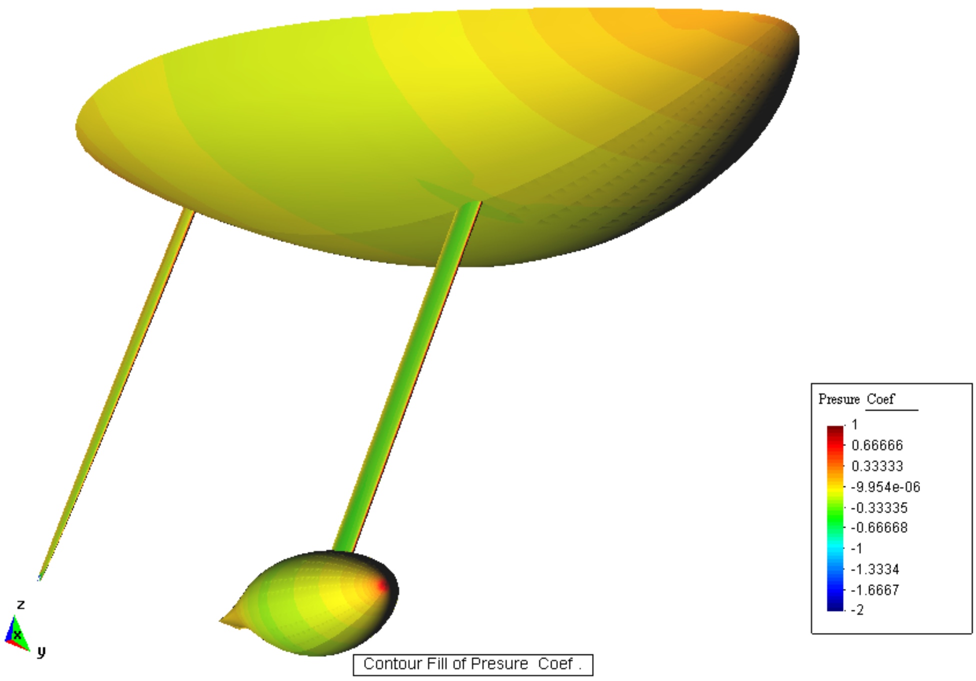
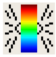


Figure 126

GiD automatically defines the scale limits for the graphic representation of all the results. We recommend that you set them manually and keep them for the incoming comparisons. *ETSIN_CFD* is basically a comparative tool. The most reliable information is obtained from comparing the results of different ship hull forms. This is why it is so important to keep the scales consistent for the different case studies for the same graph types. We have selected a scale ranging between -2 and 1 for the pressure coefficient. This can lead to the non representation of some small regions, for instance the rudder's lowest edge. This is the part of the ship where the gradients are greatest. If we do not set this limit, the entire colour range will be concentrated on the rudder and the rest of the hull will appear monochromatic. For the range definition, we use the icon 

to define the maximum and the icon 

to define the minimum. For an automatic colour range selection and to reset the range when switching visualization, we must use this icon 

Wave pattern visualization.

Another feasible visualization is the wave pattern on the free surface.

Again in the *View Results* menu, we select the visualization that we want with this function: , in this case the *wave pattern*. To only visualize the free surface, we enter into the *Select & Display Style* menu. In this menu, when deactivating *S Surface Set1,* only the free surface remains visible. Do not forget to click *Apply* after every change. We must also set maximum and minimum visualization values for the incoming comparisons.

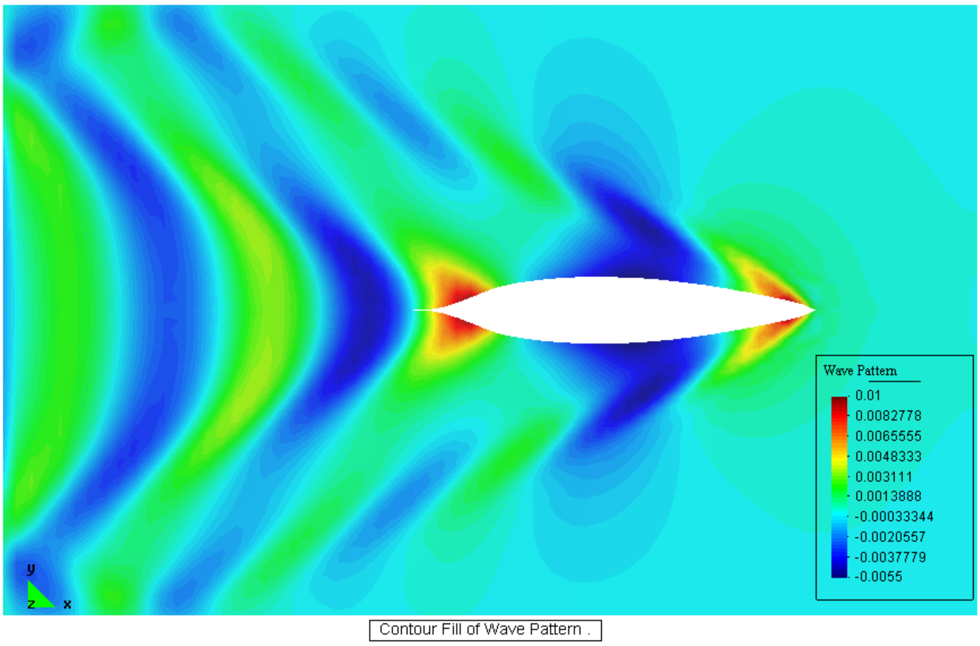


Figure 127

Velocity field visualization.

The last visualization option is the speed field. This magnitude gives a lot of information, both for the hull and for the free surface. The flow is potential and therefore, the speed on the hull surface is not the ship's speed itself. To correctly interpret this output data we must consider the ship to be still and that the water advances towards it at the ship's forward speed.

Again, we follow the same steps and visualize *Velocities.* There is also the possibility of choosing any speed component as well as the module.

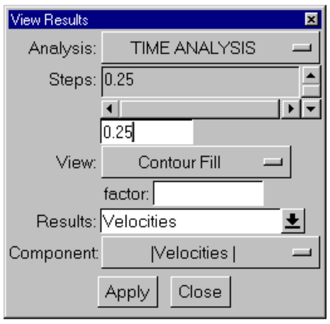
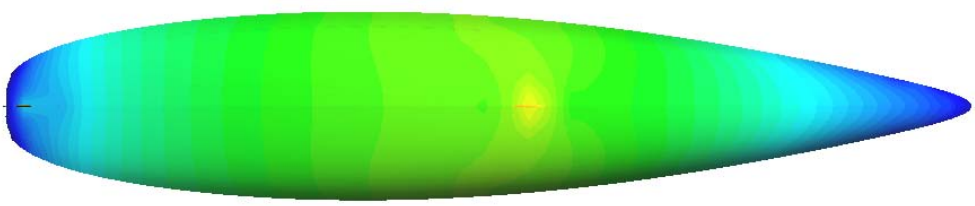


Figure 128

Figure 129

Since all the colour range is concentrated on the rudder's tip (figure 129), we must manually set the range limits to make our visualization useful. These limits must be set for every visualization option. For instance, since the smallest velocity regions correspond to the appendages, we must show the velocity gradient on the hull without the appendages and set a greater value for the minimum.



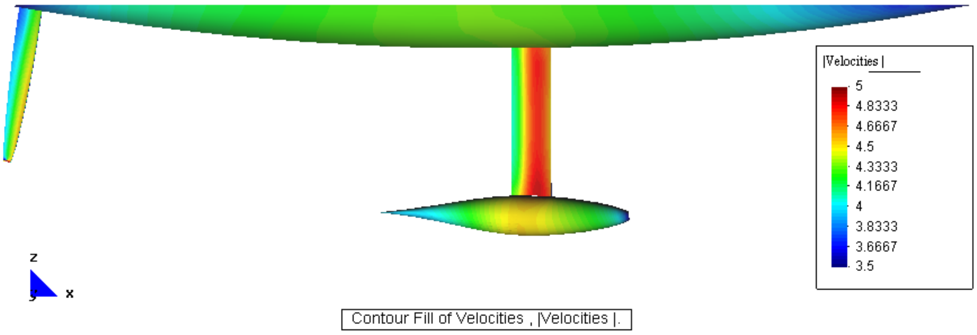


Figure 130

In figure 130, we can see that the forward part of the bulb and the rudder's leading edge do not appear because their velocity range is out of the selected range.

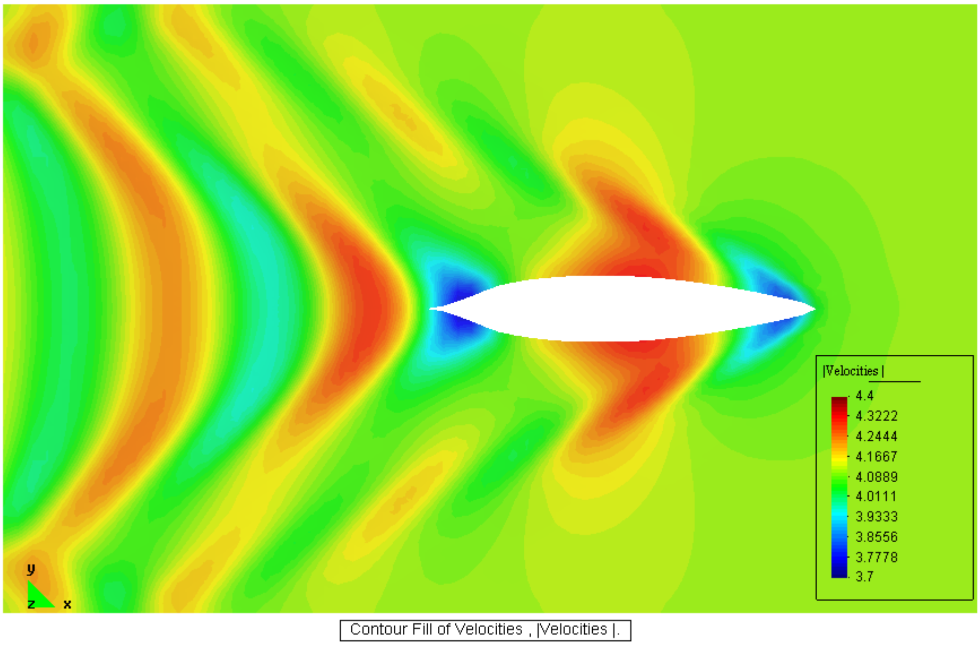


Figure 131

Longitudinal cuts

The results files are kept in folder *results* within the folder *casename.gid* where the mesh used for the calculation was saved. This folder also contains the ascii files long_cut_1.5508B.dat, long_cut_1.0587B.dat and long_cut_0.5665B.dat. Inside these files are the curves corresponding to the longitudinal sections of the free surface at 0.5665B[14], 1.0587B and 1.5508B. With these graphs we can visualise the height of the generated waves and their other characteristics. This is one of the most useful tools when optimising the ship's hull, as we will discuss later. They can easily be imported to Excel, for instance. They are represented in the following figures:

[14] B= ship's beam

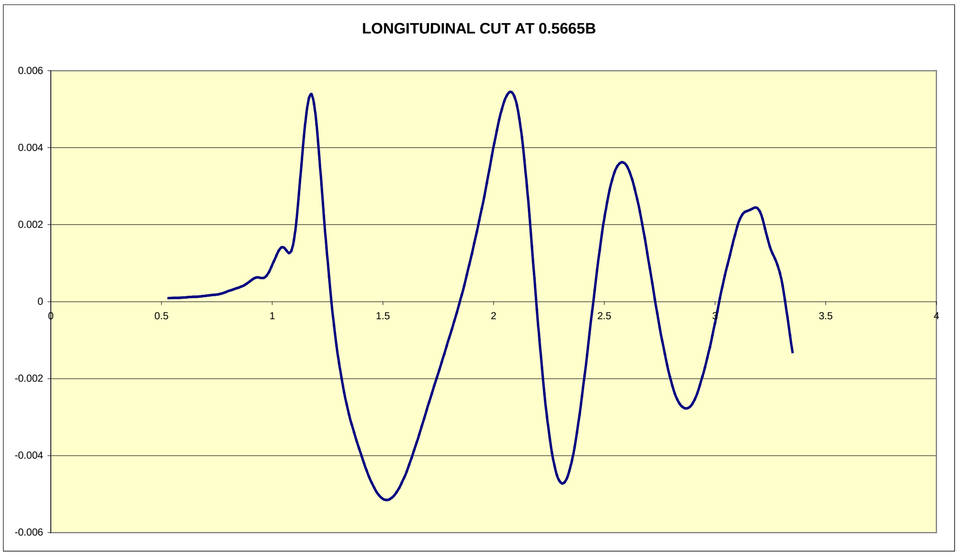


Figure 132

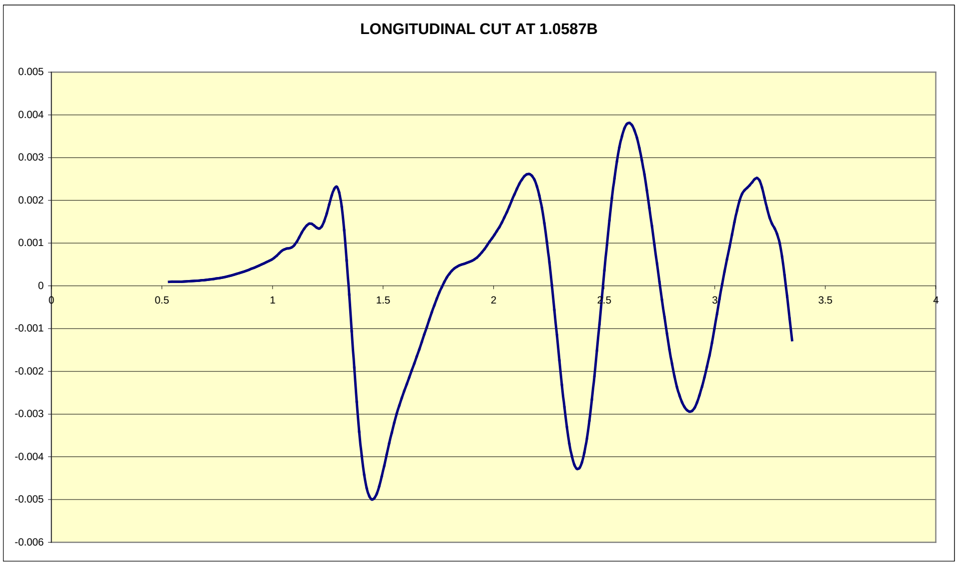


Figure 133

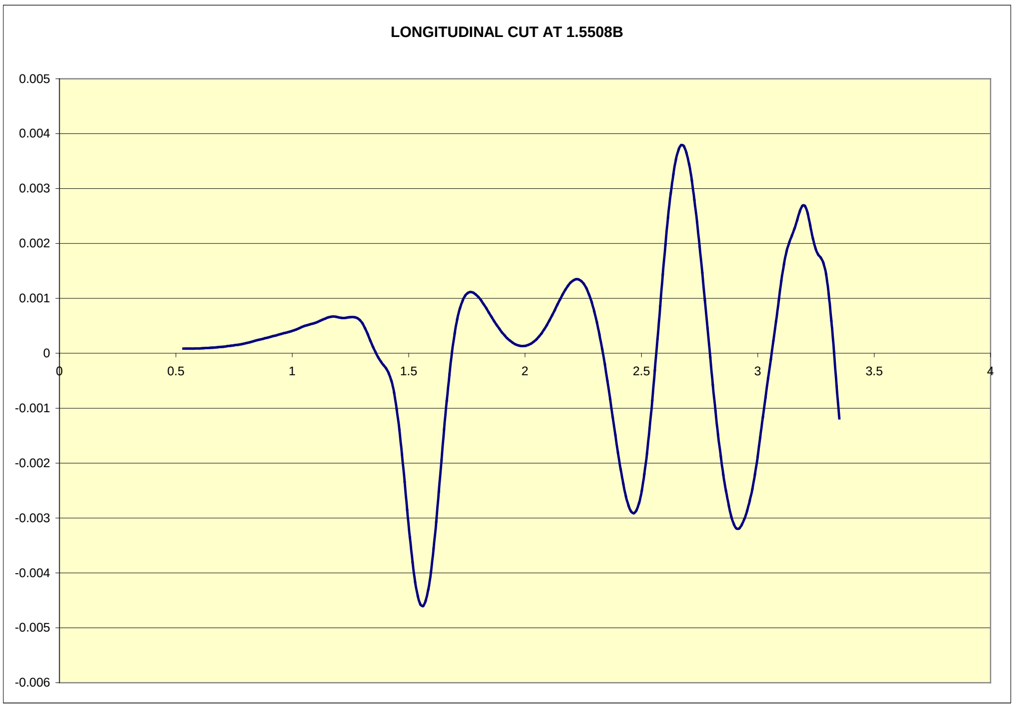


Figure 134

These are non dimensional graphs. Along the X-axis, the free surface X-component is represented. It is in its non dimensional form with LBP as the reference length. The orientation is the common one. Therefore, x=0 corresponds to the ship's bow and x=1 corresponds to the stern. Along the vertical axis we have the free surface height in non dimensional form, taking again LBP as the reference length. So, if we want to know the proper value of the height of some of the generated waves, we must multiply its graph value by LBP.

Hull Optimization Examples

In this chapter a series of examples will be presented, dealing with how to optimize hulls through the use of the *ETSIN_CFD* code.

Optimization is achieved by comparing alternatives hull forms. Meshes are prepared for the different ship forms (changing bulbs, stems, and other elements that play an important role in wave generation). The calculation case is performed for each alternative at the ship design speed. With the obtained data it is possible to decide which of these designs is best.

In our search for the optimum design, we will have to make a series of assumptions. Since this method calculates the potential flow and not the viscous one, the optimal design is the one that generates the smallest amplitude waves. The smallest amplitude is related to the smallest wave resistance and hence to the smallest forward resistance.

When the amplitudes of the waves are similar, attention has to be paid to the pressure coefficient distribution and speed fields on the hull and free surface. If gradients in these fields are greater in any particular design, this means that this choice is worse regarding wave generation simply because these gradients are a sign of higher waves.

Since the calculations do not consider the viscous effects, results will be more accurate in the forward third of the hull. Viscous effects are more important from the forward third of the hull onwards and dominate in the stern region, especially at low speeds. Therefore, this tool must be focused on the optimization of the forward part of the hull, with special attention paid to the bulbous bow if present.

In order to illustrate this optimization method, we are going to present the case of four fishing boat designs that were optimised and then decide which among them would be the one with the smallest wave resistance. Two of them are with bulb and two without.

Hull design alternatives.

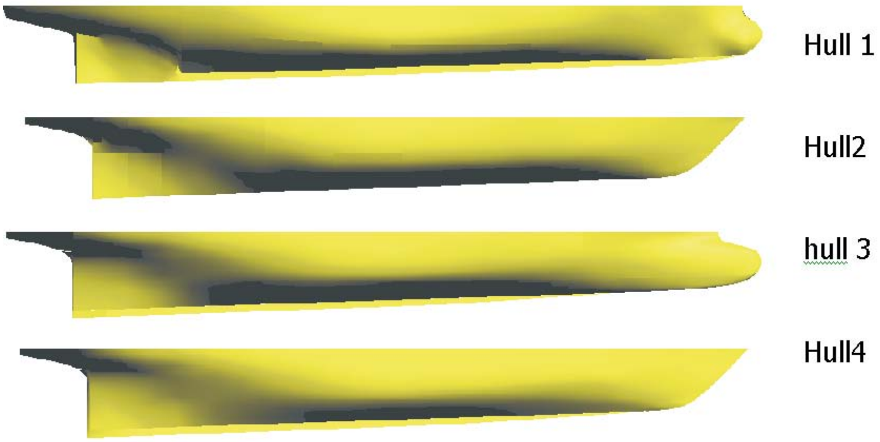


Figure 135

A hull and free surface mesh were prepared for each design proposition.

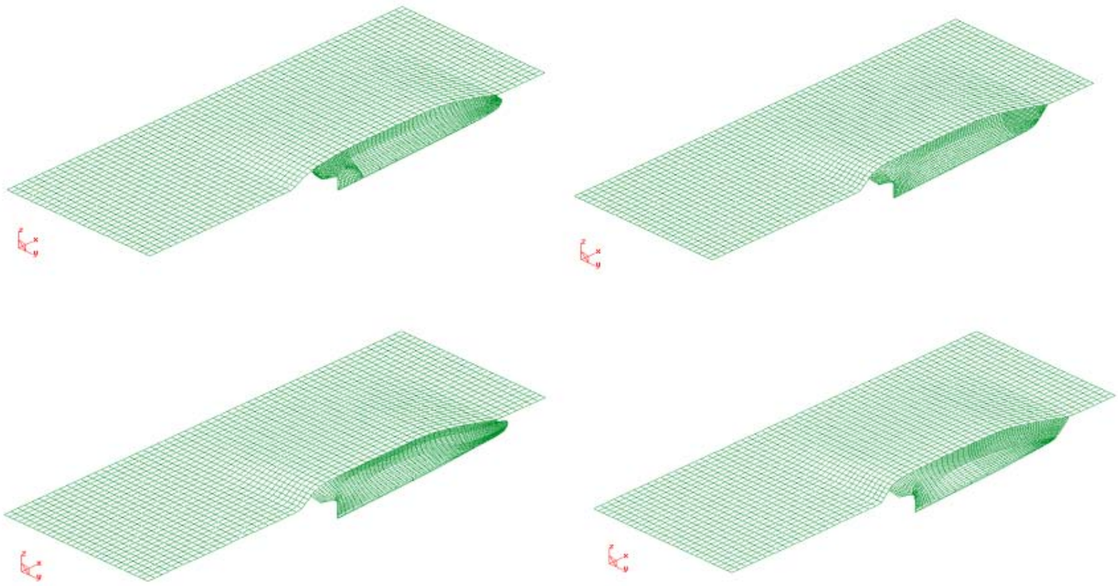


Figure 136

The following step is to launch the calculation processes and once concluded, we will obtain the maps of speed distribution within the GiD postprocess and analyze the graphs of the longitudinal cuts.

Speed field graphs on the hulls.

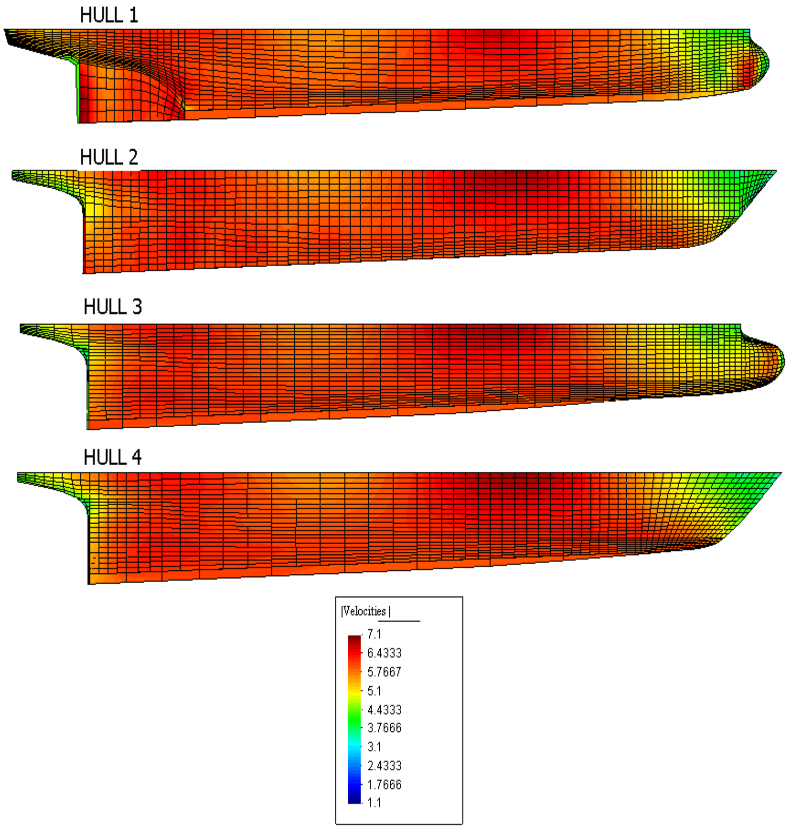


Figure 137

In these first representations we can see that the zones with greater gradients are in the forward third of the hull. We zoom in/on these regions to further analyze them and obtain the corresponding conclusions.

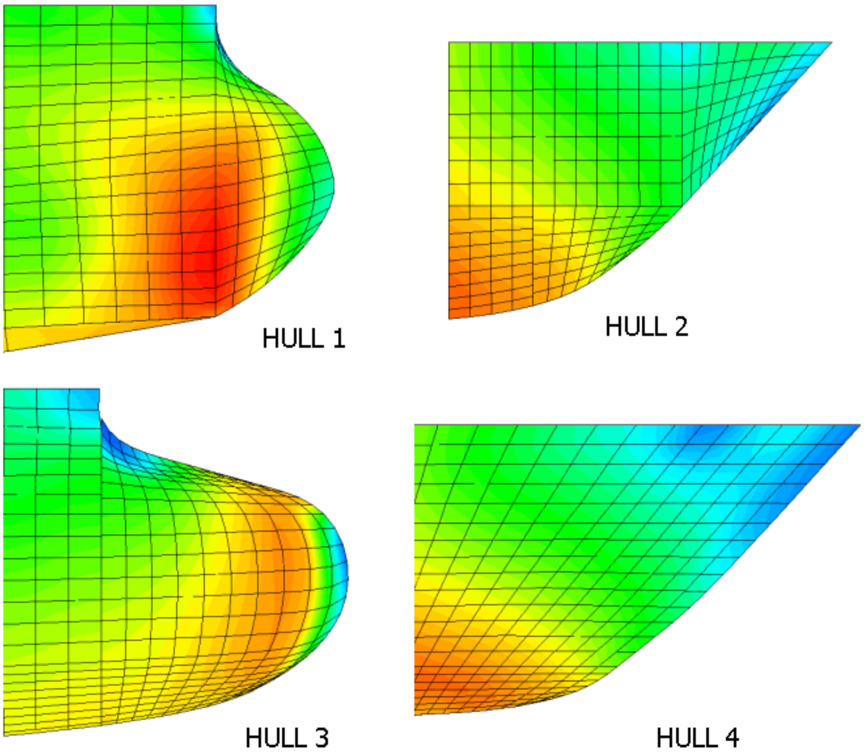


Figure 138

The first visual impression is used to notice the bad hydrodynamic properties of the bulbous bow of design 1. Considerable gradients appear for the speed field leading to higher deformations on the free surface and therefore greater wave resistance. Hulls 2 and 4, seem to have little to present in this phase when comparing them with hulls 1 and 3 due to the obvious shape difference between them. For a more precise analysis, it is essential to use the free surface longitudinal cuts corresponding to these designs.

In order to better compare the different cases, we use one graph for the four corresponding longitudinal cuts of the wave systems. The usual thing is to represent these cuts at several specific distances from the centre line. These distances are 0.5665*B, 1.0587*B and 1.5508*B as has already been mentioned.

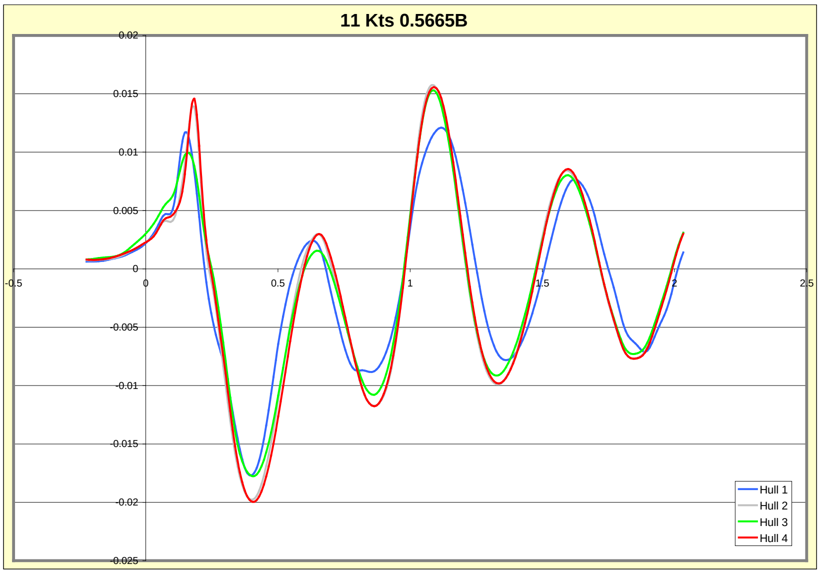


Figure 139

The first cut, the one closest to the hull, is the one that provides the best information for the amplitude of the generated waves on the free surface. Within it, the bow wave and the forward third of the ship are the regions with the most precise definition for our CFD. If we focus on this region, we can see that the wave with the smallest amplitude is the one generated by hull 3, followed by hull 1 and then, with quite a big difference, the worst ones are hulls 2 and 4. This makes the choice of a bulbous bow a good one and we now have to choose between hulls 1 and 3.

For a more educated choice between hulls 1 and 3 we study the next wave cuts and verify whether hull 3 is still the one with the smallest wave amplitude.

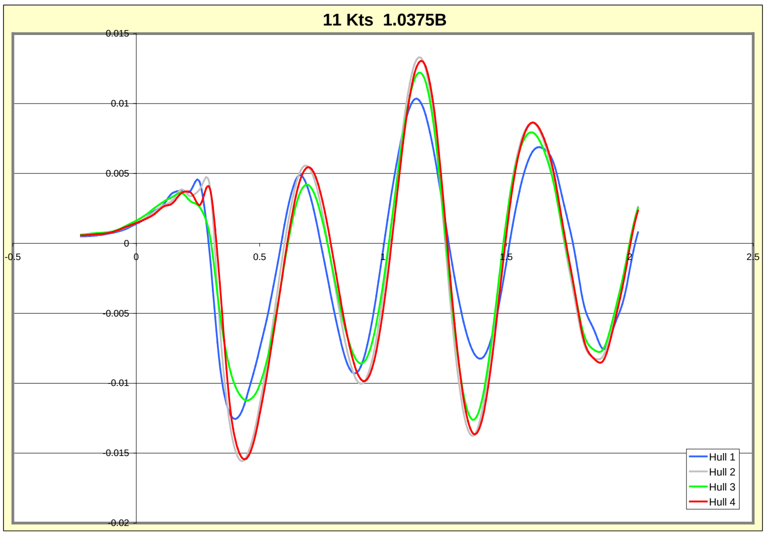


Figure 140

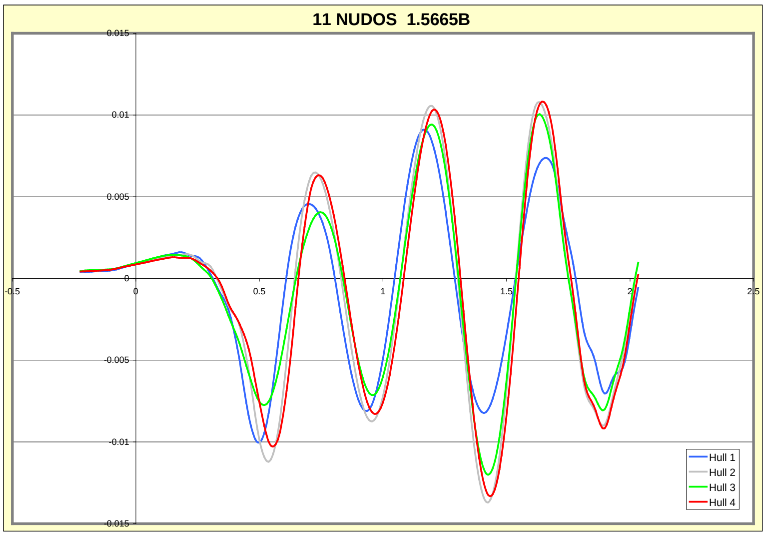


Figure 141

We can now verify that hull 3 presents the lowest amplitude waves in the forward region of the cut. This analysis allows us to state that the best choice, regarding wave generation and therefore wave resistance, is hull 3.

This example has demonstrated how to discard, among different alternatives, those that will produce a higher wave resistance. Therefore, taking this process to its limits we are able to evaluate the best design among those of a series of close alternatives. To perform this comparison we visualise, on the same graph and for a certain speed, the wave cuts corresponding to each case.

This methodology provides us with a good tool when devising modifications in hull shapes, bulbous bow designs, appendages... We get knowledge of the quality of the alternatives by comparing all of them.

To precisely define the required power in order to project the engine system, the best thing to do now is to do a towing test in a model basin of the selected alternative. With this experiment, the speed resistance curve is obtained and used afterwards to select the engine and design a suitable propeller.

References.

[1] Bruzzone, D., Numerical Evaluation of the Steady Free Surface Waves. Proceedings of CFD Workshop Tokyo 94. Vol 1. Pag 126-134.

[2] Centro Internacional para métodos numéricos en Ingeniería, CIMNE. GiD Manual de Utilización.

[3] Dawson C.W.: A Practical Computer Method for Solving Ship-Wave Problems, Proceedings of Second International Conference on Numerical Ship Hydrodynamics. Berkeley, pp. 30-38, (1977)

[4] García Espinosa, J., Pérez Rojas, L., Valle Cabezas, J. y Chacón Alonso, J,R., El Proyecto BAJEL: Una herramienta de diseño hidrodinámico de buques de pasaje. XXXIV Sesiones Técnicas de Ingeniería Naval. Barcelona. 1998.

[5] McNeel, R. & Associates. Rhinoceros Nurbs modeling for Windows. User Manual. 1993-2001.

[6] Pérez Rojas, L., Souto, González, L.M., Los CFD (Computational Fluid Dynamics) en el diseño de buques. Aportaciones de la E.T.S.I.N.(U.P.M.). IV Simposio Marítimo Internacional 13-15 de junio 2001 - La Habana, Cuba

[7] Pérez Rojas, L. Diez años de I+D en el Canal de la ETSIN. XXXVI Sesiones Técnicas de Ingeniería Naval (Nacional). Cartagena, Noviembre 1999. Asociación de Ingenieros Navales y Oceánicos de España.

[8] Pérez Rojas, L., Zamora, R., Souto A., Abad, R., La contribución hidrodinámica del Canal de la ETSIN al Proyecto Copa América Español XXXVIII Sesiones Técnicas de Ingeniería Naval, Barcelona, Noviembre 2000

[9] Souto, A., Nuevas herrramientas de diseño de formas de buques basadas en códigos de flujo potencial. Tesis Doctoral. Departamento de Arquitectura y Construcción Navales. E.T.S.I. Navales. U.P. Madrid. 2001.