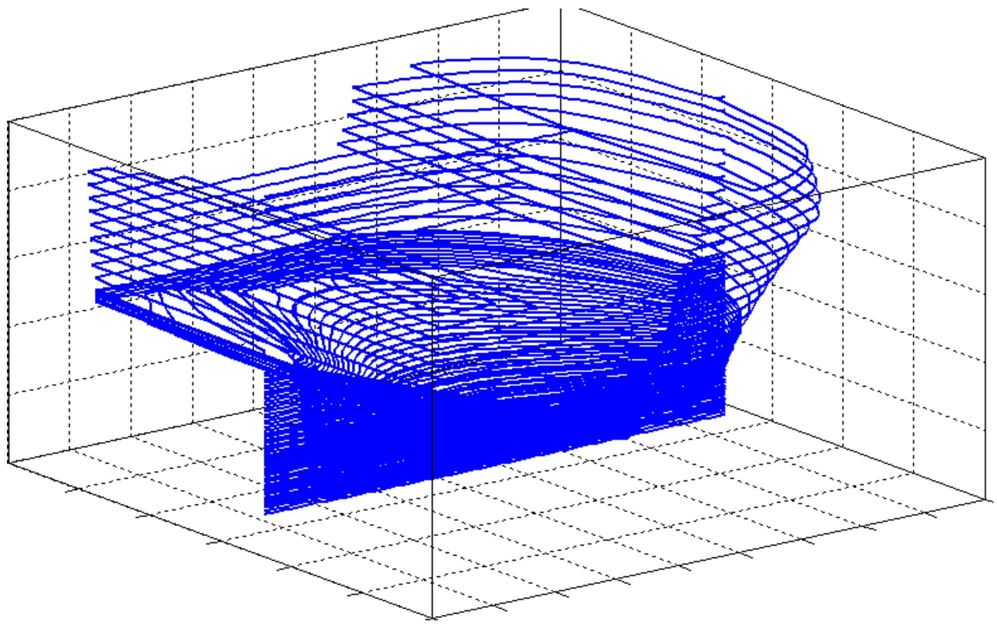
Proyecto Fin de Carrera

Ingeniería Técnica Naval

Programa de las curvas hidrostáticas para un buque, en su condición de adrizado, realizado con MATLAB



Titulación: Ingeniería Técnica Naval Esp. Estructuras Marinas

Autor: Pablo Belmonte Rodríguez

Director: Leandro Ruiz Peñalver

Juan Carlos Trillo Moya

Programa de las curvas hidrostáticas para un buque, en su condición de adrizado, realizado con MATLAB

Autor: Pablo Belmonte Rodríguez

Director: Leandro Ruiz Peñalver

Juan Carlos Trillo Moya

Noviembre 2013

Agradecimientos

Desearía agradecer a toda mi familia, en especial a mi padre y a mi madre, la paciencia que han demostrado tener conmigo todo este tiempo. A Leandro Ruiz Peñalver, mi director, por tener siempre una solución para mí y por su paciencia conmigo a largo de todo el periodo. Y agradecer especialmente a Juan Carlos Trillo, mi director también, que durante este tiempo se ha convertido en un gran amigo y me ha apoyado en los momentos más difíciles.

1- Introducción *9*
1.1- Sinopsis 9
1.2- Objetivo 10
2- Plano de formas y modelado en 3D *11*
2.1- Superficie NURBS y modelado del casco. 12
2.1.1-Curvas y superficies NURBS 12
2.2- Modelado del casco 15
2.2.1-Calcar el plano de formas original 15
2.2.2- Creación de una superficie NURBS en Rhinoceros 17
2.2.3- Proceso de alisado 20
2.2.4- Obtención de datos de las formas del buque 23
3- Propiedades geométricas de la carena. *27*
3.1- Momentos estáticos y centros de gravedad 28
3.1.1- Momentos estáticos y centros de gravedad de un área plana 28
3.1.2- Momentos estáticos y centros de gravedad de un volumen 29
3.2- Momentos de Inercia 30
3.2.1- Momento de inercia de un área plana 30
3.2.2- Teorema de Steiner 31
3.3- Figuras planas 31
3.3.1- Área 31
3.3.2- Momento estático o de primer orden respecto del eje OY 32
3.3.3- Momento estático o de primer orden respecto al eje OX 32
3.3.3- Centro de gravedad del área, centro de área, centroide o baricentro 32
3.3.4- Momento de inercia o momento de segundo orden respecto al eje OY 33
3.3.5- Momento de inercia o de segundo orden respecto al eje OX 33
3.4- Figuras del espacio 33
3.4.1- Volumen encerrado por una superficie y unos planos limítrofes 33
3.4.2- Momentos estáticos del volumen encerrado por una superficie y unos planos limítrofes 35
3.4.3- Coordenadas del centro de gravedad del volumen limitado por una superficie y planos limítrofes 36

4- Integración numérica. 37

4.1- Introducción 37

4.2- Integración numérica 38

4.3- Métodos de integración numérica 39

4.3.1-Regla del Trapecio 39

4.3.2 Método de Simpson 41

5- Curvas hidrostáticas o propiedades de la carena recta. 49

5.1- Propiedades de las áreas de flotación 50

5.1.1- Área de flotación 50

5.1.2- Área de la sección.. 51

5.1.3- Abscisa del centro de flotación 51

5.1.4- Momento de inercia transversal, momento de inercia del plano de flotación y longitudinal 53

5.1.5- Radio metacéntrico transversal y longitudinal 54

5.2- Propiedades del volumen 56

5.2.1- Volumen de carena. Posición trasversal del centro de carena.................... 56

5.3- Datos derivados 57

5.3.1- Toneladas por centímetro de inmersión.. 57

6-Introducción a MATLAB 59

6.1- Introducción a MATLAB 59

6.2- Introducción de datos. Uso de la ventana de comandos 60

6.3- Variables de entorno y variables especiales 61

6.4- Elementos de las matrices 62

6.5- Operaciones con matrices 63

6.6- Funciones orientadas al análisis de datos 65

6.7- Polinomios 65

6.8- Gráficos 66

7- Programando curvas hidrostáticas con MATLAB 69

7.1- Introducción al programa 69

7.2- Transformación del archivo GHS (*gf) a un archivo Excel (*xls) 70

7.3- Obtención de las secciones 74

7.4- Dibujo de las secciones 76

7.5- Obtener líneas de agua. 82

7.6- Dibujar líneas de agua 85

7.7- Cálculos curvas hidrostáticas 87

7.8- Interfaz gráfica 97

7.8.1-Ejecución de la interfaz gráfica 97

7.8.2-Documentación de la Interfaz Gráfica 98

7.8.3-Cómo utilizar la interfaz gráfica 101

8- Valoración de los resultados 107

9- Conclusiones 121

10- Bibliografía 123

11- ANEXO I 125

1- Introducción

1.1- Sinopsis

El presente proyecto se centra en la realización de un establecimiento del plano de formas de un buque pesquero de cerco de 15 metros de eslora, realizando las formas de dicho barco con el software Rhinoceros 4.0.

Después obtendremos toda la información necesaria de nuestro barco en cuanto a las formas, y crearemos un programa con MATLAB que nos calculará de forma automática todas las curvas hidrostáticas, en su condición de buque adrizado (volumen de carena, centro de carena, BMt…). Mostrándose: la realización del plano de formas en 3D, conocimientos matemáticos necesarios para la compresión posterior de MATLAB, breve introducción a MATLAB, cómo se ha realizado dicho programa.

En el ANEXO I se muestran todos los datos de partida.

Los datos de partida son los siguientes:

1- Información general del buque
2- Experiencia de Estabilidad
3- Plano de formas y plano de disposición general del barco

El esquema a trabajar será el siguiente:

1- Establecimiento del plano de formas
2- Conocimientos necesarios para la compresión de MATLAB
3- Introducción a MATLAB
4- Realización del programa de las curvas hidrostáticas
5- Valoración de los resultados

Cabe decir que dicho programa será válido para cualquier tipo de buque, por lo que podrá ser utilizado como medio de aprendizaje para futuros alumnos.

1.2- Objetivo

El objetivo a alcanzar en este proyecto es el de lograr completar una serie de tareas, algunas, cursadas durante la carrera de Ingeniería Técnica Naval, como la realización de las formas de un buque en 3D, y otras, como la programación con MATLAB, no habiendo sido vistas en la carrera, aprender a utilizar y manejar dicho programa, con el objeto de poder avanzar en mi aprendizaje y que a la hora de trabajar en el sector naval me sean útiles. La programación con MATLAB es bastante compleja, por lo que, el objetivo será comprender al máximo como utilizar este programa.

2- Plano de formas y modelado en 3D

La representación grafica de las formas del casco del buque se realiza de acuerdo con las normas del sistema de planos acotados, con las vistas proyectadas sobre los tres planos de un sistema de ejes cartesianos, y reciben el nombre de planos de formas.

Los tres planos ortogonales son los siguientes:

- El plano longitudinal, coincidente con el plano de simetría del buque o plano de crujía. Sobre este plano se proyectan las secciones longitudinales correspondientes a la intersección de la carena con planos paralelos al longitudinal. Una buena práctica es la de utilizar tres secciones longitudinales además del plano diametral.

- El plano horizontal, coincidente con el plano de la superficie del agua o plano de la flotación. Sobre este plano se proyectan las secciones horizontales o líneas de agua correspondientes a la intersección de la carena con planos paralelos al horizontal. Una buena práctica es utilizar siete líneas de agua equi-espaciadas hasta la flotación de proyecto, comenzando con la línea de agua 0, coincidente con el plano base, y utilizando alguna línea adicional a mitad del intervalo como línea de agua.

- El plano transversal, perpendicular a los dos anteriores, donde se proyectan las secciones transversales o cuadernas de trazado correspondientes a la intersección de la carena con planos paralelos al plano transversal. Una buena práctica es trazar 21 secciones equi-espaciadas comenzando a enumera por la 0 la correspondiente a la perpendicular de popa, y terminando con el 20 en la perpendicular de proa.

Adjunto el plano de formas realizado por el Ingeniero Naval, Juan Domínguez Caparrós, en el astillero Nicolás Casas, S. L. de Adra, junto con sus características principales, así como la información general del buque en el ANEXO I.

2.1- Superficie NURBS y modelado del casco.

Para la creación del plano de formas se usará el programa de diseño Rhinoceros, el cual ha sido visto y utilizado durante el segundo curso de carrera en la asignatura de Dibujo Naval por el profesor Leandro Ruiz Peñalver. El procedimiento será el siguiente:

- Calcar el plano de formas original (líneas de agua, secciones, longitudinales, casco…) para la posterior creación de una superficie NURBS.
- Reconstruir y alisar la superficie NURBS acercándose lo máximo posible al modelo de partida.
- Construcción del espejo, la quilla y el soporte del timón.
- Proyección en el plano de crujía de los cortes del casco con los planos definidos.

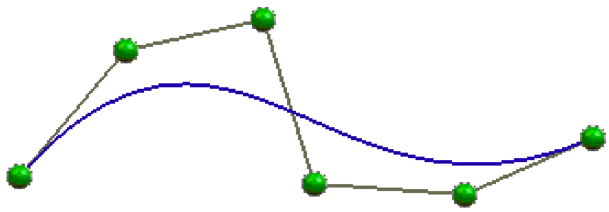
En primer lugar definiremos lo que es una superficie NURBS.

2.1.1-Curvas y superficies NURBS

El desarrollo estas curvas empezó en 1950 por ingenieros que necesitaban la representación matemática precisa de superficies de forma libre como las usadas en carrocerías de automóviles, superficies de exteriores aeroespaciales y cascos de barcos, que pudieran ser reproducidos exacta y técnicamente en cualquier momento.

Las anteriores representaciones de este tipo de diseños sólo podían hacerse con modelos físicos o maquetas realizadas por el diseñador o ingeniero.

Los pioneros en esta investigación fueron Pierre Bézier quien trabajaba como ingeniero en Renault, y Paul de Casteljau quien trabajaba en Citroën, ambos en Francia. Bézier y Casteljau trabajaron casi en paralelo, aunque ninguno de los dos conoció el trabajo que el otro desarrollaba. Bezier publicó primero sus trabajos y por esta razón tradicionalmente se le ha asociado a las Bézier-Splines (splines de Bézier, que son representadas con puntos de control describiendo a la curva misma), mientras que el nombre de Casteljau sólo es conocido por los algoritmos que desarrolló para la evaluación de superficies paramétricas. En la década de 1960 se desarrollaron las NURBS (B-splines Racionales No Uniformes), y se convirtieron en la generalización de las Bézier splines.



Las primeras NURBS fueron usadas en paquetes propietarios de CAD de las compañías automotrices. Posteriormente formaron parte del estándar en paquetes de gráficos por computadora. En 1985, el primer modelador interactivo de NURBS para PC, llamado Macsurf (posteriormente Maxsurf), fue desarrollado por Formation Design Systems, una pequeña compañía en Australia. Maxsurf es un sistema de diseño para cascos, que pretendía la creación de barcos, botes y yates, para los diseñadores quienes tenían la necesidad de alta precisión en el esculpido de superficies.

Las NURBS, B-splines racionales no uniformes, son representaciones matemáticas de geometría en 3D capaces de describir cualquier forma con precisión, desde simples líneas en 2D, círculos, arcos o curvas, hasta los más complejos sólidos o superficies orgánicas de forma libre en 3D. Gracias a su flexibilidad y precisión, se pueden utilizar modelos NURBS en cualquier proceso, desde la ilustración y animación hasta la fabricación.

La geometría NURBS tiene cinco cualidades esenciales que la convierten en la opción ideal para el modelado asistido por ordenador.

- Existen varias formas estándar industriales para intercambiar la geometría NURBS. Los usuarios pueden y deberían ser capaces de transportar todos sus modelos geométricos entre los diferentes programas de modelado, renderizado, animación e ingeniería de análisis que hay en el mercado.

- Las NURBS tienen una definición precisa y muy conocida. La geometría NURBS se enseña en las facultades de matemáticas e informática de las universidades más importantes. Eso significa que los vendedores de software especializado, los equipos de ingenieros, las empresas de diseño industrial y las empresas de animación que necesitan crear aplicaciones de software específicas para sus proyectos podrán encontrar programadores capacitados para trabajar con la geometría NURBS.

- Las NURBS pueden representar con precisión objetos geométricos estándar tales como líneas, círculos, elipses, esferas y toroides, así como formas geométricas libres como carrocerías de coches y cuerpos humanos.

- La cantidad de información que requiere la representación de una forma geométrica en NURBS es muy inferior a la que necesitan por separado las aproximaciones comunes.

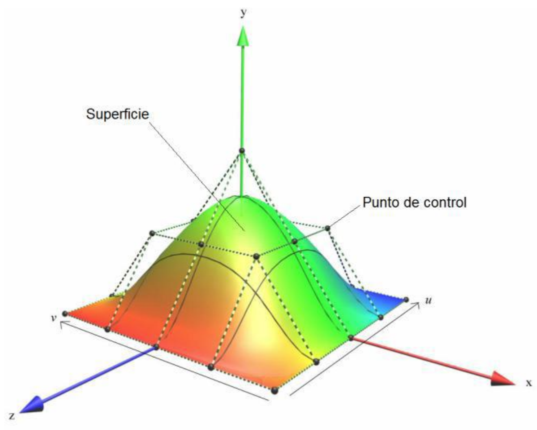
- La regla de cálculo de las NURBS, que se describe a continuación, se puede implementar en un ordenador de manera eficaz y precisa.

Para mejor comprensión de las curvas y superficies que vamos a emplear, desglosaremos los componentes que las definen, se trata del grado y los puntos de control.

Un grado es un número entero positivo. Este número normalmente es 1, 2, 3 o 5, pero puede ser cualquier número entero positivo. Las líneas y polilíneas NURBS de Rhino son grado 1, los círculos de Rhino son grado 2 y la mayoría de las formas libres de Rhino son grado 3 o 5. A veces se utilizan los siguientes términos: lineal, cuadrático, cúbico y quíntico. Lineal significa de grado 1, cuadrático significa de grado 2, cúbico significa de grado 3 y quíntico significa de grado 5.

Es posible que vea referencias del orden de una curva NURBS. El orden de una curva NURBS es un número entero positivo igual a (grado+1). En consecuencia, el grado es igual a orden-1.

Existe la posibilidad de incrementar los grados de una curva NURBS sin cambiar su forma. Generalmente, no es posible reducir el grado de una curva NURBS y no cambiar su forma.



Los puntos de control son una lista de puntos de grado+1 como mínimo. Una de las maneras más sencillas de cambiar la forma de una curva NURBS es mover los puntos de control.

Los puntos de control tienen un número asociado denominado peso. Con algunas excepciones, los pesos son números positivos. Cuando todos los puntos de control de una curva tienen el mismo peso (normalmente 1), la curva se denomina no racional; de lo contrario, se trataría de una curva racional. En NURBS, la R significa racional e indica que una curva NURBS tiene la posibilidad de ser racional. A la práctica, la mayoría de las curvas NURBS son no-racionales. Algunas curvas, círculos y elipses NURBS, ejemplos significativos, son siempre racionales.

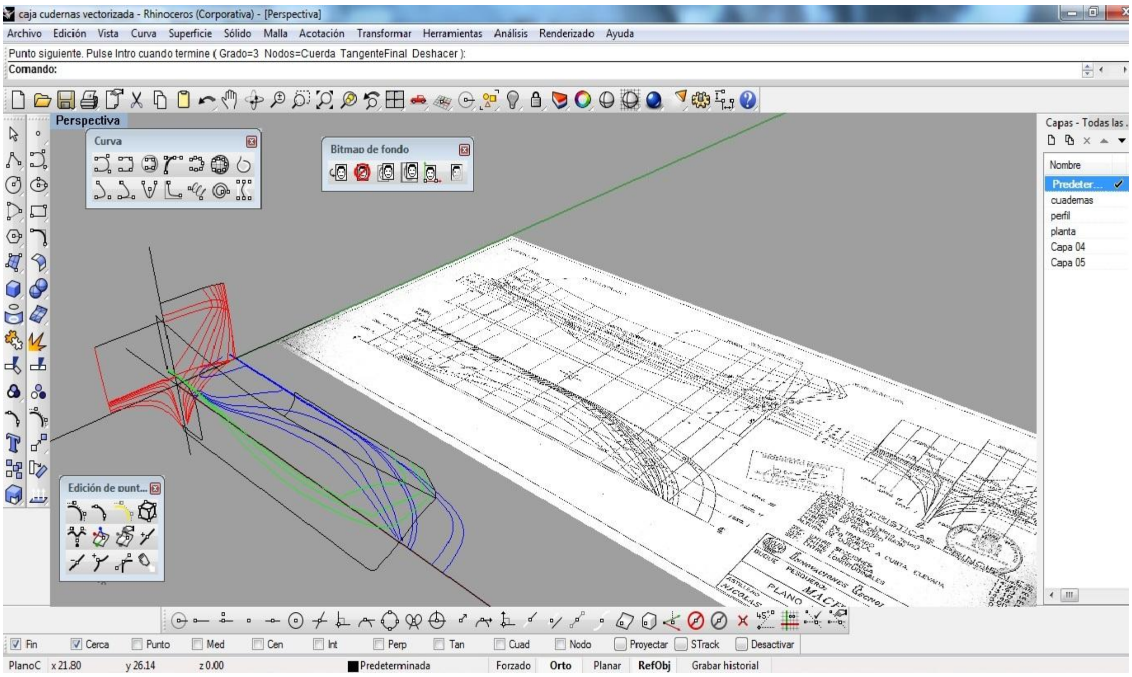
2.2- Modelado del casco

2.2.1-Calcar el plano de formas original

En Rhinoceros, se puede poner una imagen bitmap como imagen de fondo en el plano de trabajo. Partiendo de esta imagen, se pueden calcar las líneas del plano de formas, esto correspondería informáticamente a los junquillos que se utilizaban en la antigüedad cuando los planos de formas se realizaban a mano. Comenzamos con la herramienta de distribución de capas, la cual usaremos para crear varias capas en correspondencia con los planos de trabajo, por ejemplo, crearemos una capa para el plano de perfil en el que insertaremos nuestra imagen de fondo bitmap del perfil del buque. Esta herramienta nos permitirá en cualquier momento visualizar o ocultar cada una de las capas, es decir, cada una de nuestras líneas del plano de formas, por lo que nos será más cómodo trabajar y la probabilidad de error será menor.

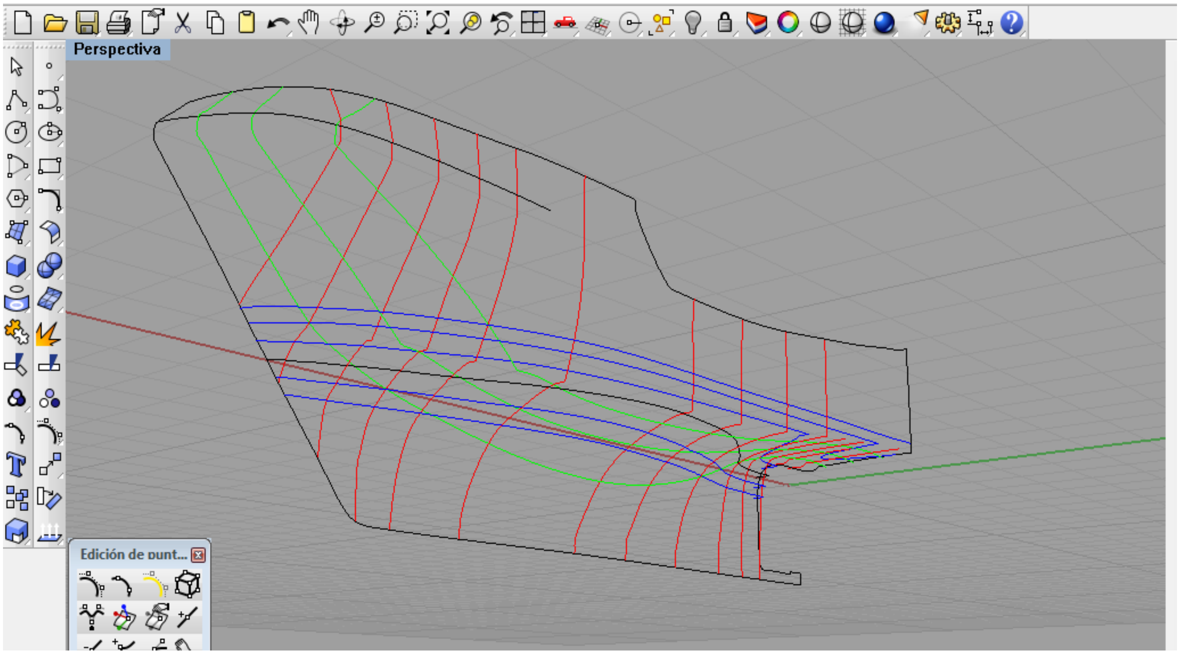
Con la imagen bitmap de fondo insertada será necesario escalarla a las medidas originales del barco y alinearla tal que la línea base quede como una línea totalmente horizontal.

Es preciso intentar aproximar estas líneas con curvas que tengan el menor número de puntos de control posible para simplificar el proceso de alisado después. Calcaremos en cada vista las líneas correspondientes del plano de formas, tanto las líneas de referencia, como las líneas de agua, longitudinales o secciones.



Una vez calcadas todas las líneas es necesario trasladarlas a su lugar correspondiente. El origen de coordenadas se encontrará en la intersección de la perpendicular de popa con la línea de base. Cada sección, longitudinal y línea de agua se deberán colocar donde correspondan.

Obtendremos una red de curvas que definirán las formas del casco.



Utilizaremos esta red para crear con Rhinoceros una superficie NURBS que se acerque a esta forma. En esta primera fase no hace falta conseguir una precisión exagerada, ya que la superficie que vamos a obtener será modificada hasta obtener la forma deseada y deberá ser alisada. Es importante obtener unas curvas correctas y precisas desde el punto de vista de las formas del casco, cuando cada una de estas curvas son colocadas deben de coincidir en cada una de las tres vistas. Para esto se aconseja trazar líneas de referencia y comprobar que coinciden en la vista 3D.

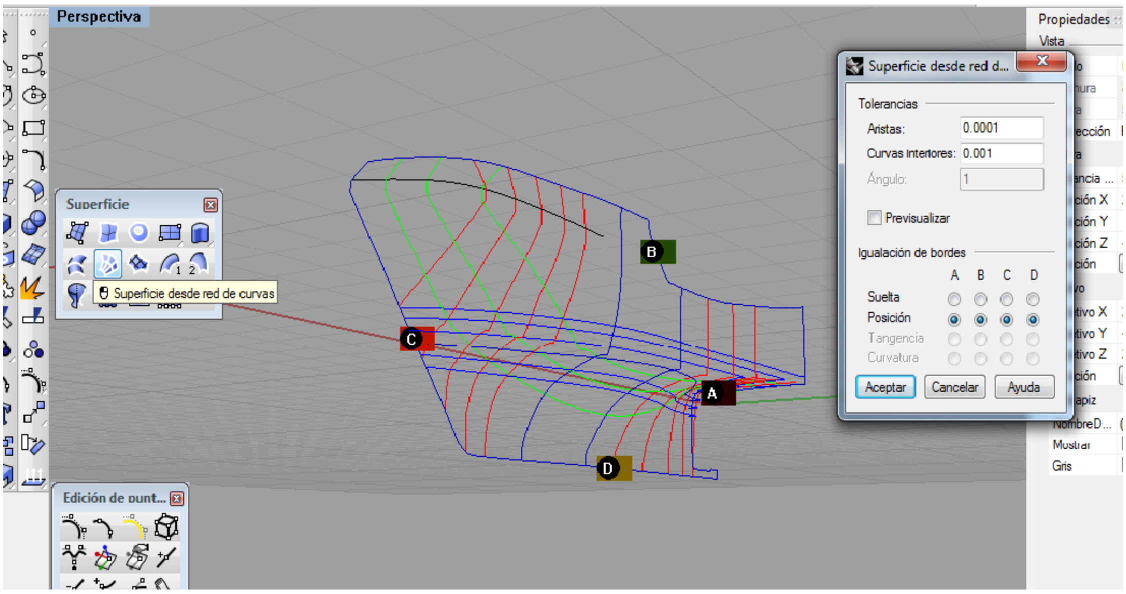
2.2.2- Creación de una superficie NURBS en Rhinoceros.

A partir de la red de curvas que hemos creado, vamos poder crear una superficie NURBS. Usaremos la herramienta *superficie para red de curvas,* comando "NetworkSrf".

Para utilizar esta herramienta debemos tener en cuenta una serie de condiciones de validez, que son:

- Las curvas deben formar una rejilla cuadrada.
- Todas las curvas en una dirección tienen que cruzar a todas las curvas en la otra dirección y no pueden cruzarse una con la otra.

Seleccionamos unas curvas de manera inteligente tal que cumplan estas condiciones y que sean representativas de las formas del casco.



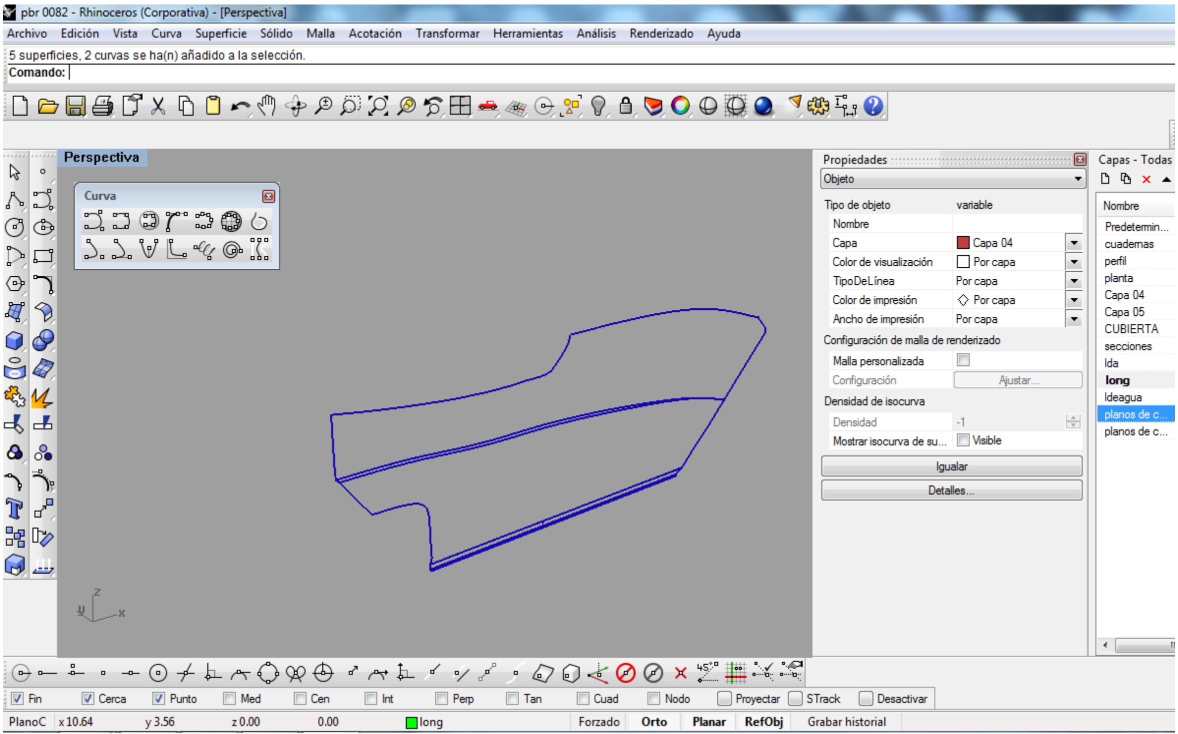
Rhinoceros nos propone una superficie que se aproxima a la red de curvas que hemos seleccionado dentro de las líneas del casco.

Esta superficie obtenida no será exactamente la que queremos, tendrá probablemente demasiados puntos de control, lo que complica el proceso de alisado.

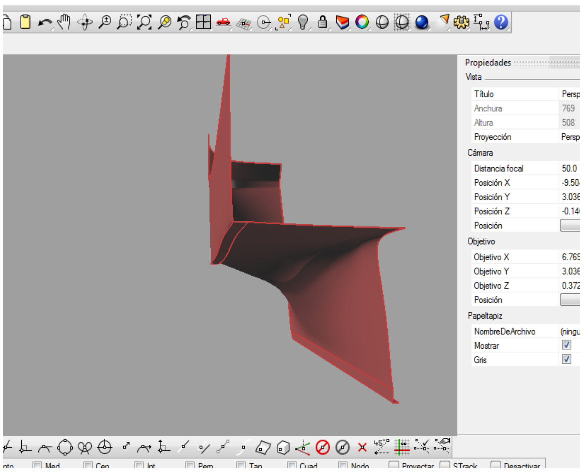
También es posible que en ciertas zonas, esta superficie esté demasiado lejos de las líneas que habíamos calcado. Moviendo los puntos de control intentaremos que la superficie se acerque lo máximo posible a las líneas calcadas.

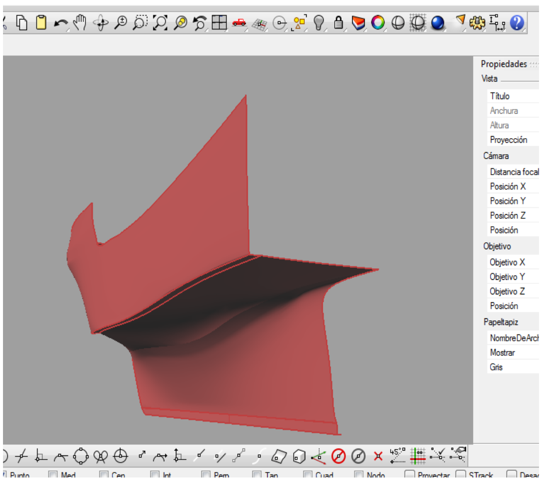
Primera superficie obtenida, simplificaremos la superficie reconstruyéndola con menos puntos de control (Edición → Reconstruir)

Por la complejidad de las formas del buque hemos decidido crear varias superficies para que al mover los puntos de control nos resulte más fácil que dichas superficies se asemejen lo máximo posible a las líneas que habíamos calcado. En la siguiente imagen se muestran las superficies creadas en nuestro barco.

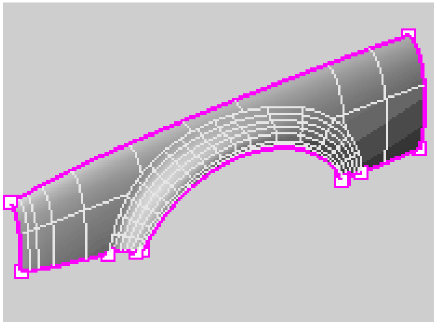


Hemos creado una superficie para la parte del costado hasta el codillo y otra, para la parte inferior del casco, en donde las formas se nos complicaban ya que pasábamos de tener un costado con una semi-manga de 2,5 metros a tener, en la parte inferior del casco, unas formas muy finas. En las siguientes imágenes podemos ver dichas formas.





Para la unión de dichas superficies se necesita cumplir un requisito fundamental, que los bordes desnudos de las superficies a unir, estén unidos.



Un borde desnudo es un borde de superficie o poli superficie que no está conectado a otro borde. Los objetos sólidos no tienen bordes desnudos.

Unir los bordes no modifica la geometría subyacente. Simplemente "pega" superficies adyacentes para que el mallado, las operaciones booleanas y las intersecciones atraviesen la costura sin que haya aberturas.

Para modificar una geometría de superficie con el fin de rellenar una abertura, podremos utilizar el comando *IgualarSup* o para rellenar la abertura con una nueva superficie creada usaremos los comandos *EmpalmarSup, MezclarSup, MezclarBorde, EmpalmarBorde, SupDesdeRed* o *Parche.*

Para cambiar dos superficies adyacentes y convertirlas en una sola superficie, está el comando *FusionarSup*.

Una vez unidas todas las superficies comenzaremos el proceso de alisado.

2.2.3- Proceso de alisado.

En Rhinoceros existen varias herramientas para controlar el estado de una superficie: hay herramientas matemáticas y herramientas visuales.

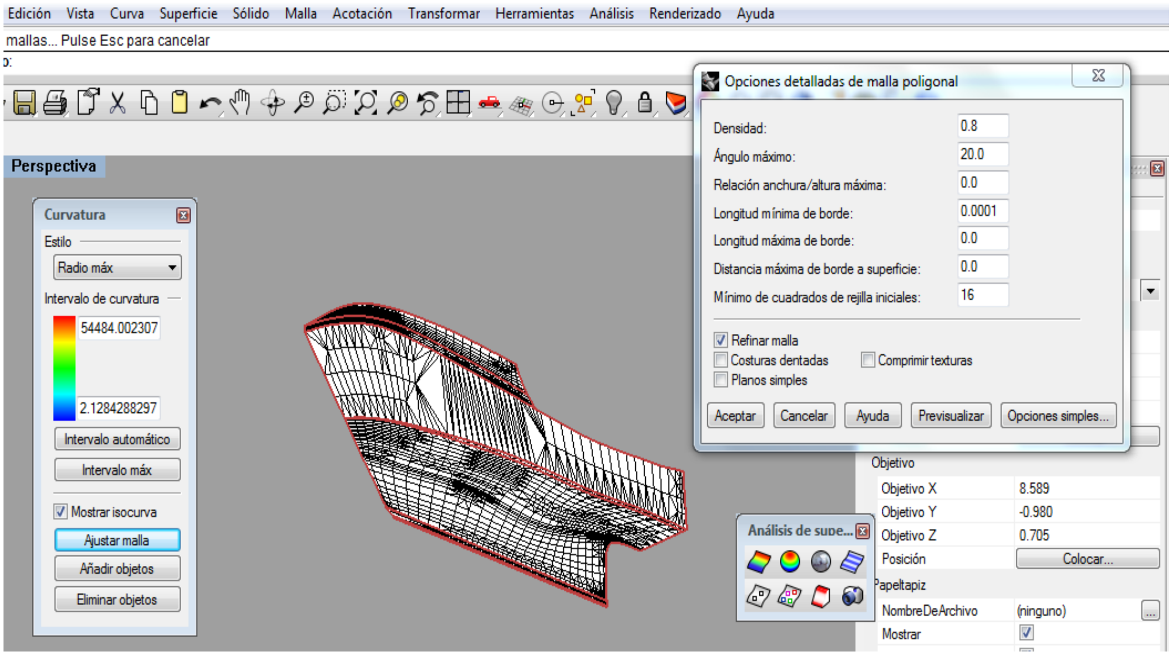
Dentro de las matemáticas está la curvatura de Gauss que permite apreciar la curvatura de una superficie. Se define la Curvatura de Gauss en un punto de una superficie como:

$$\frac{1}{R_{max}} \cdot \frac{1}{R_{min}}$$

- La curvatura de Gauss es positiva en un punto de la superficie cuando la tangente a la superficie en dicho punto toca a la superficie solamente en este punto.

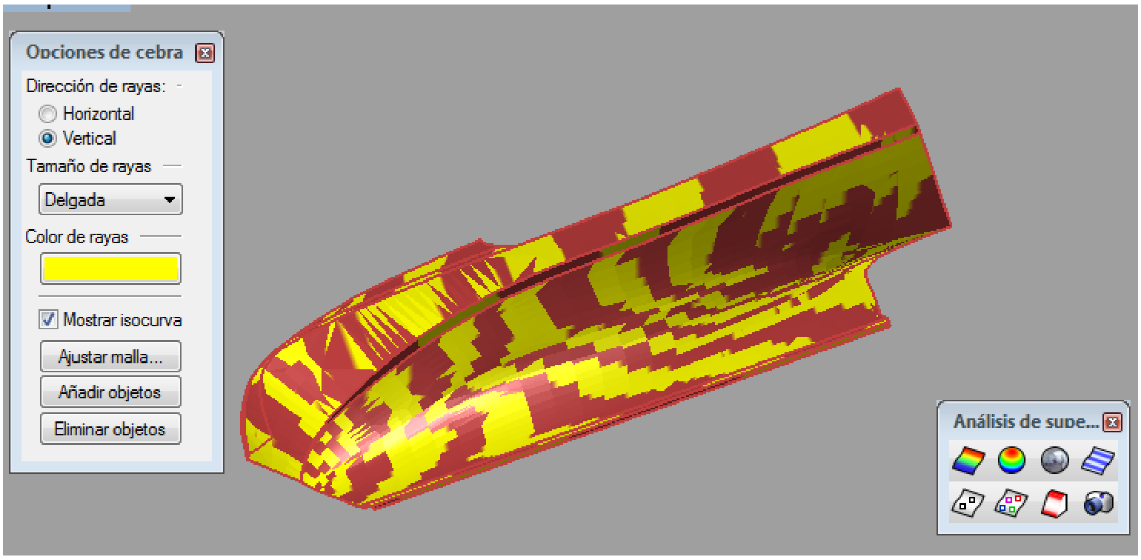
- Es negativa cuando dicha tangente toca en más de un punto.
- Es nula cuando se puede trazar una generatriz recta sobre la superficie.

Con esta herramienta podemos notar rápidamente las inversiones de curvatura, pero para los otros defectos de alisado, es difícil de interpretar.

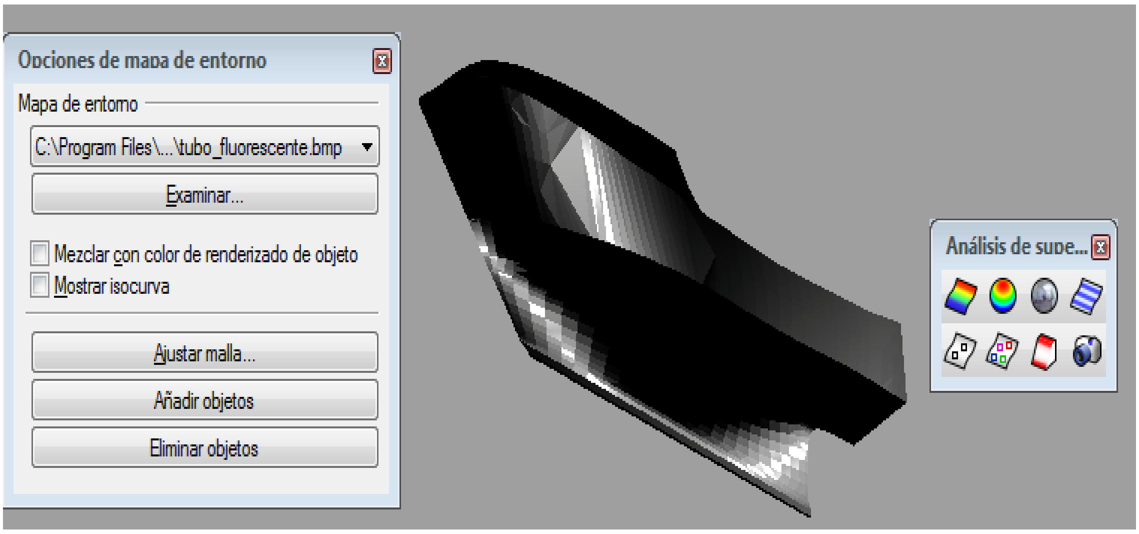


Por su aspecto práctico y directo hemos preferido usar herramientas visuales. Rhinoceros nos permite usar varias herramientas como cebras o luz de alambre.

El comando *cebra* es uno de los comandos visuales de análisis de superficies. Estos comandos utilizan cálculos de superficies NURBS y técnicas de renderizado para ayudar a analizar visualmente la suavidad, curvatura y otras características importantes de la superficie.



La herramienta de la luz de alambre revela con mucha precisión el estado de superficie, lo que hace esta herramienta es reflejar sobre la superficie un fino rayo de luz que esposa la forma del casco en el que se refleja.



El comando *MapaE* es uno de los comandos visuales de análisis de superficies. Estos comandos utilizan cálculos de superficies NURBS y técnicas de renderizado para ayudarle a analizar visualmente la suavidad, curvatura y otras características importantes de la superficie.

Si cuando utilizamos el comando *MapaE*, cualquiera de los objetos seleccionados no tiene una malla de análisis de superficie, se creará una malla invisible basada en los parámetros del cuadro de diálogo Opciones de malla poligonal.

Las mallas de análisis de superficies se guardan en los archivos de Rhino. Estas mallas pueden ser grandes. El comando *ActualizarSombreado* y la opción *Guardar sólo geometría* de los comandos *Guardar* y *GuardarComo* eliminan las mallas de análisis de las superficies existentes.

Para analizar correctamente una superficie NURBS de forma libre, el comando de análisis requiere generalmente una malla detallada.

El mapeado de entorno es un estilo de renderizado que hace parecer como si una escena estuviera reflejada por un metal muy pulido. Puede que haya varios casos en que el mapeado del entorno muestre realmente un defecto en la superficie que no se pueda observar cuando se usa el comando *Cebra* y se rota la escena.

La superficie de costado de nuestro barco es casi una superficie plana por lo que el alisado es fácil de conseguir actuando sobre los puntos de control. En la superficie inferior también actuaremos sobre los puntos de control para conseguir un correcto alisado. El principio base es: una malla regular de los puntos de control genera una superficie lisa. Por esto hay que mover los puntos hasta conseguir tener una malla lo más ordenada y homogénea posible.

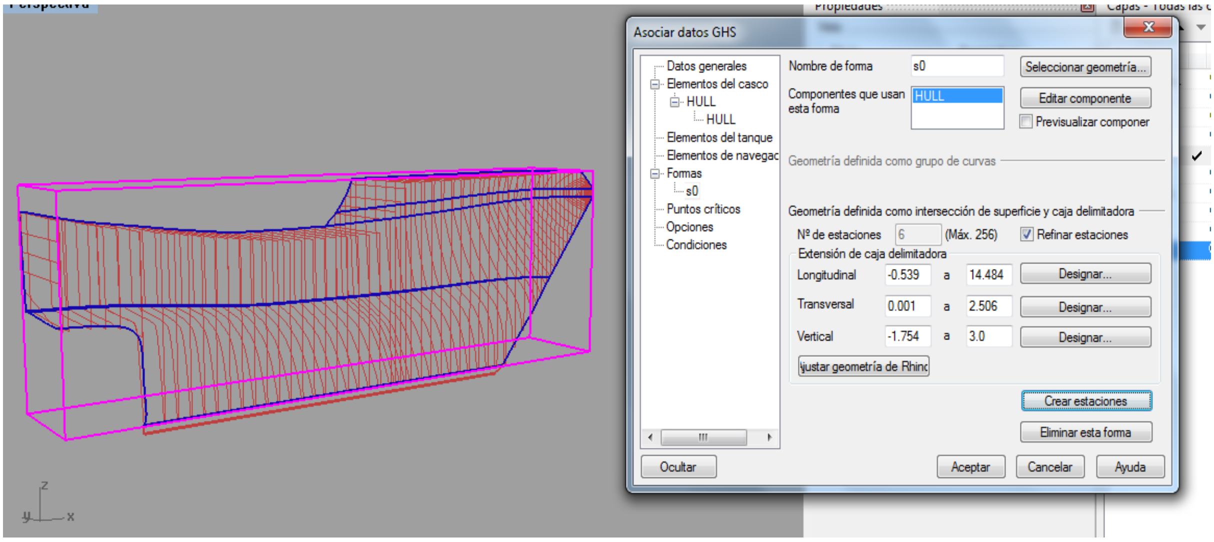
2.2.4- Obtención de datos de las formas del buque

Para la obtención de la información de las formas de nuestro buque, utilizaremos una herramienta de Rhinoceros llamada *Asociar datos GHS.* Lo que hace esta herramienta es añadir información GHS específica a un modelo de Rhino marcando la geometría como objetos GHS. Los archivos de geometría GHS se utilizan para analizar cascos flotantes (botes, barcos, embarcaderos).

La creación de un archivo GHS consta de dos pasos:

- Utilizamos el comando *AsociardatosGHS* para definir las partes del casco, una vez que tengamos añadidos los elementos que forman el casco, crearemos las formas donde deben añadirse las secciones. Podemos editar tanto las formas, como el número de secciones que queremos, pero a nosotros lo que nos interesa es tener el mayor número de secciones posible, para que las formas del barco queden mejor definidas y así, a la hora de hacer los cálculos de áreas/volúmenes y centros de gravedad, sean lo cálculos muy precisos.

- Una vez creadas las secciones, procedemos a obtener el archivo con toda la información. Utilizamos el comando *GuardarComo* y en el apartado *tipo* seleccionamos “Archivo de geometría GHS (*gf)”.



Abriendo nuestro archivo con el bloc de notas, encontramos una estructura en la información de nuestro barco bastante curiosa, la cual explicamos seguidamente con una imagen explicativa, Figura 3.1.

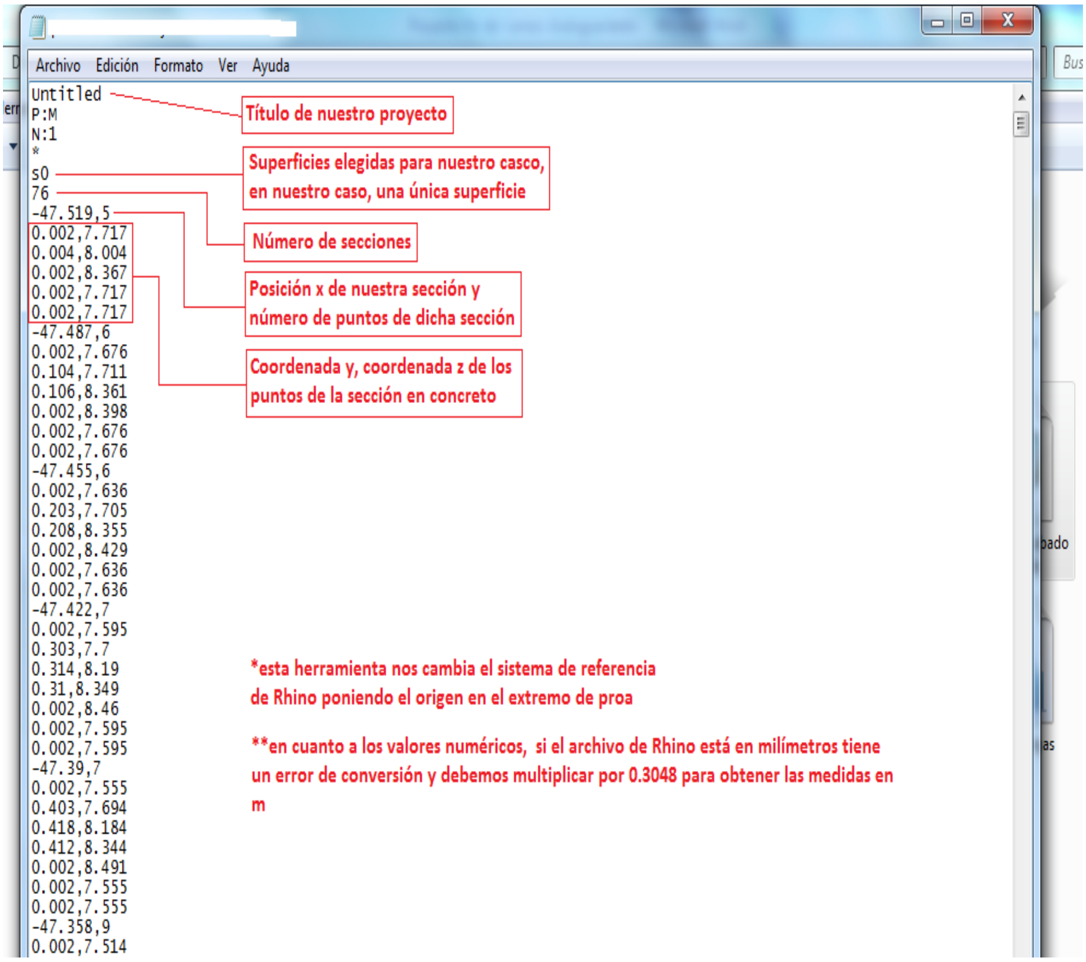


Figura 3.1 – Imagen explicativa del archivo *gf

3- Propiedades geométricas de la carena.

La generalidad de los cálculos de geometría de la carena puede resolverse mediante la integración definida de funciones. En general la expresión de estas funciones no es conocida lo que impide realizar la integración utilizando el cálculo integral directo.

No obstante al disponer de la representación grafica de estas funciones, podemos conocer el valor que toma la función para determinados valores de la variable midiéndolo directamente en la representación grafica de la función o calculando su valor, lo que permite el cálculo de la integral definida mediante la aplicación de métodos numéricos aproximados.

Antes de comenzar con los cálculos de las propiedades geométricas de curvas planas y volúmenes será necesario hacer una serie de definiciones necesarias para la correcta interpretación de los mismos.

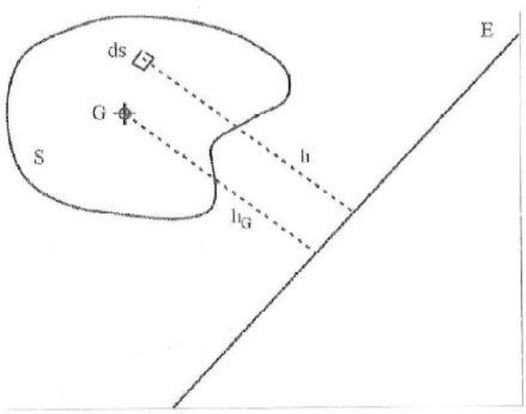
3.1- Momentos estáticos y centros de gravedad

3.1.1- Momentos estáticos y centros de gravedad de un área plana

Se denomina momento estático de una superficie plana (S) respecto a un eje (E), tal como se indica en la figura siguiente, a la integral del producto de cada elemento diferencial de superficie (dS) por su distancia al eje considerado.

$$M_E = \int_s h * ds$$

Se define el centro de gravedad de una superficie como el punto donde, supuesta concentrad el área de la superficie, se verifica que su momento estático respecto a cualquier eje es igual al momento estático de la superficie respecto al mismo eje. Por tanto la distancia del centro de gravedad a un eje cualquiera (E) viene determinado por la relación:



$$M_E = h_G * \int_s ds = \int_s h * ds$$

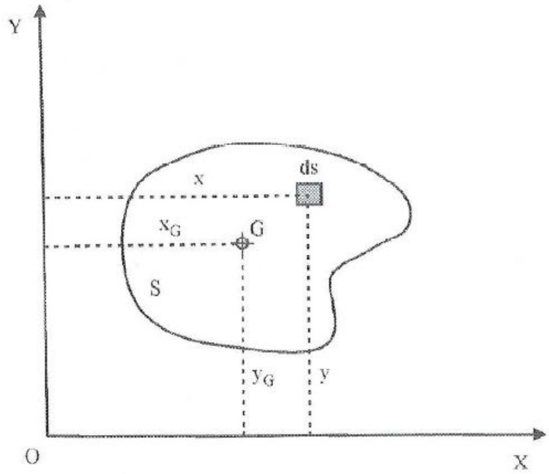
$$h_G = \frac{M_E}{S} = \frac{\int_s h * ds}{\int_s ds}$$

$\int_s ds$ = S (área de la superficie)

Para determinar la posición del centro de gravedad de una superficie se necesita dos coordenadas, por lo que es necesario tomar momentos estáticos respecto a dos ejes. En el caso de que los dos ejes sean ortogonales como se indica en la siguiente figura:

$$X_G = \frac{M_{oy}}{S} = \frac{\int_s x * ds}{\int_s ds}$$

$$Y_G = \frac{M_{ox}}{S} = \frac{\int_s y * ds}{\int_s ds}$$

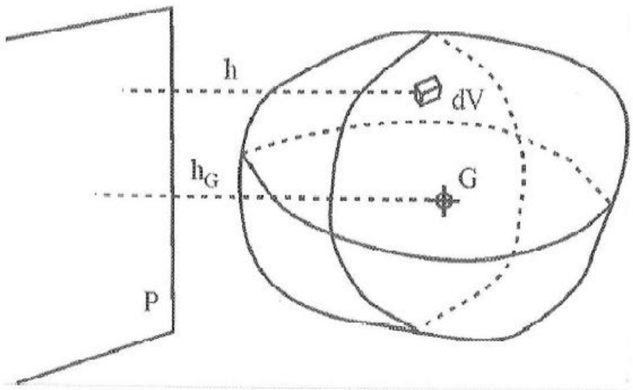


3.1.2- Momentos estáticos y centros de gravedad de un volumen

Se define el momento estático de un volumen (V) respecto a un plano (P), tal como se indica en la figura siguiente, a la integral del producto de cada elemento diferencial de volumen (dV) por su distancia al plano (h):

$$M_P = \int_V h * dv$$

Se define el centro de gravedad del volumen como el punto donde concentrado el volumen considerado, se verifica que su momento estático respecto a cualquier plano es igual al momento estático del volumen respecto al mismo plano. Por tanto la distancia del centro de gravedad a un plano cualquiera (P) viene determinado por la relación:



$$M_P = h_G * \int_V dv = \int_V h * dv$$

$$h_G = \frac{M_P}{V} = \frac{\int_V h * dv}{\int_V dv}$$

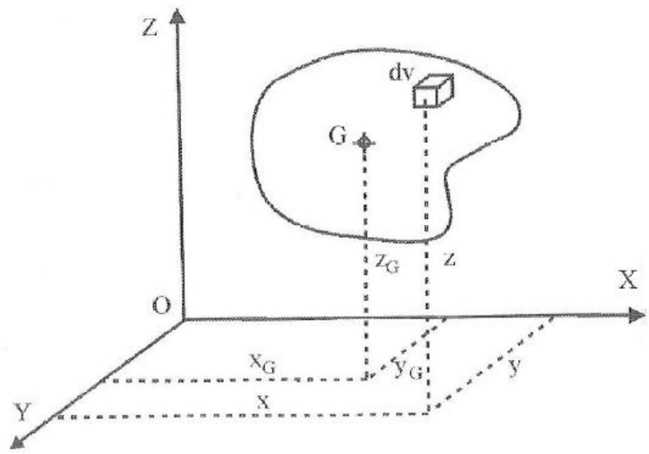
$\int_V dv$ = V (volumen de la superficie)

Para determinar la posición del centro de gravedad de un volumen se necesitan tres coordenadas, por lo que es necesario tomar momentos estáticos respecto a los tres planos. En el caso de que los tres planos sean ortogonales como se indica en la figura siguiente:

$$X_G = \frac{M_{YOZ}}{V} = \frac{\int_V x * dv}{\int_V dv}$$

$$Y_G = \frac{M_{XOZ}}{V} = \frac{\int_V y * dv}{\int_V dv}$$

$$Z_G = \frac{M_{XOY}}{V} = \frac{\int_V z * dv}{\int_V dv}$$



3.2- Momentos de Inercia

3.2.1- Momento de inercia de un área plana

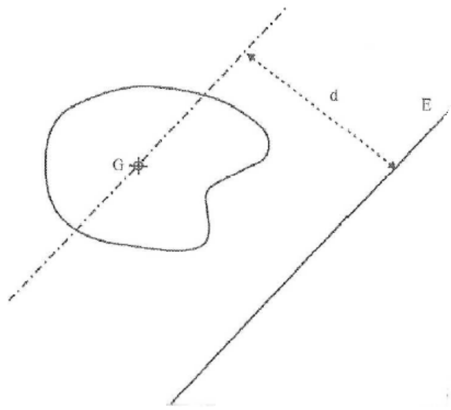
Se define el momento de inercia de una superficie plana (S) respecto a un eje (E) como la integral del producto de cada elemento diferencial de área (dS) por el cuadrado de la distancia al eje considerado (h):

$$I_E = \int_S h^2 * ds$$

El momento de inercia, que recibe también el nombre de momento de segundo orden, tiene siempre un signo positivo.

3.2.2- Teorema de Steiner

El momento de inercia respecto a cualquier eje (I_E) es igual al momento de inercia respecto a un eje paralelo que pasa por el centro de gravedad del área (I_G) aumentado en el producto del área de la superficie (A) por el cuadrado de la distancia entre ambos ejes (d), tal como se indica en la figura siguiente:



$$I_E = I_G + A * d^2$$

Una vez realizadas las definiciones oportunas estableceremos los cálculos más frecuentes de geometría de la carena que se van abordas con ayuda del calculo numérico aproximado, tanto en el caso de las figuras planas como el de la figuras en el espacio que se exponen a continuación.

3.3- Figuras planas

Calculo del área plana encerrada por la curva de ecuación: *y=f(x)*, el eje de abscisas y las ordenadas extremas x_1 y x_2 tal y como se indica en la figura siguiente:

3.3.1- Área

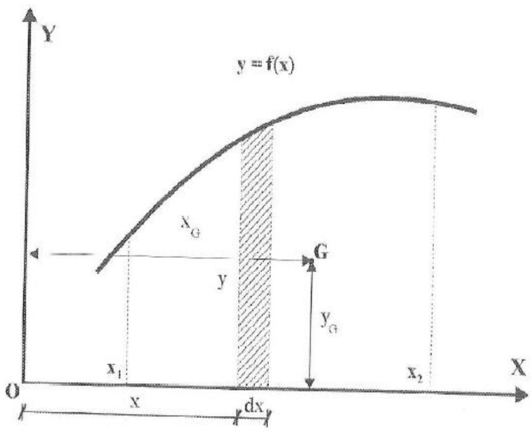
Diferencial de área:

$$dA = y * dx$$

Área entre los límites x_1 y x_2:

$$A = \int_{x1}^{x2} y * dx$$

Función a integrar: *y*



3.3.2- Momento estático o de primer orden respecto del eje OY

Diferencial de momento:

$$dM_{OY} = x * dA = x * y * dx$$

Momento entre los límites x_1 y x_2:

$$M_{OY} = \int_{x1}^{x2} x * \ y * dx$$

Función a integrar: $x * y$

3.3.3- Momento estático o de primer orden respecto al eje OX

Diferencial de momento:

$$dM_{OX} = \frac{y}{2} * dA = \frac{1}{2} * y^2 * dx$$

Momento entre los límites x_1 y x_2:

$$M_{OX} = \int_{x1}^{x2} \frac{1}{2} * y^2 * dx$$

Función a integrar:

$$\frac{1}{2} * y^2$$

3.3.3- Centro de gravedad del área, centro de área, centroide o baricentro

$$X_G = \frac{M_{OY}}{A} = \frac{\int_{x1}^{x2} x * \ y * dx}{\int_{x1}^{x2} y * dx}$$

$$Y_G = \frac{M_{OX}}{A} = \frac{1}{2} * \frac{\int_{x1}^{x2} \frac{1}{2} * y^2 * dx}{\int_{x1}^{x2} y * dx}$$

$X_G \ e \ Y_G$: Distancias del centro de gravedad del area de los ejes OY y OX respectivamente.

3.3.4- Momento de inercia o momento de segundo orden respecto al eje OY

Aplicando el teorema de Steiner al rectángulo diferencial:

Momento de inercia diferencial:

$$dI_{OY} = \frac{1}{12} * dx^3 * y + (x + \frac{dx}{2})^2 * dA = x^2 * y * dx$$

Momento de Inercia entre los límites x_1 y x_2:

$$I_{OY} = \int_{x1}^{x2} x^2 * y * dx$$

Función integrar:

$$x^2 * y$$

3.3.5- Momento de inercia o de segundo orden respecto al eje OX

Momento de inercia diferencial:

$$dI_{OX} = \frac{1}{12} * y^3 * dx + y * dx * (\frac{y}{2})^2 = \frac{1}{3} * y^3 * dx$$

Momento de Inercia entre los límites x_1 y x_2:

$$I_{OX} = \int_{x1}^{x2} \frac{1}{3} * y^3 * dx$$

Función integrar:

$$\frac{1}{3} * y^3$$

3.4- Figuras del espacio

3.4.1- Volumen encerrado por una superficie y unos planos limítrofes

Sea la superficie indicada en la figura, de la que conocemos la función del área de las secciones obtenidas por intersección con planos paralelos al plano OY: $A_{ZY} = f(x)$

Consideremos una rebanada transversal de área A_{ZY} y de espesor *dx.*

Volumen de la rebanada diferencia:

$$dV = \mathrm{A_{ZY}} * \mathrm{dx}$$

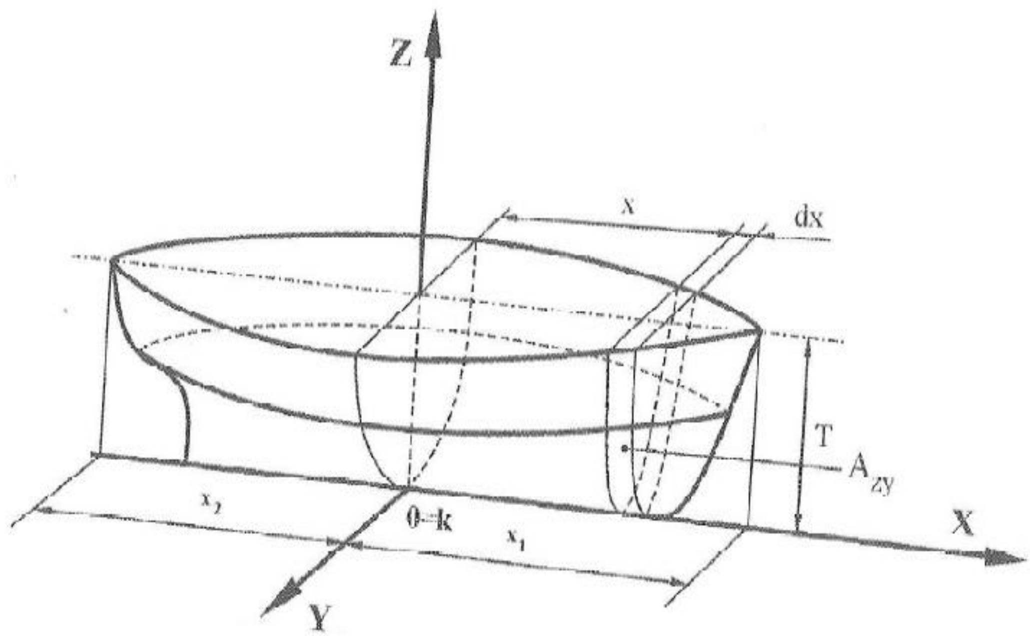
Volumen comprendido entre los planos definidos por x_1 y x_2:

$$V = \mathrm{A_{ZY}} * \mathrm{dx}$$

Función a integrar:

$$\mathrm{A_{ZY}} * \mathrm{dx}$$

El mismo volumen puede calcularse integrando rebanadas horizontales y longitudinales, es decir, integrando de forma análoga las áreas correspondientes a planos paralelos a los planos XOY: $\mathrm{A_{XY}}$ o XOZ: $\mathrm{A_{XZ}}$.



3.4.2- Momentos estáticos del volumen encerrado por una superficie y unos planos limítrofes

Respecto al plano XOY

Consideremos una rebanada horizontal del área $\mathrm{A_{XY}}$ y espesor *dZ.*

Momento estático de la rebanada diferencial:

$$dM_{XOY} = A * z * dz$$

Momento estático del volumen comprendido entre los planos definidos por z_1 y z_2:

$$M_{XOY} = \int_{z1}^{z2} \mathrm{A_{XY}} * z * dz$$

Función a integrar:

$$\mathrm{A_{XY}} * z$$

Respecto al plano XOZ

Consideremos una rebanada horizontal del área $\mathrm{A_{XZ}}$ y espesor *dY.*

Momento estático de la rebanada diferencial:

$$dM_{XOZ} = A * y * dy$$

Momento estático del volumen comprendido entre los planos definidos por y_1 e y_2:

$$M_{XOZ} = \int_{y1}^{y2} \mathrm{A_{XZ}} * y * dy$$

Función a integrar:

$$\mathrm{A_{XZ}} * y$$

Respecto al plano ZOY

Consideremos una rebanada horizontal del área $\mathrm{A_{ZY}}$ y espesor *dX.*

Momento estático de la rebanada diferencial:

$$dM_{ZOY} = A * x * dx$$

Momento estático del volumen comprendido entre los planos definidos por x_1 y x_2:

$$M_{ZOY} = \int_{x1}^{x2} \mathrm{A_{ZY}} * x * dx$$

Función a integrar:

$$\mathrm{A_{ZY}} * x$$

3.4.3- Coordenadas del centro de gravedad del volumen limitado por una superficie y planos limítrofes

$$X_G = \frac{M_{ZOY}}{V}$$

$$Y_G = \frac{M_{XOZ}}{V}$$

$$Z_G = \frac{M_{XOY}}{V}$$

4- Integración numérica.

4.1- Introducción

Muchos valores de interés para las ciencias se expresan a través de integrales, y algunas de esas integrales son difíciles de resolver analíticamente. Esto debe deberse a la complejidad de la función a integrar, al dominio de integración o a ambos.

Existen dos posibles pasos a seguir si no se puede llegar a una solución analítica de alguna integral.

1- Hacer algunas aproximaciones, haciendo algún parámetro pequeño, análisis asintótico etc.
2- Hacer la integración numérica.

Para el cálculo de áreas, volúmenes, centros de masas, momentos de inercia, etc. disponemos de fórmulas en las que interviene la integral definida de una función f

Si conocemos una primitiva F de la función continua f, $(\frac{\partial}{\partial\infty})\,F(x)= f(x)$, obtenemos el valor de la integral aplicando la regla de Barrow.

$$\int_a^b f(x)dx = F(b) - F(a)$$

En construcción naval en general, y en particular en los cálculos de flotabilidad y estabilidad, se requiere la obtención de los valores de integrales de funciones que dependen de las formas del buque.

Sin embargo, normalmente, en la cartilla de trazado del buque únicamente se da un conjunto discreto de puntos y la representación gráfica de las curvas del buque, pero no conocemos la expresión analítica de las funciones a integrar, por lo cual el concepto de primitiva carece de sentido.

Para calcular una aproximación de dichas integrales se recurre a distintos métodos de integración numérica.

4.2- Integración numérica

La integración numérica es una técnica que se puede usar para aproximar el valor de la integral de una función que no sea posible integrar.

Con el objeto de integrar numéricamente la integral comprendida en el intervalo cerrado [a, b], lo podemos hacer a través de dos métodos de integración numérica: la Regla del trapecio y la Regla de Simpson.

Hay varias razones para llevar a cabo la integración numérica. La principal puede ser la imposibilidad de realizar la integración de forma analítica. Es decir, integrales que requerirían de un gran conocimiento y manejo de matemática avanzada pueden ser resueltas de una manera más sencilla mediante métodos numéricos. Incluso existen funciones integrables pero cuya primitiva no puede ser calculada, siendo la integración numérica de vital importancia. La solución analítica de una integral nos arrojaría una solución exacta, mientras que la solución numérica nos daría una solución aproximada. El error de la aproximación, que depende del método que se utilice y de qué tan fino sea, puede llegar a ser tan pequeño que es posible obtener un resultado idéntico a la solución analítica en las primeras cifras decimales.

4.3- Métodos de integración numérica

4.3.1-Regla del Trapecio

La Fig. 1 muestra de color verde como sería el cálculo del área bajo la curva de la función f(x) entre los límites a y b si se dividiera dicha sub-área en un solo trapecio. El error que se cometería sería demasiado grande con respecto al área real que se desea obtener. Dependiendo de la forma de la curva el error que se cometería sería por exceso o por defecto. En el caso del ejemplo, el error seria por defecto, es decir el valor que arroje el cálculo de la integral será menor al valor real del área.

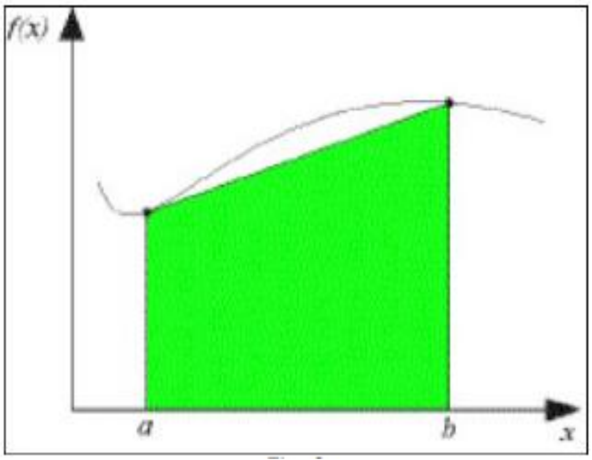


Fig. 1

Si se divide el intervalo (área a calcular) en más de una sub área, en el caso de la Fig. 2 (dividida en 3 sub áreas), el error en el cálculo de la integral o área total, se disminuye.

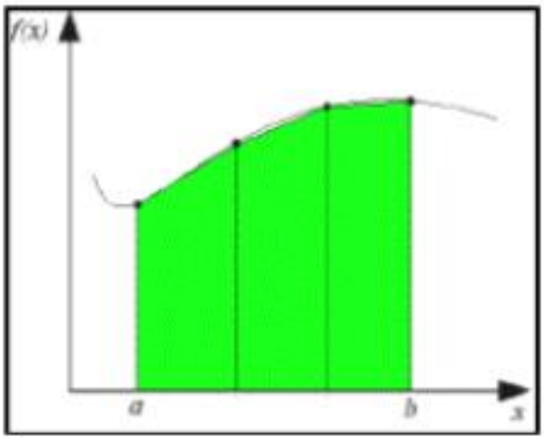


Fig. 2

La estrategia más simple y que evitaría menor error en el cálculo, consiste en subdividir el intervalo pedido para el cálculo del área en n sub intervalos de pequeño tamaño y aproximar el área como la suma de las áreas de cada uno de los trapecios que se forman:

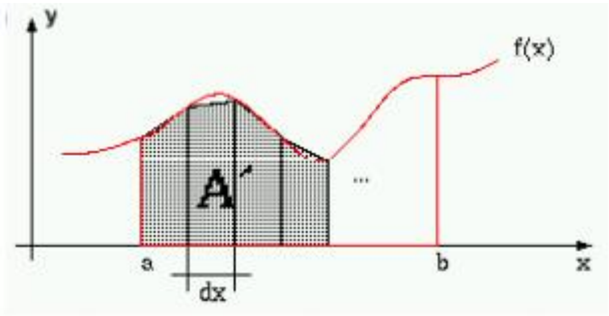


Fig. 3

De la Fig. 3 se puede deducir que,

$$\partial x = \frac{\mathrm{b} - \mathrm{a}}{\mathrm{n}}$$

si n es suficientemente grande (delta sería suficientemente pequeño), el área de los trapecios será aproximadamente el área pedida. El área total que correspondería a la suma del área de cada uno de los trapecios se calcula de la siguiente forma:

- Se determinan los puntos del eje x que delimitarán cada trapecio. Estos puntos son:
  xi= a+i*dx, con i= 0, 1, 2, ..., n
- Se evalúa la función f en cada uno de los puntos Xi:
  yi= f(xi), i= 0, 1, 2, ...,n
- Se calcula el área de cada trapecio como:
  ai= (yi+y(i+1))*dx/2, i= 0, 1, 2, ..., n-1
- Se suman las áreas de cada uno de los trapecios.

4.3.2 Método de Simpson

En integración numérica, Una forma de aproximar una integral definida en un intervalo [*a*,*b*] es mediante la regla del trapecio, es decir, que sobre cada sub-intervalo en el que se divide [*a*,*b*] se aproxima *f* por un polinomio de *primer grado*, para luego calcular la integral como suma de las áreas de los trapecios formados en esos sub-intervalos . El método utilizado para la regla de Simpson sigue la misma filosofía, pero aproximando los sub-intervalos de *f* mediante polinomios de segundo grado.

Primera Regla de Simpson

Esta regla lleva el nombre de Thomas Simpson, matemático británico que la utilizo a mitad del siglo XVIII, aunque ya había sido formulada en 1668 por James Gregory.

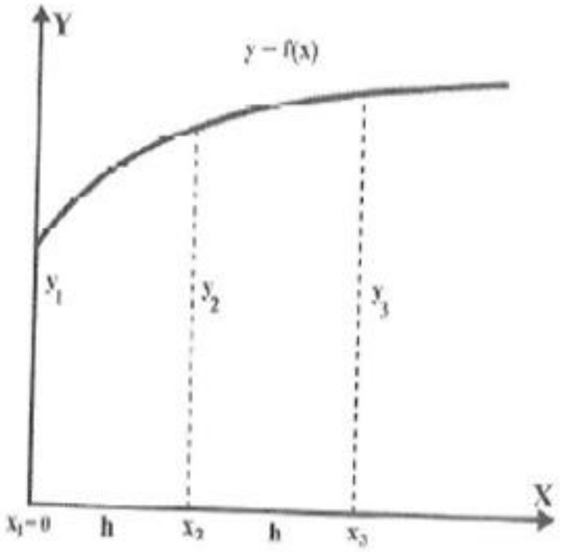
El método se emplea para calcular la integral definida de la función *y=f(x)* entre los límites x_1 y x_3, tal como se indica en la figura siguiente, se sustituye en ese tramo la función desconocida por una parábola de segundo grado de ecuación:

$$y = a_0 + a_1 * x + a_2 * x^2$$

$$\int_{x1}^{x3} y * dx = \int_{x1}^{x3} (a_0 + a_1 * x + a_2 * x^2) * dx$$

$$\int_0^{2h} (a_0 + a_1 * x + a_2 * x^2) * dx = \left[a_0 * x + a_1 * \frac{x^2}{2} + a_2 * \frac{x^3}{3} \right]_0^{2h}$$

$$= 2 * a_0 * h + a_1 * \frac{4 * h^2}{2} + a_2 * \frac{8 * h^3}{3}$$



De la función se conocen tres puntos que deberán de satisfacer la parábola que la sustituye, de donde se obtienen los valores de a_0, a_1 y a_2.

($x=0$; $y=y_1$)	$y_1 = a_0$
($x = h$; $y = y_2$)	$y_2 = a_0 + a_1*h + a_2*h^2$
($x =2*h$; $y = y_3$)	$y3 = a_0 + 2*a_1*h + 4*a_2*h^2$

Resolviendo este sistema obtenemos:

$$a_0 = y_1$$

$$a_1 = \frac{2 * y_2 - 1{,}5 * y_1 - 0{,}5 * y_3}{h}$$

$$a_2 = \frac{y_3 + y_1 - 2 * y_2}{2 * h^2}$$

Sustituyendo en la expresión de la integral:

$$\int_{x1}^{x3} y * dx = \int_{0}^{2h} (a_0 + a_1 * x + a_2 * x^2) * dx = \left[a_0 * x + a_1 * \frac{x^2}{2} + a_2 * \frac{x^3}{3}\right]_0^{2h}$$

$$= 2 * a_0 * h + a_1 * \frac{4 * h^2}{2} + a_2 * \frac{8 * h^3}{3} =$$

$$\int_{x1}^{x3} y * dx = \frac{h}{3}\ (y_1 + 4 * y_2 + y_3)$$

Veamos cómo podemos hacer extensivos estos resultados a casos de más de tres ordenadas, tal como se indica en la figura siguiente:

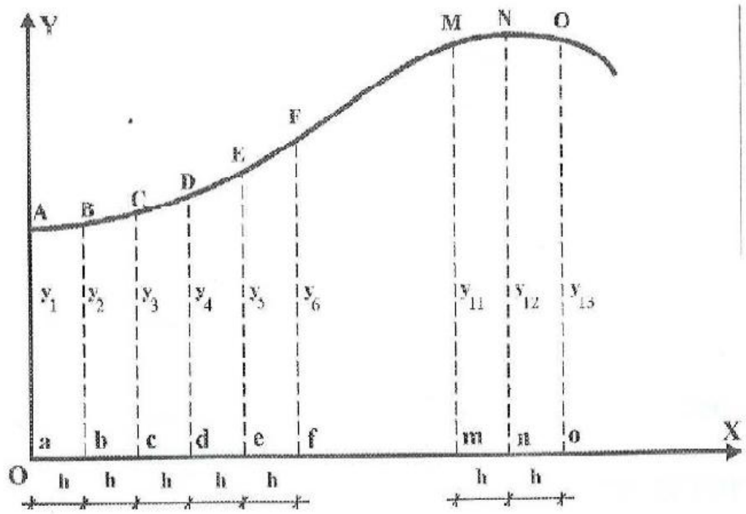
$$\int_{a}^{c} y * dx = \frac{h}{3}\ (y_1 + 4 * y_2 + y_3)$$

$$\int_{c}^{e} y * dx = \frac{h}{3}\ (y_3 + 4 * y_4 + y_5)$$

$$\int_{m}^{o} y * dx = \frac{h}{3}\ (y_{11} + 4 * y_{12} + y_{13})$$

La integral definida o área total será:

$$\int_{x1}^{x3} y * dx = \frac{h}{3}\ (y_1 + 4y_2 + 2y_3 + 4y_4 + 2y_5 + \cdots + 2y_{11} + 4y_{12} + y_{13})$$



La primera regla de Simpson solamente se puede emplear si disponemos de un número impar de ordenadas equiespaciada.

<u>Segunda Regla de Simpson</u>

Si se desea calcular la integral definida de la función: *y=f(x)* cuya expresión se desconoce entre los limites x_1 y x_2, tal y como se indica en la siguiente figura, se sustituye en ese tramo la función por una parábola de tercer grado de ecuación:

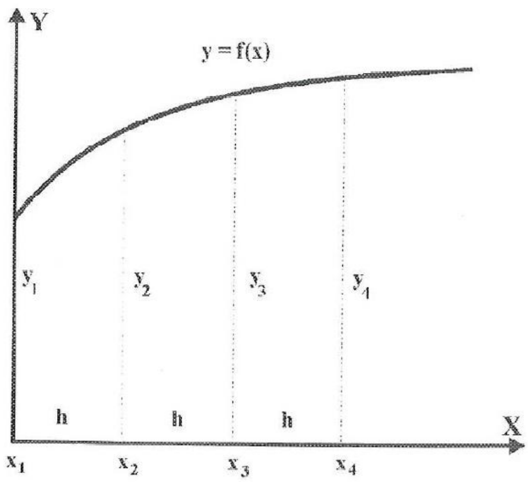
$$y = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3$$

$$\int_{x1}^{x4} y * dx = \int_0^{3h} (a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3) * dx$$

$$\int_0^{3h} (a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3) * dx = \left[a_0 * x + a_1 * \frac{x^2}{2} + a_2 * \frac{x^3}{3} + a_3 * \frac{x^4}{4}\right]_0^{3h}$$

$$= 3 * a_0 * h + a_1 * \frac{9 * h^2}{2} + a_2 * \frac{27 * h^3}{3} + a_3 * \frac{81 * x^4}{4}$$

Se conocen cuatro puntos de la función, por donde deberá de pasar también la parábola que la sustituye, por tanto la ecuación de la parábola deberá satisfacer para estos cuatro puntos lo que da lugar a cuatro ecuaciones, de donde se obtienen los valores de a_0, a_1, a_2 y a_3.



$(x=0;\ y=y_1)$	$y_1 = a_0$
$(x = h;\ y = y_2)$	$y_2 = a_0 + a_1*h + a_2*h^2 + a_3*h^3$
$(x = 2*h;\ y = y_3)$	$y3 = a_0 + 2*a_1*h + 4*a_2*h^2 + 8*a_3*h^3$
$(x = 3*h;\ y = y4)$	$y3 = a_0 + 3*a_1*h + 9*a_2*h^2 + 27*a_3*h^3$

Resolviendo el sistema y sustituyendo en la expresión de la integral obtenemos la siguiente expresión:

$$\int_{x1}^{x4} y * dx = \frac{3 * h}{8}\ (y_1 + 3y_2 + 3y_3 + y_4)$$

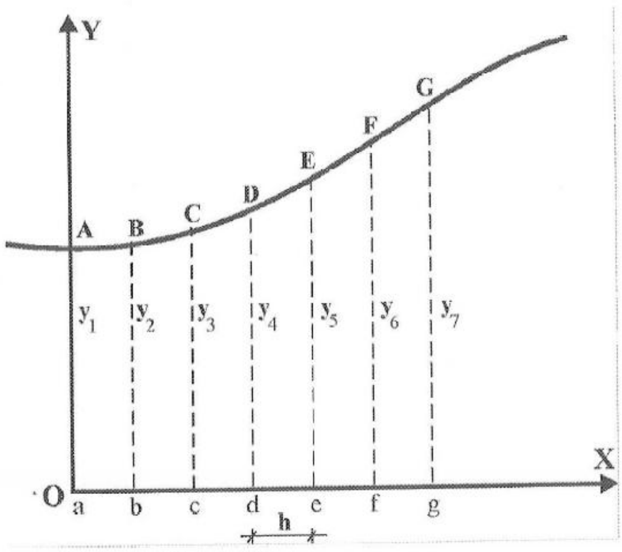
Como en el caso de la primera regla de Simpson, se puede hacer extensiva esta segunda regla a más de cuatro coordenadas, bastando para ello realizar la integración entre tramos sucesivos de cuatro ordenadas tal como se indica en la siguiente figura:

$$\int_a^d y * dx = \frac{3}{8} * h * (y_1 + 3 * y_2 + 3 * y_3 + y_4)$$

$$\int_d^g y * dx = \frac{3}{8} * h * (y_4 + 3 * y_5 + 3 * y_6 + y_7)$$

$$\int_a^g y * dx = \frac{3}{8} * h * (y_1 + 3 * y_2 + 3 * y_3 + 2y_4 + 3 * y_5 + 3 * y_6 + y_7)$$

La regla de Simpson es aplicable solamente cuando el numero de ordenadas equiespaciadas conocidas es de: 4, 7, 10, 13… (3*n + 1).

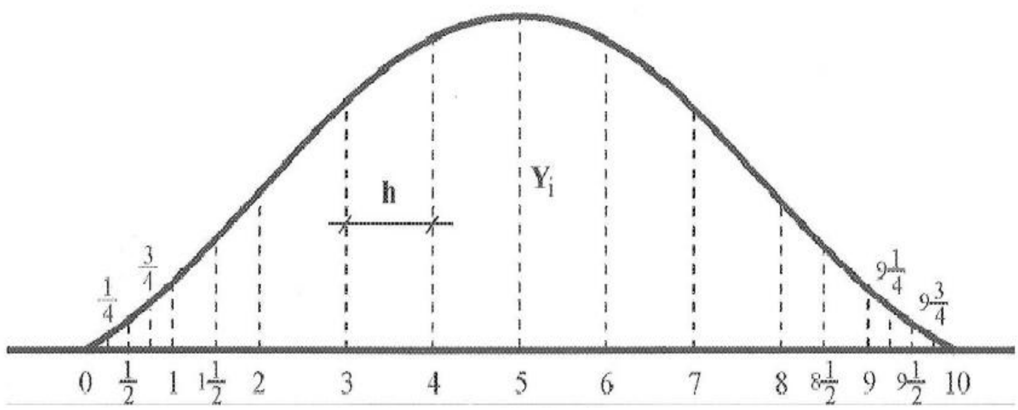


Empleo de ordenadas semi-espaciadas.

Hasta ahora, se han considerado siempre intervalos iguales de valor h. Puede suceder que, en zonas de extrema curvatura, interese disminuir el espaciado entre ordenadas, con el fin de que la curva parabólica se ajuste mejor a la curva real y obtener así una mayor exactitud. Esto sucede, concretamente, en los finos de proa y popa de las formas de un buque.

En estas zonas, y con el fin de definir más exactamente la carena, se recurre al empleo de ordenadas situadas a la mitad de la separación entre secciones transversales e incluso, espaciadas un cuarto de distancia.

Por ejemplo, en la curva representada en la figura siguiente, se han definido en los extremos ordenadas dispuestas a la mitad y a la Cuarta parte del espaciado.



Aplicando la primera regla de Simpson en tramos definidos por un número impar de coordenadas equiespaciadas, es decir entre las secciones 0-1, entre 1-2, entre 2-8, entre 8-9 y entre 9-10.

$$\int_0^1 y * dx = \frac{\frac{h}{4}}{3} * (y_0 + 4 * y_{\frac{1}{4}} + 2 * y_{\frac{1}{2}} + 4 * y_{\frac{3}{4}} + y_1)$$

$$\int_1^2 y * dx = \frac{\frac{h}{2}}{3} * (y_1 + 4 * y_{1-1/2} + y_2)$$

$$\int_2^8 y * dx = \frac{h}{3} * (y_2 + 4 * y_3 + 2 * y_4 + 4 * y_5 + 2 * y_6 + 4 * y_7 + y_8)$$

$$\int_8^9 y * dx = \frac{\frac{h}{2}}{3} * (y_8 + 4 * y_{8-1/2} + y_9)$$

$$\int_9^{10} y * dx = \frac{\frac{h}{4}}{3} * (y_9 + 4 * y_{9-\frac{1}{4}} + 2 * y_{9-\frac{1}{2}} + 4 * y_{9-\frac{3}{4}} + y_{10})$$

$$\int_0^{10} y * dx = \frac{h}{3} * (0{,}25y_0 + y_{\frac{1}{4}} + 0{,}5 * y_{\frac{1}{2}} + y_{\frac{3}{4}} + 0{,}75y_1 + 2 * y_{1-\frac{1}{2}} + 1{,}5y +_2 4 * y_3 + 2 \\ * y_4 + 4 * y_5 + 2 * y_6 + 4 * y_7 + 1{,}5y_8 + 2 * y_{8-\frac{1}{2}} + 0{,}75y_9 + y_{9-\frac{1}{4}} + 0{,}5 \\ * y_{9-\frac{1}{2}} + y_{9-\frac{3}{4}} + y_{10})$$

Por tanto, por cada espaciado intermedio en particular pueden calcularse los coeficientes correspondientes de la regla que se desea aplicar, en este caso la 2a Regla de Simpson, o combinar las diferentes reglas expuestas anteriormente.

5- Curvas hidrostáticas o propiedades de la carena recta.

Las propiedades geométricas de la carena dependen de las formas de la carena y la posición de la flotación correspondiente.

Si se considera una red de flotaciones paralelas entre si, por tanto con el mismo asiento, y con la carena adrizada, es decir con un ángulo de escora nulo, las diferentes propiedades geométricas de la carena dependerán solamente del calado medio.

Estas flotaciones reciben el nombre de isóclinas.

Reciben el nombre de Curvas Hidrostáticas o Carenas Rectas la representación tabular o grafica de estas propiedades geométricas correspondientes a las diferentes flotaciones, en función del calado medio.

La representación grafica permite, mediante interpolación grafica, calcular los valores para calados intermedios diferentes a los calculados.

Es importante tener en cuenta que estas curvas son validas solamente para flotaciones que tengan el mismo asiento que las flotaciones utilizadas en el cálculo y que normalmente coinciden con el asiento de proyecto. En caso contrario habrá que tener en cuenta esta diferencia y realizar las correcciones pertinentes.

El cálculo de las propiedades hidrostáticas se efectúa para un valor determinado del peso específico del agua. Para valores diferentes habrá que realizar las correcciones correspondientes.

Los cálculos de dichas curvas los seguiremos a partir del libro, recomendado por el tutor Leandro Peñalver, "Ship Hydrostatics and Stability" de A.B.Biran.

Usaremos la siguiente notación para dichos cálculos:

i número de sección; en el plano de formas

j número de sección, se define de tal manera que la distancia desde el origen de coordenadas (eje x) es *j dL*

X_i x-coordinada en la sección *i*

y_i semi-manga de cada sección *i* en cada línea de agua

α_i multiplicadores de integración para cada sección *i* según las reglas de Simpson

SL sub-intervalo de integración a lo largo del eje x

ST sub-intervalo de integración a lo largo del eje z

Para las definiciones anteriores tenemos, obviamente, j = 0 en el origen de coordenadas.

5.1- Propiedades de las áreas de flotación

5.1.1- Área de flotación

En esta sección, nos referimos a la figura 6.1 y supondremos que todas las líneas de agua son simétricas sobre la línea base. Este supuesto es cierto para casi todos los buques en la condición de la posición vertical.

Calculamos el área de flotación, de una línea de agua dada,

$$A_W = 2\int_a^b y\,\mathrm{d}x \approx 2\left(\sum_{i=n_1}^{n_2} \alpha_i y_i\right)\delta L$$

donde la línea de flotación comienza en la sección de n_1, con x - a, y termina en la sección n_2, con x = b.

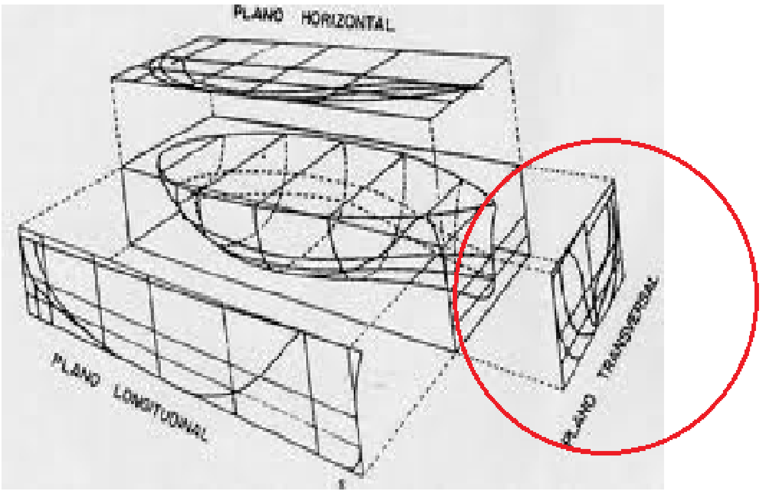
Para cada flotación o línea de agua, los valores de las semi-mangas correspondientes a las diferentes cuadernas de trazado se obtienen del plano de formas, por lo que la integral puede resolverse mediante alguno de los métodos de integración numérica explicados en el apartado 5 de integración numérica.

5.1.2- Área de la sección

Es el área delimitada por la sección vertical transversal del casco del barco. Se muestra imagen con dicha propiedad geométrica del casco del buque.

Calculamos el área como,

$$As = 2 * \int_a^b y * dz$$



5.1.3- Abscisa del centro de flotación

Para cada una de las flotaciones definidas anteriormente, se puede calcular la posición del centro de gravedad, centroíde o baricentro, correspondiente.

Por razón de simetría de la flotación respecto al plano de crujía, el centro de gravedad buscado estará en la línea de intersección del plano de la flotación con el diametral. Por tanto, solamente será necesario calcular la posición longitudinal o abscisa del centro de flotación.

$$x_F = \frac{M_x}{A_W} = \frac{2\left(\sum \alpha_i j_i y_i\right)\delta L^2}{2\left(\sum \alpha_i y_i\right)\delta L} = \frac{\left(\sum \alpha_i j_i y_i\right)}{\left(\sum \alpha_i y_i\right)}\delta L$$

El momento del área de flotación alrededor de un eje transversal que pasa por el origen de coordenadas es,

$$M_x = 2\int_a^b xy\,\mathrm{d}x \approx 2\left(\sum_{i=n_1}^{n_2} \alpha_i x_i y_i\right) 6L = 2\left(\sum_{i=n_1}^{n_2} \alpha_i j_i y_i\right) \delta L^2$$

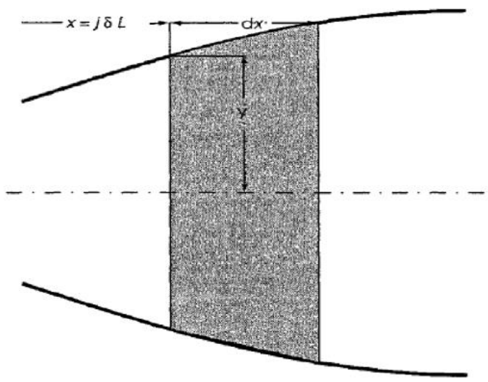
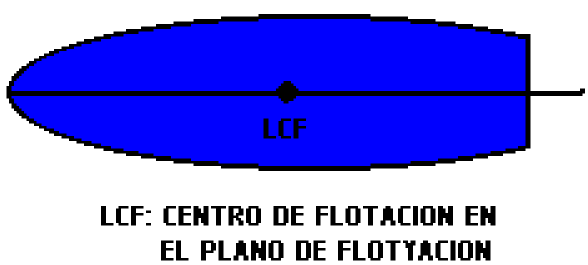


Fig. 6.1

Como eje longitudinal se considera la intersección de la flotación con el plano diametral y como eje transversal uno perpendicular al anterior en la intersección de la perpendicular de popa o de la sección media.

Para cada flotación o línea de agua, los valores de las semi-mangas correspondientes a las diferentes cuadernas de trazado, así como la posición de las cuadernas de trazado se obtienen del plano de formas, por lo que la integral puede resolverse a través de alguno de los métodos numéricos explicados anteriormente.

La notación X_F corresponde a la norma DIN 81209. La notación utilizada en los textos de idioma Inglés es LCF, un acrónimo de centro longitudinal de flotación.



5.1.4- Momento de inercia transversal, momento de inercia del plano de flotación y longitudinal

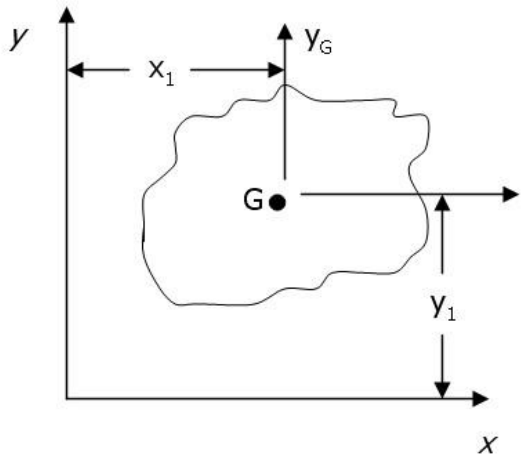
Para cada calado medio el momento de inercia transversal de la flotación respecto al eje baricéntrico longitudinal se define con la expresión:

$$I_{\mathrm{T}} = \int_a^b \frac{2}{3} y^3 \, \mathrm{d}x \approx \frac{2}{3} \left(\sum_{i=0}^{n} \alpha_i y^3 \right) \delta L$$

Eje baricéntrico longitudinal, es el eje longitudinal que pasa por el centro de la flotación. Coincide con el eje de simetría de la flotación.

El momento de inercia del área de flotación respecto al eje transversal que pasa por el origen de coordenadas es calculado como:

$$Iy = 2 * \int_a^b x^2 * y * dx$$



Para cada calado medio el momento de inercia longitudinal de la flotación respecto al eje baricéntrico transversal se calcula empleando el teorema de Steiner, mediante la expresión,

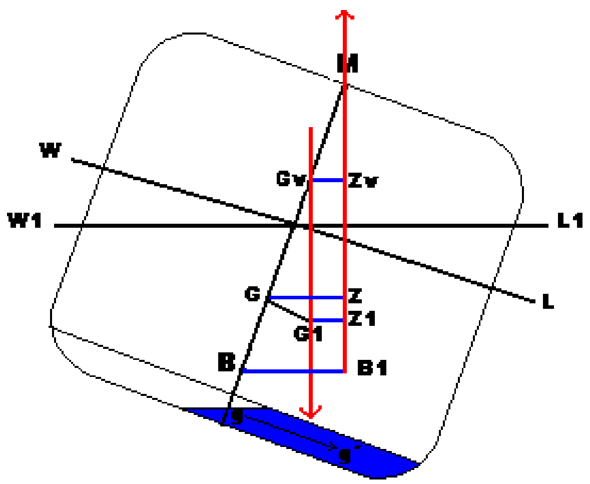
$$I_L = I_y + A_F * {X_F}^2$$

5.1.5- Radio metacéntrico transversal y longitudinal

Para cada flotación o línea de agua, los valores de las semi-mangas correspondientes a las diferentes cuadernas de trazado se obtienen del plano de formas, por lo que la integral puede resolverse a través de alguno de los métodos numéricos explicados anteriormente.

Supóngase un buque con volumen de carena igual a V, y su centro de carena en el punto. B. Si luego lo escoramos un ángulo θ sin alterar el desplazamiento, entonces el centro de carena adoptará una nueva posición B1, tal como se muestra en la figura. La recta de acción del empuje que antes pasaba por B ahora pasará por B1, Prolongando esa recta hasta cortar el plano de la crujía, o dicho de otro modo a la recta de acción primitiva para cuando el buque estaba adrizado, tendremos en la intersección de ambas rectas, el punto M. La coordenada vertical de este punto variará con el ángulo de escora, pero para inclinaciones no mayores a 10^{o} se pueden asumir como invariables y recibe el nombre de metacentro transversal inicial, ó abreviadamente metacentro transversal.

Dado que por definición el metacentro se encuentra en la vertical del centro de carena del buque adrizado, bastará con conocer la distancia vertical BM para fijar su posición.



$$BM_T = \frac{I_T}{V}$$

V = Volumen de carena correspondiente a la flotación considerada

Para cada flotación o línea de agua, los valores de las semi-mangas correspondientes a las diferentes cuadernas de trazado y las abscisas de las cuadernas de trazado se obtienen del plano de formas, por lo que la integral puede resolverse por cualquiera de los métodos de integración numérica explicados anteriormente.

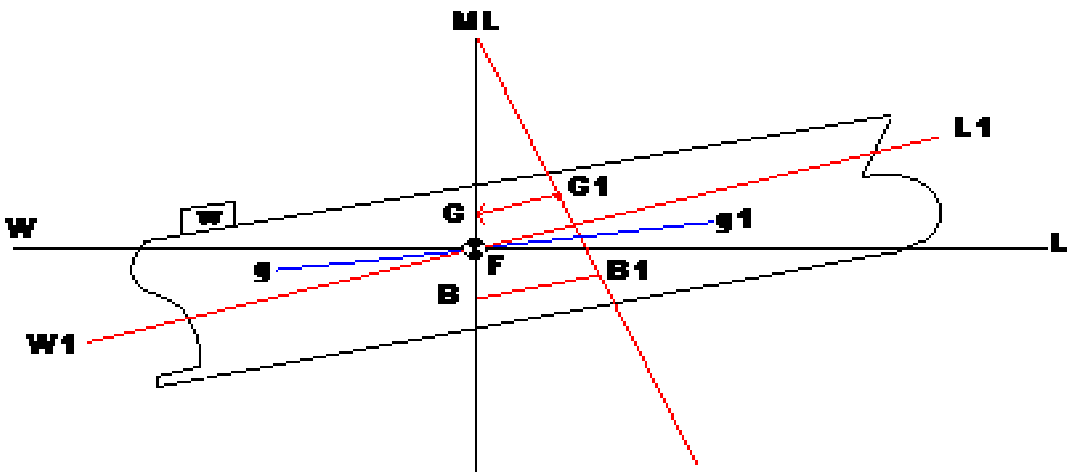
Para una inclinación longitudinal infmitesimal, los empujes que pasan por la posición inicial y final del centro de carena intersectarán en un punto denominado metacentro longitudinal. Partiendo de la situación de equilibrio para el buque sin asiento, el empuje correspondiente a un ángulo infinitesimal, cortará a la línea de empuje del centro de carena inicial en un punto, ML, metacentro longitudinal inicial.

Dentro de los primeros grados de inclinación longitudinal, las diferentes líneas de empuje pasarán, prácticamente, por el punto ML.

El radio metacéntrico longitudinal, BM_L, se deducirá a partir del movimiento del centro de carena debido a una inclinación longitudinal isocarena, cuyo valor hallado es:

$$BM_L = \frac{I_L}{V}$$

V = Volumen de carena correspondiente a la flotación considerada



5.2- Propiedades del volumen

5.2.1- Volumen de carena. Posición trasversal del centro de carena.

Carena se denomina al volumen limitado por el casco y por la superficie de flotación en un buque. Volumen de carena es el volumen de la parte sumergida del barco. El centro de carena (z_B) es el centro de gravedad del volumen de la carena para la flotación considerada

Podemos obtener el volumen de carena correspondiente a un calado determinado, mediante la integración "vertical" de las áreas de flotación desde el punto más bajo del casco.

$$\nabla = \int_0^{T_0} A_W \, dz \approx \left(\sum_{i=1}^{i_T} \alpha_i A_{Wi} \right) \delta T$$

El momento del volumen de desplazamiento por encima de la línea de base también se puede conseguir por la integración "vertical".

$$\mathrm{MB} = \int_0^{T} \circ \, T A_W dz * \left(\sum_{i=1}^{i_T} \alpha_i z_i A_{Wi} \right) \delta T = \left(\sum_{i=1}^{i_T} \alpha_i j_i A_{Wi} \right) \delta^2 T \quad (4.9)$$

Donde z_i es la coordenada z de la línea de flotación y j_i el número de la línea de flotación contando desde la línea base.

Para cada calado, la carena derecha correspondiente tendrá un determinado centro de carena. Por razones obvias de simetría, sobre el plano de crujía, bastará con dos coordenadas para definir su posición perfectamente: su distancia a una sección transversal determinada, generalmente la sección maestra o la perpendicular de popa, y su altura sobre el plano de construcción.

Calculamos la coordenada vertical del centro de carena como,

$$ZB = \frac{MB}{\nabla}$$

La notación z_b se establece en la norma DIN 1209 8. Las notaciones comunes en los libros en inglés son KB o VCB, siendo este último el acrónimo de centro vertical de carena.

Calculamos el momento del volumen de desplazamiento respecto de la sección maestra o de la sección de perpendicular de popa,

$$MBx = \int_0^x x * As * dx$$

La posición longitudinal del centro de carena será,

$$XB = \frac{MBx}{\nabla}$$

5.3- Datos derivados

5.3.1- Toneladas por centímetro de inmersión

Se definen las toneladas por centímetro de inmersión (TPC) como el incremento de desplazamiento correspondiente a una flotación determinada de las Curvas Hidrostáticas, cuando el calado aumenta paralelamente un centímetro.

Considerando que el área de la flotación permanece prácticamente constante cuando el calado aumenta paralelamente un centímetro, se verifica que el incremento de volumen de carena viene dado por la expresión:

$$v = A_W * 0.01$$

El aumento de empuje o desplazamiento buscado será:

$$TPC = A_W * 0.01 * \rho$$

$\rho =$ Peso especifico del agua, utilizado en las curvas hidrostáticas

6-Introducción a MATLAB

6.1- Introducción a MATLAB

MATLAB (MATrix LABoratory) es un programa orientado al cálculo con matrices, al que se reducen muchos de los algoritmos que resuelven problemas de Matemática Aplicada e Ingeniería.

MATLAB ofrece un entorno interactivo sencillo mediante una ventana en la que podemos introducir órdenes en modo texto y en la que aparecen los resultados. Los gráficos se muestran en ventanas independientes. Cada ventana dispone de una barra de menús que controla su funcionalidad.

Aprenderemos a asignar, borrar, guardar y recuperar variables, utilizar las funciones incorporadas y, más adelante, a definir funciones nuevas. MATLAB opera directamente con números complejos y con números reales como caso particular.

Lo que distingue a MATLAB de otros sistemas de cálculo es su facilidad para trabajar con vectores y matrices. Las operaciones ordinarias, suma, producto, potencia, operan por defecto sobre matrices, sin más restricción que la compatibilidad de tamaños en cada caso.

Una de las características más destacables de MATLAB es su capacidad gráfica. Explicaremos algunos comandos gráficos para representación de funciones de una o dos variables en distintos sistemas de coordenadas.

6.2- Introducción de datos. Uso de la ventana de comandos

El elemento básico en MATLAB es la matriz compleja de doble precisión, estructura que abarca realmente todo tipo de datos, desde escalares tales como números reales o complejos, hasta vectores o matrices de tamaños arbitrarios. Implícitamente se usa la notación matricial para introducir polinomios y funciones de transferencia, de la forma que se explicará más adelante. Por otro lado, si se dispone de una representación de un sistema lineal en el espacio de estados de la forma:

x = A x + B u
y = C x + D u

bastaría con introducir los valores de los elementos de las matrices A, B, C y D, para tener descrito al sistema. Estos elementos se podrían introducir de la siguiente forma:

A = [1 0 2;2 2 0;0 0 1]

B = [1, 0,0]'

C = [1 1 sqrt(2)]

D = 0;

A la vista de esta serie de comandos se pueden comentar varias cosas:

- Si al final de la introducción de un comando cualquiera no se pone punto y coma (;), aparecerá explícitamente en pantalla el resultado de dicho comando. En caso contrario, el comando se ejecutará, pero no se mostrará su resultado. Dicho resultado se habrá almacenado en la variable a la que se asigna o, si no se realiza asignación, se guardará en una variable de entorno llamada ans. En caso de que se asigne a una variable, ésta se creará automáticamente, sin necesidad de una declaración previa.
- Los elementos de cada fila de una matriz se pueden introducir separados por espacios o por comas, indistintamente.
- Para separar filas de una matriz se usa ";" o un simple retorno de carro. Esta última opción puede facilitar muchas veces la visualización de la matriz que se está introduciendo.

- Para transponer matrices se usa el apostrofe.
- Los elementos de vectores y matrices pueden ser reales, complejos e incluso expresiones, como vemos en el caso del último elemento del vector C.
- Si se está introduciendo un comando o conjunto de ellos cuya sintaxis sea muy larga, se puede continuar en la siguiente línea introduciendo al final de la actual tres puntos seguidos (...).
- Las variables a las que se asignan resultados, así como las variables de entorno, se almacenan en lo que se denomina el espacio de trabajo de MATLAB (workspace).

En este caso, se han creado una serie de variables (en particular, matrices) mediante la introducción explícita de sus elementos en línea de comandos. Otras formas de producir variables podrían ser: generándolas mediante funciones y declaraciones, creándolas en un archivo .m, cargándolas desde un archivo de datos externo mediante el comando load (bien se trate de ficheros de datos ASCII o bien de ficheros binarios con formato de datos de MATLAB .mat).

6.3- Variables de entorno y variables especiales

Existen una serie de variables predefinidas en MATLAB, son las siguientes:

- *ans*: Contiene la respuesta (answer) del ´ultimo comando ejecutado, cuando el resultado de dicho comando no se asigna explícitamente a ninguna variable.
- *eps*: Da el valor de la precisión con la que la máquina realiza las operaciones en punto flotante. Típicamente, esta precisión es del orden de 10−17.
- *pi*: π.
- *i, j*: √−1. Constante imaginaria.
- *inf*: ∞. Se trata de un valor excesivamente grande para ser almacenado.
- *NaN*: Not a number. Es el resultado que se proporciona si durante una operación se produce una indeterminación, del tipo $0 \cdot \infty, \frac{\infty}{\infty}, \frac{0}{0}$...
- *clock*: Reloj.
- *date*: Fecha.
- *flops*: Número de operaciones en punto flotante realizadas hasta el momento.

El comando *who* muestra las variables existentes en el espacio de trabajo generadas por el usuario, pero no las variables especiales.

Para borrar alguna variable de memoria se utiliza *clear* nombre-variables separadas por espacios. Pueden borrarse todas las variables a la vez si no se especifica ningún nombre a continuación del nombre del comando.

6.4- Elementos de las matrices

En este punto es importante comentar uno de los elementos más potentes de MATLAB, que es el símbolo ":", que permite generar una secuencia, y en particular permitirá referenciar varios elementos de una matriz. Veamos algunos ejemplos en los que se usa este operador:

- 1:0.1:10 Generará una secuencia comenzando por 1 hasta 10, cada elemento de la secuencia estará separado del anterior en 0.1.

- 1:10 Si se obvia el valor central, la separación entre cada dos elementos de la secuencia será 1.

- [1:0.1:10] Si lo ponemos entre corchetes, estaremos generando un vector con los elementos de la secuencia.

En la forma más directa, los elementos de una matriz se referencian mediante A(i, j), donde i y j son los índices del elemento correspondiente. Podemos usar una secuencia que facilitar la indexación de múltiples elementos, como en los siguientes ejemplos:

- A(1,2:3) dará como resultado los elementos de las columnas 2 y 3 pertenecientes a la primera fila.

- A(:,2) dará como resultado todos los elementos pertenecientes a la segunda columna.

Lógicamente, en estos casos, los elementos especificados como inicio, final e incremento para producir la secuencia deben ser enteros.

Otra forma de generar datos secuencialmente es usando los comandos "linspace" y "logspace", su formato es:

t = linspace(n1,n2,n);

w = logspace(n1,n2,n);

El comando linspace genera un vector desde n1 a n2 de longitud n, cuyos componentes poseen valores espaciados linealmente. Por su parte, logspace produce también un vector de n elementos, pero sus valores están espaciados logarítmicamente desde 10^{n1} a 10^{n2}. Este ultimo comando resultará útil para la generación de escalas frecuenciales para el análisis de sistemas mediante diagramas de Bode, Nyquist, etc

6.5- Operaciones con matrices

Las operaciones comunes con matrices son:

- Suma: +
- Resta: -
- Multiplicación: *
- División derecha / (x = b/A es la solución del sistema de ecuaciones x ∗ A = b. Es decir calcula la inversa de la matriz A y multiplica b por la derecha por dicha inversa)
- División izquierda \ (x = A \ b es la solución de A ∗ x = b. Es decir, igual que en el caso anterior, pero realiza la multiplicación de la inversa con b por la izquierda)
- Potenciación ^. Este operador permite, en particular, implementar otra forma de realizar la inversión de una matriz: A^(-1).
- Conjugada traspuesta ’

Cabe mencionar la potencia de los operadores /, \, y ^, puesto que si la matriz A no es cuadrada, automáticamente se realiza el cálculo de su pseudo-inversa, lo que equivaldría a resolver el sistema de ecuaciones correspondiente por mínimos cuadrados.

Las mismas operaciones que se han enumerado se pueden realizar elemento a elemento, anteponiendo un punto a cualquiera de los operándoos anteriores. Como ejemplo, el siguiente comando realizaría el producto de cada elemento de la matriz A con su correspondiente de la matriz B (para que dicho producto sea realizable, obviamente, dichas matrices deben tener las mismas dimensiones):

A .* B

Además de los operadores anteriores, existen funciones tales como:

- Trigonométricas estándar: *sin, cos, tan, asin, acos, atan, atan2*
- Trigonométricas hiperbólicas: *sinh, cosh, tanh, asinh, acosh, atanh*
- Trascendentales: *log, log10, exp, sqrt*
- Manipulación de números complejos:

    • real: parte real de un escalar o de los elementos de una matriz.

    • imag: parte imaginaria.

    • conj: proporciona el conjugado de un escalar o la matriz conjugada a una dada.

- Cálculo del módulo: abs permite calcular tanto el valor absoluto de un escalar real como el módulo de un escalar complejo o el módulo de un vector.
- Funciones típicas de matrices:

    • det: determinante de una matriz
    • inv, pinv: inversa y pseudoinversa
    • eig: obtención de auto-valores
    • rank: rango de la matriz
    • norm: norma de una matriz (norma 2, norma 1, norma infinito, norma de Frobenius)
    • trace: traza de la matriz
    • diag: produce un vector conteniendo los elementos de la diagonal de una matriz, o si recibe un vector como parámetro, genera una matriz diagonal.
    • tril: devuelve la matriz triangular inferior de una matriz dada
    • triu: devuelve la matriz triangular superior de una matriz dada

- Funciones para generar matrices:

    • eye(n): produce una matriz identidad de dimensiona n × n

    • zeros(n,m): genera una matriz de ceros de dimensión n × m

    • ones(n,m): genera una matriz de unos de dimensión n × m

• rand(n,m): permite generar una matriz de valores aleatorios, entre 0 y 1, de dimensión n × m

• A = [A11,A12;A21,A22]: podemos producir una nueva matriz por bloques, mediante su composición a partir de sub-matrices ya existentes.

6.6- Funciones orientadas al análisis de datos

Se trata de funciones que operan con vectores. Si se aplican a matrices operan columna a columna. Permiten realizar análisis sobre el conjunto de datos contenido en los vectores correspondientes, tales como calcular su valor mínimo, máximo, media, mediana, desviación típica, suma de los elementos de dicho vector, etc. *min, max, mean, median, std, sum, prod, etc.*

6.7- Polinomios

Esta es una sección importante, dado que las funciones de transferencia de los sistemas se introducirán habitualmente en la forma *numerador-denominador*, los cuales serán tratados como polinomios por MATLAB. En las demos que acompañan a estas notas se podrán analizar numerosos ejemplos.

Los polinomios se representan por vectores, cuyos elementos son los coeficientes del polinomio en orden descendente. Por ejemplo, el polinomio s^3+2s^2+3s+4 se representa:

p=[1 2 3 4];

que muy bien podría ser el denominador de una función de transferencia.

Mediante la función *roots* se pueden encontrar las raíces de esa ecuación: roots(p)

De modo complementario, se puede calcular un polinomio a partir de sus raíces usando la función *poly*:

p2=poly([-1 -2]);

Si el argumento de entrada a *poly* es una matriz, devuelve el polinomio característico de la matriz *(det(λI −A)*) como un vector fila.

Un polinomio puede ser evaluado en un punto determinado usando *polyval(p,s),* donde p es el polinomio y s es el punto donde va a ser evaluado. Por ejemplo:

p2=[1 3 2]; a=[1 2; 3 4]; polyval(p2,a)

si se introduce, como en este caso, un vector o una matriz, en lugar de un valor individual, la evaluación se hace elemento a elemento.

Podemos realizar cómodamente operaciones de multiplicación y división de polinomios mediante las funciones *conv y deconv*, respectivamente:

conv([1,2],[2,0])

6.8- Gráficos

MATLAB es muy potente a la hora de generar gráficos (sobre todo en sus últimas versiones), no sólo por la variedad de comandos que ofrece para ello, sino también por la versatilidad de dichos comandos. En las demostraciones aparecerán varios tipos de gráficos. De momento, comentaremos los comandos fundamentales para la realización de los mismos. En primer lugar, comandos genéricos y comandos orientados a gráficos bidimensionales:

- *figure(n):* Las representaciones de gráficos en MATLAB se realizan en ventanas gráficas. En un momento dado puede haber varias ventanas gráficas abiertas. La función figure se utiliza para abrir una nueva ventana gráfica que será numerada de acuerdo con el parámetro, o bien, si ya existe una ventana con ese número, se convertirá en la ventana gráfica activa, donde se realizará la próxima representación gráfica.

- *clf*: Limpia la ventana gráfica activa.

- *close(n)*: Para cerrar una ventana gráfica. *close all* cierra todas las ventanas gráficas.

- *plot*: es la función básica de representación gráfica de datos en dos dimensiones. La representación se realiza en la ventana gráfica que esté activa en un momento dado. En caso de no haber ninguna, se crea una ventana gráfica nueva. Ejemplos de uso:

  • *plot(v)*: representa en el eje vertical los valores contenidos en el vector v, frente a los valores del índice en el eje horizontal.

  • *plot(t,v)*: representa los valores del vector v frente a los del vector t.

  • *plot(t,A), plot(t,[v1,v2])*: presentará varias gráficas, puesto que cada columna de la matriz A es considerada como un vector a representar frente al vector t. En la segunda variante indicada, se consigue lo mismo mediante la agrupación de los vectores v1, v2 en una matriz.

  • *plot(t1,v1,t2,v2):* En este caso también se obtendrán dos gráficas, pero cada una de ellas tiene un conjunto de valores diferente para el eje horizontal.

- *loglog*: representación en escala logarítmica en ambos ejes.

- *semilogx*: representación en escala semi-logarítmica, el eje vertical aparecerá en escala lineal.

- *semilogy*: representación en escala semi-logarítmica, el eje horizontal aparecerá en escala lineal.

- *polar*: representación de datos dados en forma polar, es decir en lugar de dar un par de vectores de componentes horizontales y verticales, se dan los vectores conteniendo el vector de ángulo y módulo.

Cuando se representan varias curvas simultáneamente en una misma ventana gráfica, se utiliza una secuencia predefinida de colores para aplicar uno diferente a cada una de ellas. Se puede cambiar manualmente el color que por defecto tendrá una determinada curva con la adición de un parámetro: *plot(t,y,’r’).* En este ejemplo, en lugar de representarse la curva con el color por defecto (azul), aparecerá en color rojo. Para ver los códigos de colores, puede consultarse la ayuda del comando *plot*.

También pueden realizarse gráficos en tres dimensiones:

- *plot3(x,y,z):* comando análogo a *plot* para dibujar curvas, pero en tres dimensiones.
- *mesh(x,y,Z):* para dibujar superficies, Z debe ser una matriz con tantas filas como longitud del vector x y tantas columnas como la longitud del vector y. Los puntos que se representan son: (x(i), y(j), Z(i, j)).

- *contour*: representa en un plano horizontal las curvas de nivel de una superficie tridimensional.

Por otro lado, existen comandos que permiten añadir determinados complementos a estos gráficos:

- –*title*: permite añadir un título a la gráfica
- –*xlabel*: añadir una etiqueta al eje horizontal de la gráfica
- –*ylabel*: añadir etiqueta al eje vertical
- –*grid*: añadir una rejilla
- –*axis*: permite modificar los límites de los ejes horizontal y vertical
- –*text*: añadir un texto en una posición cualquiera de la gráfica
- –*gtext*: igual que *text* pero permite seleccionar la ubicación del texto mediante el ratón.

Por otra parte, muchos de los elementos gráficos pueden manipularse como objetos que tienen una serie de propiedades asociadas. Por ejemplo:

handlePlot = plot(x,y);

con este comando estamos asignando el objeto de tipo *plot* a una variable. Podemos ver las propiedades asociadas a un objeto mediante la función *get(handlePlot*), o bien especificar alguna de ellas: *get(handlePlot,'LineStyle'*). Cualquiera de las propiedades de un objeto pueden ser alteradas mediante la función *set(handlePlot,'Color','g'*).

Por otra parte, también se dispone de cierta capacidad de modificación de las gráficas mediante opciones de la propia ventana gráfica, en lugar de usar instrucciones desde la ventana de comandos.

7- Programando curvas hidrostáticas con MATLAB

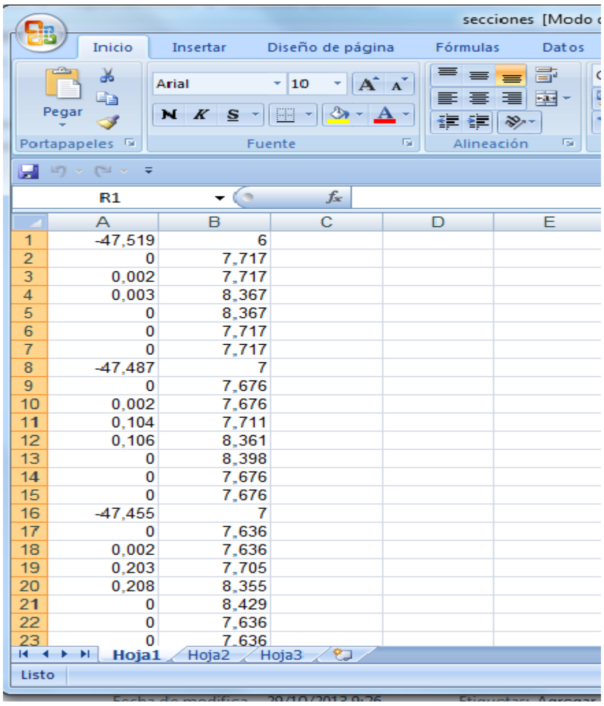
7.1- Introducción al programa

Con toda la información recogida en los apartados anteriores, comenzaremos a crear nuestro programa. El objetivo fundamental será realizar los cálculos de las principales propiedades de la carena adrizada de una manera totalmente automática. Para ello haremos lo siguiente:

- Creamos un programa que transforme nuestro archivo .gf en un archivo .xls.
- Obtenemos las secciones, es decir unimos los puntos de cada sección para obtenerlas.
- Dibujamos las secciones.
- Creamos un programa que realice todas las líneas de agua.
- Dibujamos las líneas de agua.
- Hacemos todos los cálculos de las propiedades geométricas de la carena.
- Interfaz gráfica.

7.2- Transformación del archivo GHS (*gf) a un archivo Excel (*xls)

Una vez obtenido el archivo *gf de Rhinoceros, el cual contiene toda la información, en cuanto a formas, de nuestro barco, lo que haremos será transformarlo en un archivo *xls, es decir, en un archivo de Excel, que contendrá cada uno de nuestros puntos en diferentes filas, y cada coordenada en diferentes columnas, en la siguiente imagen podemos ver cómo quedaría.

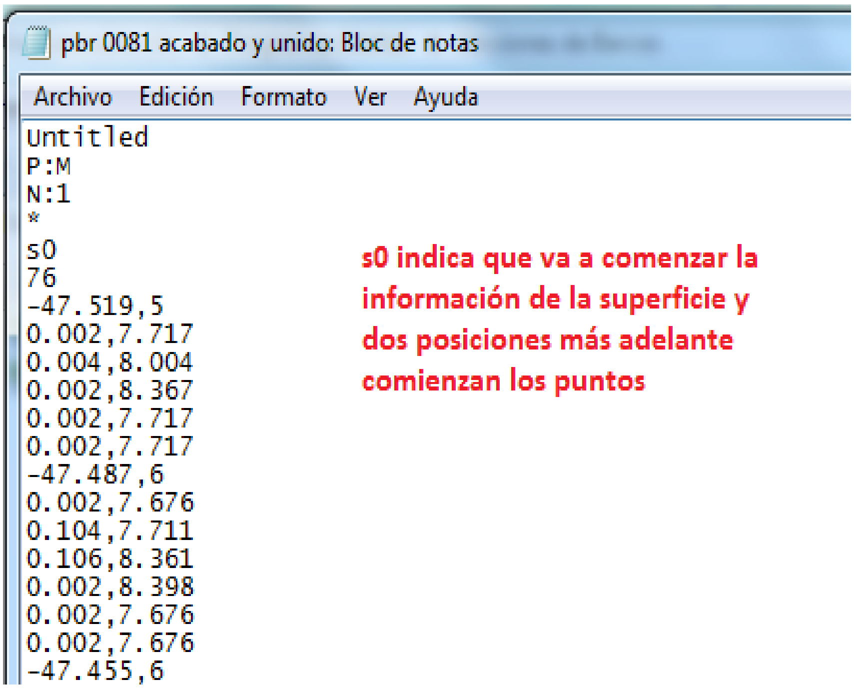


	A	B	C	D	E
1	-47,519	6			
2	0	7,717			
3	0,002	7,717			
4	0,003	8,367			
5	0	8,367			
6	0	7,717			
7	0	7,717			
8	-47,487	7			
9	0	7,676			
10	0,002	7,676			
11	0,104	7,711			
12	0,106	8,361			
13	0	8,398			
14	0	7,676			
15	0	7,676			
16	-47,455	7			
17	0	7,636			
18	0,002	7,636			
19	0,203	7,705			
20	0,208	8,355			
21	0	8,429			
22	0	7,636			
23	0	7,636			

Los valores numéricos, así como la estructura, quedaron explicados en el apartado "3.2.4 – Obtención de datos de las formas del buque".

Para transformar dicho en archivo, crearemos un programa con MATLAB que hará lo siguiente:

El archivo de entrada o archivo .gf, contiene una información mayor a la que nosotros necesitamos, es decir contiene líneas de caracteres que nosotros no queremos, por lo que haremos que el programa solo coja lo fundamental para trabajar en nuestro proyecto, es decir, la posición de cada sección y los puntos de cada sección. Para ello le diremos a MATLAB que busque, donde comienza dicha información en nuestro archivo y donde finaliza. Vemos en las dos siguientes imágenes este hecho.



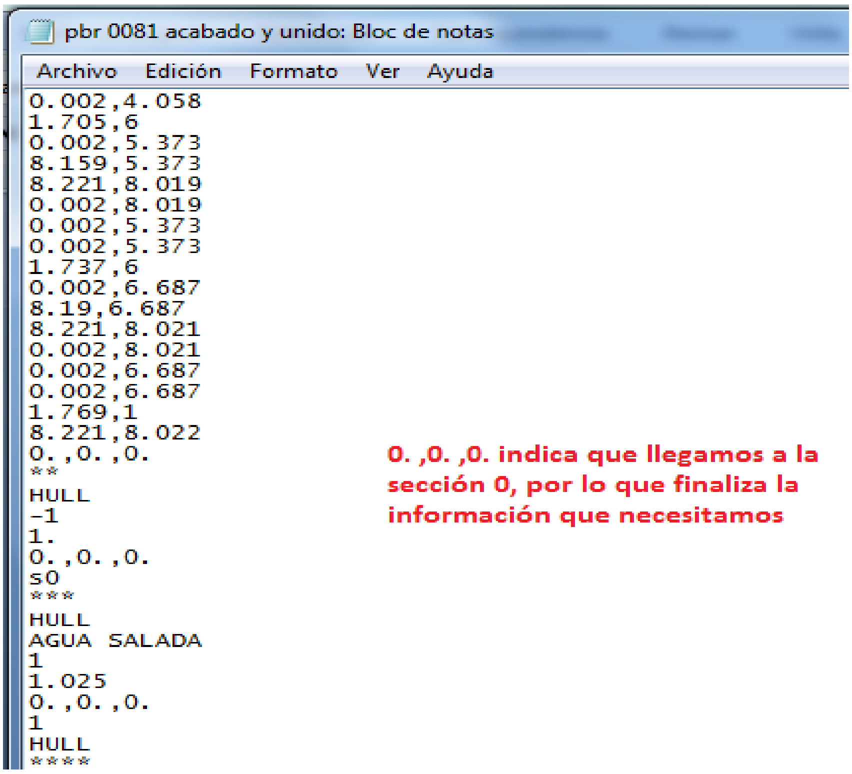
pbr 0081 acabado y unido: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
Untitled
P:M
N:1
*
s0
76
-47.519,5
0.002,7.717
0.004,8.004
0.002,8.367
0.002,7.717
0.002,7.717
-47.487,6
0.002,7.676
0.104,7.711
0.106,8.361
0.002,8.398
0.002,7.676
0.002,7.676
-47.455,6
```

s0 indica que va a comenzar la información de la superficie y dos posiciones más adelante comienzan los puntos

Diremos a MATLAB que localice "s0" y que sume dos posiciones a dicha posición, después haremos que lea y guarde toda la información a partir de ahí, hasta llegar al final. Localizamos el final, todas las estructuras de este archivo deben finalizar la información de los datos de las formas, con una estructura tipo 0., 0. , 0. , por lo que diremos a MATLAB que finalice de leer información cuando llegue justamente ahí.




```
0.002,4.058
1.705,6
0.002,5.373
8.159,5.373
8.221,8.019
0.002,8.019
0.002,5.373
0.002,5.373
1.737,6
0.002,6.687
8.19,6.687
8.221,8.021
0.002,8.021
0.002,6.687
0.002,6.687
1.769,1
8.221,8.022
0. ,0. ,0.
**
HULL
-1
1.
0. ,0. ,0.
s0
***
HULL
AGUA SALADA
1
1.025
0. ,0. ,0.
1
HULL
****
```

Separaremos el número a la derecha y a la izquierda de la coma, para que los separe en diferentes columnas de Excel. Por último guardaremos como un archivo .xls el fichero de salida.

A continuación expongo el programa realizado con MATLAB con una breve explicación. El programa se llamará *“transformar_fichero”.*

```
function transformar_fichero(fichero_in,fichero_out)

% Esta función transforma un fichero con formato .gf que contiene
% las secciones de un barco en un fichero con excel con formato
% .xls
% El fichero .gf es generado con el programa Rhinoceros
% fichero_out=transformar_fichero(fichero_in)
% Variables de entrada:
% fichero_in  archivo con extensión .gf extraído de Rhinoceros
%       que contiene la información de las secciones del barco
% fichero_out contiene el nombre del archivo excel con extensión
%      .xls que contendrá también la información de las secciones
```

```
% componemos para formar la ruta
fichero_in=['./Secciones de Barcos/',fichero_in];



% leemos todos los datos del archivo en una variable de celda
data=textread(fichero_in,'%s');

% localizamos la posición donde empiezan las secciones

k=1;

while ~strcmp(data{k}(1),'s')
    k=k+1;
end

% le sumamos 2 a la posición de s0, s1, ...
k=k+2;

p=k; % variable que controla el bucle

while ~strcmp(data{p},'0.,0.,0.')

    % Separamos el número a la izquierda y a la derecha de la coma
    aux=data{p};   % aux es char y no cell

    c=1;
    while aux(c)~= ','
          c=c+1;
    end

    A(p-k+1,1)=str2num(aux(1:c-1));
    A(p-k+1,2)=str2num(aux(c+1:end));

    p=p+1;
end


% guardamos en un fichero excel .xls la matriz A

success = xlswrite(['./Secciones de Barcos/',fichero_out],A);
```

7.3- Obtención de las secciones

Con nuestro archivo *xls, MATLAB podrá leer de una forma más ordenada y precisa todos nuestros puntos de las secciones. Esta función nos leerá el archivo *xls y nos dará en una variable “s” toda la información con la siguiente estructura:

s.x: contendrá la posición x de la sección.

s.numero: número de puntos de la sección considerada.

s.puntos: coordenada (y,z) de los puntos de la sección.

s.tipo: nos dirá que tipo de línea tenemos, es decir, MATLAB unirá los puntos y así crear una polilínea, esta variable nos dirá el tipo de cada línea. Podrá ser una línea paralela al Eje Y (si las dos coordenadas z de los dos puntos que forman la línea son iguales), paralela al Eje Z (si las dos coordenadas y de los dos puntos que forman la línea son iguales) y oblicua (las coordenadas de los dos puntos son diferentes).

s.poligonal: contiene la información de cada línea para poder crear la sección.

s.limites: nos indica de donde hasta donde se extiende cada línea.

El objetivo de crear está función es que nuestro programa dibuje cada una de nuestras secciones de manera automática, a través de una interfaz gráfica.

Esta función se llamará “obtener_secciones”, a continuación expongo cómo se ha realizado en MATLAB.

```
function s=obtener_secciones(fichero)

% Esta función lee las secciones de un barco en una variable
% de tipo estructura a partir del nombre de un fichero .xls
% s=obtener_secciones(fichero);
% Variables de entrada:
% fichero nombre del archivo que contiene el fichero .xls
% Variables de salida:
% s estructura que contiene las diferentes secciones del barco
%   Sus campos son los siguientes:
%   s.x  posición espacial x de la sección
%   s.numero  número de puntos de control en la sección considerada
%   s.puntos coordenadas (y,z) de los puntos de control
%   s.tipo tipo de poligonal entre un punto y el siguiente. Puede
%          ser paralela al eje y, paralela al eje z, u oblicua
%   s.poligonal información necesaria para construir la poligonal
%   s.limites indica de dónde a dónde se extiende cada línea de la
%             poligonal
```

```
fichero=['./Secciones de Barcos/',fichero];
datos=xlsread(fichero);

n=size(datos,1);
ns=0;  % número de secciones

conversion=0.3048;  % convierte las medidas del Rhinoceros a metros

% Leemos las secciones

s=struct;
m=1;



while m<n

    ns=ns+1;
    s(ns).x=conversion*datos(m,1);
    s(ns).numero=round(datos(m,2));
    s(ns).puntos=conversion*datos(m+1:m+s(ns).numero,:);

    % calcula la poligonal entre cada dos puntos

    s(ns).tipo='';
    s(ns).poligonal=[];
    s(ns).limites=[];


for j=1:s(ns).numero-1

        puntoA=s(ns).puntos(j,:);
        puntoB=s(ns).puntos(j+1,:);

        if abs(puntoA(1)-puntoB(1))<10^(-15)
            s(ns).tipo=strvcat(s(ns).tipo,'Paralela al Eje z');
            s(ns).poligonal=[s(ns).poligonal;[puntoA(1),0]];

s(ns).limites=[s(ns).limites;[min(puntoA(2),puntoB(2)),max(puntoA(2),p
untoB(2))]];
        elseif abs(puntoA(2)-puntoB(2))<10^(-15)
            s(ns).tipo=strvcat(s(ns).tipo,'Paralela al Eje y');
            s(ns).poligonal=[s(ns).poligonal;[puntoA(2),0]];

s(ns).limites=[s(ns).limites;[min(puntoA(1),puntoB(1)),max(puntoA(1),p
untoB(1))]];
        else
            s(ns).tipo=strvcat(s(ns).tipo,'Oblicua');
            a=(puntoA(1)-puntoB(1))/(puntoA(2)-puntoB(2));
            b=puntoA(1)-a*puntoA(2);
            s(ns).poligonal=[s(ns).poligonal;[a,b]];

s(ns).limites=[s(ns).limites;[min(puntoA(1),puntoB(1)),max(puntoA(1),p
untoB(1))]];

        end
```

```
    end

    puntoA=s(ns).puntos(s(ns).numero,:);
    puntoB=s(ns).puntos(1,:);

    if abs(puntoA(1)-puntoB(1))<10^(-15)
        s(ns).tipo=strvcat(s(ns).tipo,'Paralela al Eje z');
        s(ns).poligonal=[s(ns).poligonal;[puntoA(1),0]];

s(ns).limites=[s(ns).limites;[min(puntoA(2),puntoB(2)),max(puntoA(2),p
untoB(2))]];
    elseif abs(puntoA(2)-puntoB(2))<10^(-15)
        s(ns).tipo=strvcat(s(ns).tipo,'Paralela al Eje y');
        s(ns).poligonal=[s(ns).poligonal;[puntoA(2),0]];

s(ns).limites=[s(ns).limites;[min(puntoA(1),puntoB(1)),max(puntoA(1),p
untoB(1))]];

     else
        s(ns).tipo=strvcat(s(ns).tipo,'Oblicua');
        a=(puntoA(1)-puntoB(1))/(puntoA(2)-puntoB(2));
        b=puntoA(1)-a*puntoA(2);
        s(ns).poligonal=[s(ns).poligonal;[a,b]];

s(ns).limites=[s(ns).limites;[min(puntoA(1),puntoB(1)),max(puntoA(1),p
untoB(1))]];

    end

    m=m+s(ns).numero+1;

end
```

7.4- Dibujo de las secciones

Está función dibujará cada una de las secciones de manera automática en nuestro programa. Utilizará la función *obtener_secciones* explicada anteriormente como variable de entrada, es decir, la información de cada una de las secciones la obtendrá de dicha función.

Nuestro programa hará que dibuje todas las secciones de varias formas diferentes a elegir, es decir, en la interfaz gráfica tendremos varias opciones las cuales serán:

- Dibujar la sección que queramos indicando el número de sección deseada.
- Dibujar un rango de secciones, indicando la sección inicial y la sección final.

- Dibujar todas las secciones de manera ordenada y con un pause entre cada sección.
- Dibujar las secciones de manera agrupada en grupos de m x n (m filas, m columnas).
- Dibujar un rango de secciones superpuestas, indicando la sección inicial y la sección final.

Utilizaremos algunos comandos para que cuando realice la imagen, MATLAB la centre y la escale.

```
function dibujar_secciones(s,varargin)

% Este programa dibuja las diferentes secciones de un barco
% dados unos ciertos puntos de control por sección
% dibujar_secciones(s,varargin)
% Variables de entrada:
% s estructura que contiene la información de las secciones.
%   Sus campos son los siguientes:
%   s.x  posición espacial x de la sección
%   s.numero  número de puntos de control en la sección considerada
%   s.puntos coordenadas (y,z) de los puntos de control
%   s.tipo tipo de poligonal entre un punto y el siguiente. Puede
%          ser paralela al eje y, paralela al eje z, u oblicua
%   s.poligonal información necesaria para construir la poligonal
%   s.limites indica de dónde a dónde se extiende cada línea de la
%             poligonal
% k argumento opcional que indica el número de sección que se quiere
%   dibujar
% rango  vector [k1,k2] que indica desde que sección a qué sección se
%   quiere dibujar
% all dibuja todas las secciones secuencialmente con una pausa entre
%     ellas
% grupo,m,p  las dibuja agrupadas en grupos de m x p secciones
% superpuestas,[k1,k2]



% creamos una ventana nueva donde dibujar
h=figure('Tag','secciones','Name',[blanks(6),'Secciones'],'Position',[
100 100 1000 600]);
axis off

% Centramos y escalamos
set( h,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');

position = get( h,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h,'Position', position );
```

```
% número de secciones
n=length(s);

% calculamos los valores mínimos y máximos de las variables y,z

ym=min(s(1).puntos(:,1));
yM=max(s(1).puntos(:,1));
zm=min(s(1).puntos(:,2));
zM=max(s(1).puntos(:,2));

for i=2:n

    aux_ym=min(s(i).puntos(:,1));
    if aux_ym < ym
        ym=aux_ym;
    end
    aux_yM=max(s(i).puntos(:,1));
    if aux_yM>yM
        yM=aux_yM;
    end
    aux_zm=min(s(i).puntos(:,2));
    if aux_zm<zm
        zm=aux_zm;
    end
    aux_zM=max(s(i).puntos(:,2));
    if aux_zM>zM
        zM=aux_zM;
    end

end

dy=(yM-ym)/10;
dz=(zM-zm)/10;

if nargin==2

    in=varargin{1};

    if isnumeric(in)      % o una sección o un rango de secciones

        if length(in)==1    % es una sección



            aux=s(in).puntos;
            aux=[aux;aux(1,1:2)];
            plot(aux(:,1),aux(:,2));
            ym=min(aux(:,1));
            yM=max(aux(:,1));
            zm=min(aux(:,2));
            zM=max(aux(:,2));
            if yM~=ym
                dy=(yM-ym)/10;
            end
            if zM~=zm
```

```
                dz=(zM-zm)/10;
            end
            axis([ym-dy yM+dy zm-dz zM+dz]);
            title(['Sección ',num2str(in),'    x=',num2str(s(in).x)])
            xlabel('Eje y')
            ylabel('Eje z')

        elseif length(in)==2     % es un rango de secciones

            for k=in(1):in(2)

                aux=s(k).puntos;
                aux=[aux;aux(1,1:2)];
                plot(aux(:,1),aux(:,2));
                ym=min(aux(:,1));
                yM=max(aux(:,1));
                zm=min(aux(:,2));
                zM=max(aux(:,2));
                if yM~=ym
                    dy=(yM-ym)/10;
                end
                if zM~=zm
                    dz=(zM-zm)/10;
                end
                axis([ym-dy yM+dy zm-dz zM+dz]);
                title(['Sección ',num2str(k),'    x=',num2str(s(k).x)])
                xlabel('Eje y')
                ylabel('Eje z')
                pause

            end
        end

    elseif ischar(in)    % todas las secciones


        for k=1:n

            aux=s(k).puntos;
            aux=[aux;aux(1,1:2)];
            plot(aux(:,1),aux(:,2));
            ym=min(aux(:,1));
            yM=max(aux(:,1));
            zm=min(aux(:,2));
            zM=max(aux(:,2));
            if yM~=ym
                dy=(yM-ym)/10;
            end

            if zM~=zm
                dz=(zM-zm)/10;
            end
            axis([ym-dy yM+dy zm-dz zM+dz]);
            title(['Sección ',num2str(k),'    x=',num2str(s(k).x)])
            xlabel('Eje y')
            ylabel('Eje z')
            pause
```

```
        end


    end

elseif nargin==3

    in=varargin{2};

    for k=in(1):in(2)

        aux=s(k).puntos;
        aux=[aux;aux(1,1:2)];
        plot(aux(:,1),aux(:,2));
        hold on
        ym=min([ym,min(aux(:,1))]);
        yM=max([yM,max(aux(:,1))]);
        zm=min([zm,min(aux(:,2))]);
        zM=max([zM,max(aux(:,2))]);
        if yM~=ym
            dy=(yM-ym)/10;
        end
        if zM~=zm
            dz=(zM-zm)/10;
        end
        axis([ym-dy yM+dy zm-dz zM+dz]);
        title(['Sección ',num2str(k),'   x=',num2str(s(k).x)])
        xlabel('Eje y')
        ylabel('Eje z')
        pause

    end

elseif nargin==4

    m=varargin{2};
    p=varargin{3};
    mp=m*p;

    co=floor(n/mp);           % número de plots a hacer menos 1
    re=rem(n,mp);             % número de secciones en el último plot


    for np=1:co


        for k=1:mp




            indice=mp*(np-1)+k;
            aux=s(indice).puntos;
            aux=[aux;aux(1,1:2)];
            subplot(m,p,k);
            plot(aux(:,1),aux(:,2));
```

```
            ym=min(aux(:,1));
            yM=max(aux(:,1));
            zm=min(aux(:,2));
            zM=max(aux(:,2));
            if yM~=ym
                dy=(yM-ym)/10;
            end
            if zM~=zm
                dz=(zM-zm)/10;
            end
            axis([ym-dy yM+dy zm-dz zM+dz]);
            title(['Sección                         ',num2str(indice),'
x=',num2str(s(indice).x)])
            xlabel('Eje y')
            ylabel('Eje z')

        end

        pause


    end


    hold off
    close(h)


 % creamos una ventana nueva donde dibujar

h=figure('Tag','secciones','Name',[blanks(6),'Secciones'],'Position',[
100 100 1000 600]);
    axis off


    for k=1:re


        indice=mp*co+k;
        aux=s(indice).puntos;
        aux=[aux;aux(1,1:2)];
        subplot(m,p,k);
        plot(aux(:,1),aux(:,2));
        ym=min(aux(:,1));
        yM=max(aux(:,1));
        zm=min(aux(:,2));
        zM=max(aux(:,2));
        if yM~=ym
            dy=(yM-ym)/10;
        end
        if zM~=zm
            dz=(zM-zm)/10;
        end
```

```
        axis([ym-dy yM+dy zm-dz zM+dz]);
        title(['Sección                              ',num2str(indice),'
x=',num2str(s(indice).x)])
        xlabel('Eje y')
        ylabel('Eje z')

    end


end
```

7.5- Obtener líneas de agua.

Las líneas de agua son planos horizontales que cortan al casco del buque, la intersección entre dicho plano y el casco es a lo que llamamos línea de flotación. Para cada x a lo largo de la eslora, obtendremos una y (semimanga). Las líneas de agua estarán pues en función del calado, por ejemplo para un calado, T (coordenada z) tendremos una línea de agua. Está función creará tantas líneas de agua como queramos en nuestro buque, entre una altura mínima y otra máxima. Para la altura máxima, el buque quedará totalmente sumergido. La estructura anterior “s”, explicada anteriormente, nos dará la información de cada semimanga necesaria.

Obtendremos un archivo *xls con todas las líneas de agua que queramos, dadas una x (eslora) y una z (calado), tendremos su correspondiente y (semimanga).

En nuestra interfaz gráfica tendremos la opción de poner cuantas líneas de agua deseamos y hasta que calado queremos dichas líneas de agua.

```
function [semimanga]=obtener_LineasDeAgua(s,nl,fichero,varargin)

% Esta función calcula para cada z entre la altura mínima y la altura
% máxima del barco (o el calado máximo introducido)
% tantas líneas de agua como sean necesarias
% [z,semimanga]=obtener_LineasDeAgua(s,nl,varargin)
% Variables de entrada
% s estructura que contiene la información de las secciones.
%   Sus campos son los siguientes:
%   s.x  posición espacial x de la sección
%   s.numero  número de puntos de control en la sección considerada
%   s.puntos coordenadas (y,z) de los puntos de control
%   s.tipo tipo de poligonal entre un punto y el siguiente. Puede
%          ser paralela al eje y, paralela al eje z, u oblicua
%   s.poligonal información necesaria para construir la poligonal
```

```
%   s.limites indica de dónde a dónde se extiende cada línea de la
%              poligonal
% nl número de líneas de agua que se quieren calcular
% fichero nombre del fichero donde se quieren guardar las lineas de
agua
% calado valor máximo de la altura para el que se quieren realizar
%         los cálculos
% Variables de salida:
% semimanga contiene lo siguiente:
%            En la posición (1,1) un cero que no indica nada
%               En la primera fila, a partir de la segunda celda, la
coordenada
%                x de las cuadernas
%            En la primera columna, a partir de la segunda celda, la
%            coordenada z que indica la altura
%               En las celdas (2:end, 2:end)  las semimangas para los
valores
%                de x y de z correspondientes



% número de secciones
n=length(s);

% calculamos el valor mínimo y máximo de la variable z

zm=min(s(1).puntos(:,2));
zM=max(s(1).puntos(:,2));

for i=2:n

    aux_zm=min(s(i).puntos(:,2));
    if aux_zm<zm
        zm=aux_zm;
    end
    aux_zM=max(s(i).puntos(:,2));
    if aux_zM>zM
        zM=aux_zM;
    end

end


% comprueba si quiere todas las alturas de calado, o
% desde un calado máximo dado

if nargin==4
    aux=zM;
    zM=varargin{1};
    if zM>aux
         uiwait(msgbox('El calado introducido es mayor que el máximo
permitido', 'Mensaje de error',...
        'error','modal'));
         return;
    end
end
```

```
% calculamos la altura de las diferentes líneas de agua

z=fliplr(linspace(zm,zM,nl));

% calculamos la semimanga para cada z y para cada sección
% y la guardamos en la matriz semimanga
% La primera fila de semimanga contiene la línea de agua
% a altura máxima del barco

for i=1:nl

    % cortamos la recta z=altura de la línea de agua
    % con cada una de las secciones

    for j=1:n

        % miramos si la recta corta a cada una de los
        % lados de la poligonal que forma la sección

        semimanga(i,j)=0;

        for k=1:s(j).numero

            if strcmp(deblank(s(j).tipo(k,:)),'Paralela al Eje z')
                if              (z(i)>=s(j).limites(k,1))           &
(z(i)<=s(j).limites(k,2))
                    if s(j).poligonal(k,1)>semimanga(i,j)
                        semimanga(i,j)=s(j).poligonal(k,1);
                    end
                end
            elseif strcmp(deblank(s(j).tipo(k,:)),'Paralela al Eje y')
                if abs(z(i)-s(j).poligonal(k,1))<10^(-15)
                    if s(j).limites(k,2)>semimanga(i,j)
                        semimanga(i,j)=s(j).limites(k,2);
                    end
                end
            elseif strcmp(deblank(s(j).tipo(k,:)),'Oblicua')
                aux=s(j).poligonal(k,1)*z(i)+s(j).poligonal(k,2);
                if s(j).limites(k,1)<=aux & aux<=s(j).limites(k,2)
                    if aux>semimanga(i,j)
                        semimanga(i,j)=aux;
                    end
                end
            end

        end




    end

end

% generamos las coordenadas x de las secciones de popa a proa
```

```
for i=1:length(s)
    x(i)=s(i).x;
end

x=fliplr(-x);

% ordenamos las semimangas de popa a proa

semimanga=fliplr(semimanga);

% ponemos el vector de cuadernas en la primera fila de la
% matriz de semimangas

semimanga=[x;semimanga];

% ponemos el vector de alturas z en la primera columna, completando
% la primera entrada con el valor de 0

z=[0;z'];
semimanga=[z,semimanga];

% guardamos en un fichero excel .xls la matriz semimanga

success = xlswrite(['./Líneas de Agua/',fichero],semimanga);
```

7.6- Dibujar líneas de agua

Cómo en el apartado anterior *"8.4- Dibujar secciones"*, esta función nos dibujará en 3D las líneas de agua que hayamos querido generar. En este caso, no tendremos las diferentes opciones que teníamos con la función "dibujar_secciones", si no que el programa de forma automática, nos mostrará una imagen en 3D centrada y escalada de las líneas de agua del buque.

```
function dibujar_LineasDeAgua(fichero)

% Esta función dibuja en 3D las líneas de agua de un barco
% contenidas en un fichero dentro del directorio ./Líneas de Agua
% dibujar_LineasDeAgua(fichero)
% Variables de entrada:
% fichero nombre del archivo que contiene las líneas de agua del barco


% Lee las alturas de las líneas de agua, las secciones, y las
semimangas
```

```
a=xlsread(['./Líneas de Agua/',fichero]);
[n,m]=size(a);

% secciones
x=a(1,2:m);

% alturas
z=a(2:n,1);

% semimangas
y=a(2:n,2:m);


% Creamos una figura
h=figure('Tag','lineas    de    Agua','Name',[blanks(6),'Líneas    de
Agua'],'Position',[100 100 1000 600]);
axis off

% Centramos y escalamos
set( h,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');

position = get( h,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h,'Position', position );

% dibujamos las líneas de agua

for k=1:n-1
    y_linea = y(k,1:m-1);
    z_linea = ones(1,m-1)*z(k);
    plot3(x,y_linea,z_linea,'LineW',2)
    hold on
    plot3(x,-y_linea,z_linea,'LineW',2)
    hold on
end

% Ponemos las indicaciones de las variables


grid on
box on
zlabel('Calado [m]','FontW','Bold')
xlabel('Secciones [m]','FontW','Bold')
ylabel('Semimangas [m]','FontW','Bold')
title('Imagen 3D de las líneas de agua','FontW','Bold')
set(gca,'FontW','Bold')
```

7.7- Cálculos curvas hidrostáticas

En este apartado crearemos una función capaz de calcular todos los cálculos hidrostáticos descritos en el apartado “curvas hidrostáticas o propiedades de la carena recta”, a partir de los archivos que contienen la información sobre las secciones y las líneas de agua. Haremos el cálculo de áreas/volúmenes utilizando la regla de los trapecios, ya que al tener todas las secciones/líneas de agua deseadas el error cometido aproximadamente no deferirá mucho, con relación a los cálculos realizados con las reglas de Simpson y por lo tanto el resultado de los cálculos será bastante bueno, sin dejar de ser aproximado.

A continuación realizaremos una función llamada “trapecio”, la cual llamaremos cada vez que tengamos que utilizar la regla de los trapecios para realizar los pertinentes cálculos. El objetivo es generalizar y facilitar lo máximo posible el programa para poder utilizarlo con otros buques.

```
function integral=trapecios(x,y)

% Esta función aproxima el cálculo de una integral por el método
% de los trapecios
% integral=trapecios(x,y);
% Variables de entrada:
% x vector de abscisas
% y vector de ordenadas
% Variables de salida:
% integral valor aproximado de la integral
% número de abscisas


n=length(x);

if n<=1

    integral=0;

else

    % aplicamos la regla simple de los trapecios n-1 veces

    h=diff(x);

    % fórmula que nos aproxima la integral

    integral=sum(((y(1:n-1)+y(2:n))/2).*h);

end
```

Seguidamente comenzamos a realizar los cálculos, que serán los siguientes:

- Área de flotación para cada línea de agua, (Aw).
- Momento del área de flotación, (Mx).
- Centro de flotación, posición longitudinal (x), (Xf).
- Momento de inercia transversal, (It).
- Momento de inercia de flotación, (Iy).
- Momento de inercia longitudinal, (Il).
- Toneladas por centímetro de inmersión, (TPC).
- Volumen de desplazamiento o carena, (V).
- Momento de desplazamiento de volumen, (MB).
- Centro vertical de carena, (ZB).
- Radio metacéntrico transversal, (BMt).
- Radio metacéntrico longitudinal, (BMl).
- Área de la sección, (As).
- Momento respecto de la sección, (MBx).
- Centro longitudinal de carena, (XB).

La función nos mostrará los cálculos en forma de gráfica y en función del calado, además de mostrar los resultados numéricamente.

```
function  curvas_hidrostaticas(fichero,op,ksec)

% Esta función realiza cálculos hidrostáticos a partir
% del conocimiento de la posición de las secciones, y
% de las semimangas para diferentes calados
% Los cálculos hidrostáticos que realiza son:
% Aw  Área de flotación para cada línea de agua
% Mx  Momento de área del plano del agua
% Xf  Centro de flotación
% It  Momento de inercia transversal
% Iy  Momento de inercia del plano de agua
% IL  Momento de inercia longitudinal
% TPC Toneladas por centimetro de inmersión
% V   Volumen de desplazamiento o de carena
% MB  Momento de desplazamiento del volumen
% ZB  Centro vertical de carena
% BMt Radio metacéntrico transversal
% BMl Radio metacéntrico longitudinal
% As  Área de las secciones
% MBx Momento respecto de la sección (sin curva)
% XB  Centro longitudinal de carena  (sin curva)
%
% curvas_hidrostaticas(fichero,op,ksec)
% fichero es el nombre del archivo excel que contiene las lineas de
agua
```

```
% op indica si quiere gráfica de los distintos elementos o no
%    op='sssssssssssn' indica que se quieren todas las gráficas menos
la
%    de As
%    Notar que se trata de 13 curvas hidrostáticas distintos
% ksec número de sección del cual se quiere saber el área


% Leemos el fichero con la posición de las secciones, y
% las semimangas para cada altura z

a=xlsread(['./Líneas de Agua/',fichero]);
[n,m]=size(a);

% secciones
x=a(1,2:m);     % tamaño 1 x m-1

% alturas
z=a(2:n,1);     % tamaño n-1 x 1

% semimangas
y=a(2:n,2:m);   % tamaño n-1 x m-1




% Densidad del aguas en el oceano en el Sistema Internacional
pw = 1.025;


% Realiza el cálculo de los parámetros hidrostáticos de cada uno de
los calados

for k=1:n-1

    y_linea=y(k,1:m-1);
    Aw(k)=2*trapecios(x,y_linea);               % Área de flotación de
cada línea de agua
    Mx(k)  = 2*trapecios(x,x.*y_linea);          % Momento de área del
plano del agua
    Xf(k)  = Mx(k)/Aw(k);                       % Centro de flotación
    It(k)  = 2/3*trapecios(x,y_linea.^3);       % Momento transversal de
inercia
    Iy(k)  = 2*trapecios(x,(x.^2).*y_linea); % Momento de inercia del
plano de agua
    IL(k)  = Iy(k)+ Xf(k)^2* Aw(k);             % Momento longitudinal de
inercia
    TPC (k)  = Aw(k)*pw/100;                    % Toneladas por centimetro
de inmersión
end

for k=1:n-1
    V(k)=-trapecios(z(k:n-1)',Aw(k:n-1));              %    Volumen    de
desplazamiento para cada calado
    MB(k)=-trapecios(z(k:n-1)',z(k:n-1)'.*Aw(k:n-1));    %   Momento   de
desplazamiento respecto de z
    ZB(k)=MB(k)/V(k);                               % Coordenada z del centro
```

```
de carena
end


for k=1:n-1
    BMt(k)=It(k)/V(k);                          % Radio metacéntrico
transversal
    BMl(k)=IL(k)/V(k);                          % Radio metacéntrico
longitudinal
end


% Realiza el cálculo de los parámetros hidrostáticos de cada uno de
las secciones

for i=1:m-1
    y_linea=y(1:n-1,i)';
    As(i)=-2*trapecios(z',y_linea);          % Área de cada sección
end

MBx=trapecios(x,x.*As);                       % Momento respecto de la
sección
XB=MBx/V(1);                                  % Centro Longitudinal de
la carena



% Dibujamos el área de flotación

if op(1)=='s'

h1=figure('Name',[blanks(6),'Área de flotación'],'Position',[100 100
1000 600]);

plot(Aw,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Área de Flotación','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h1,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h1,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h1,'Position', position );


end

% Dibujamos el Momento de área del plano del agua

if op(2)=='s'
```

```
h2=figure('Name',[blanks(6),'Momento   de   área   del   plano   del
agua'],'Position',[100 100 1000 600]);

plot(Mx,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Momento de área del plano del agua','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h2,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h2,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h2,'Position', position );

end

% Dibujamos el Centro de flotación

if op(3)=='s'

h3=figure('Name',[blanks(6),'Centro de flotación'],'Position',[100 100
1000 600]);

plot(Xf,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Centro de flotación','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h3,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h3,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h3,'Position', position );


end


% Dibujamos el Momento transversal de inercia

if op(4)=='s'

h4=figure('Name',[blanks(6),'Momento          transversal            de
inercia'],'Position',[100 100 1000 600]);

plot(It,z,'b','LineW',2)
```

```
grid on
box on
set(gca,'FontW','Bold')
xlabel('Momento transversal de inercia','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h4,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h4,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h4,'Position', position );


end


% Dibujamos el Momento de inercia del plano de agua

if op(5)=='s'

h5=figure('Name',[blanks(6),'Momento  de  inercia  del  plano  de
agua'],'Position',[100 100 1000 600]);

plot(Iy,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Momento de inercia del plano de agua','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h5,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h5,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h5,'Position', position );

end


% Dibujamos el Momento longitudinal de inercia

if op(6)=='s'

h6=figure('Name',[blanks(6),'Momento         longitudinal          de
inercia'],'Position',[100 100 1000 600]);

plot(IL,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
```

```
xlabel('Momento longitudinal de inercia','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h6,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h6,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h6,'Position', position );

end

% Dibujamos el Toneladas por centimetro de inmersión

if op(7)=='s'

h7=figure('Name',[blanks(6),'Toneladas por centimetro de inmersión'],'Position',[100 100 1000 600]);

plot(TPC,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Toneladas por centimetro de inmersión','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h7,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h7,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h7,'Position', position );

end

% Dibujamos el Volumen de desplazamiento

if op(8)=='s'

h8=figure('Name',[blanks(6),'Volumen de desplazamiento'],'Position',[100 100 1000 600]);

plot(V,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Volumen de desplazamiento','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h8,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h8,'Position');
```

```
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h8,'Position', position );

end

% Dibujamos el Momento de desplazamiento respecto de z

if op(9)=='s'

h9=figure('Name',[blanks(6),'Momento  de  desplazamiento  respecto  de 
z'],'Position',[100 100 1000 600]);

plot(MB,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Momento de desplazamiento respecto de z','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h9,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h9,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h9,'Position', position );

end

% Dibujamos el Coordenada z del centro de carena

if op(10)=='s'

h10=figure('Name',[blanks(6),'Coordenada    z    del    centro    de 
carena'],'Position',[100 100 1000 600]);

plot(ZB,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Coordenada z del centro de carena','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h10,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h10,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h10,'Position', position );

end
```

```
% Dibujamos el Radio metacéntrico transversal

if op(11)=='s'

h11=figure('Name',[blanks(6),'Radio                          metacéntrico
transversal'],'Position',[100 100 1000 600]);

plot(BMt,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Radio metacéntrico transversal','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h11,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h11,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h11,'Position', position );

end

% Dibujamos el Radio metacéntrico longitudinal

if op(12)=='s'

h12=figure('Name',[blanks(6),'Radio                          metacéntrico
longitudinal'],'Position',[100 100 1000 600]);

plot(BMl,z,'b','LineW',2)

grid on
box on
set(gca,'FontW','Bold')
xlabel('Radio metacéntrico longitudinal','FontW','Bold')
ylabel('Calado ','FontW','Bold')

% Centramos y escalamos
set( h12,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h12,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h12,'Position', position );

end

% Dibujamos el Área de las Secciones

if op(13)=='s'

h=figure('Name',[blanks(6),'Área  de  las  secciones'],'Position',[100
100 1000 600]);
plot(x,As,'b','LineW',2)
```

```
grid on
box on
set(gca,'FontW','Bold')
xlabel('Secciones','FontW','Bold')
ylabel('Área de las Secciones','FontW','Bold')

% Centramos y escalamos
set( h,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get( h,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set( h,'Position', position );

end

% Mandamos los datos a una figura en la pantalla
g=figure('Tag','datos','Name',[blanks(6),'Datos de
salida'],'Position',[100 100 1000 600]);
axis off

% Definimos una cadena de carácteres que contiene los datos de salida
str={['Calado',blanks(2),num2str(z(1))],['Área de flotación
(Aw):',blanks(2),num2str(Aw(1))],...
    ['Momento de área del plano del agua
(Mx):',blanks(2),num2str(Mx(1))],...
    ['Centro de flotación (Xf):',blanks(2),num2str(Xf(1))],...
    ['Momento de inercia transversal
(It):',blanks(2),num2str(It(1))],...
    ['Radio metacéntrico transversal
(BMt):',blanks(2),num2str(BMt(1))],...
    ['Momento de inercia del plano de agua
(Iy):',blanks(2),num2str(Iy(1))],...
    ['Momento de inercia longitudinal
(IL):',blanks(2),num2str(IL(1))],...
    ['Radio metacéntrico longitudinal
(BMl):',blanks(2),num2str(BMl(1))],...
    ['Volumen de carena (V):',blanks(2),num2str(V(1))],...
    ['Momento de desplazamiento (MB):',blanks(2),num2str(MB(1))],...
    ['Centro vertical de carena (ZB):',blanks(2),num2str(ZB(1))],...
    ['Toneladas por centímetro de inmersión
(TPC):',blanks(2),num2str(TPC(1))],...
    ['Centro longitudinal de carena (XB):',blanks(2),num2str(XB)],...
    ['Área de la sección (As)
',num2str(ksec),':',blanks(2),num2str(As(ksec))]};

g1=uicontrol('Style','Text','String',str,'FontSize',16,'FontWeight','b
old','Units','normalized',...
    'Position',[0 0 1
0.95],'Background','White','HorizontalAlignment','left');

g2=uicontrol('Style','Text','String', {'Valores Hidrostáticos'},
'FontSize',18,'FontWeight','bold',...
    'Units','normalized','Position',[0 0.95 1
0.05],'Background','green','HorizontalAlignment','center','ForegroundC
olor','blue');
```

```
% Centramos y escalamos
set(g,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');
position = get(g,'Position');
position(1) = (screenSize(3)-position(3))/2;
position(2) = (screenSize(4)-position(4))/2;
set(g,'Position', position );
```

7.8- Interfaz gráfica

Para poder probar todos los cálculos realizados de manera más rápida y sencilla se ha desarrollado una interfaz gráfica en el entorno de desarrollo *GUIDE (Graphical User Interface Development Environment)* que nos proporciona MATLAB. Además, esta interfaz ha sido testeada ante posibles fallos que pueda cometer el usuario a la hora de introducir los distintos valores. También es fácilmente ampliable ante la eventualidad de añadir nuevas funciones a calcular.

7.8.1-Ejecución de la interfaz gráfica

Para ejecutar la interfaz gráfica principal, debemos introducir en la línea de comando de MATLAB

>> *hidrostática*

Antes de ejecutar dicha instrucción debemos asegurarnos de que nos encontramos en el directorio que contiene la GUI, en caso contrario debemos cambiar nuestro directorio de trabajo por aquel que la contenga. Al ejecutar este comando, nos aparecerá la ventana que se muestra en la Figura 7.1.

Figura 7.1 – Interfaz Gráfica de Usuario

7.8.2-Documentación de la Interfaz Gráfica

A continuación explicaremos por partes los botones que contiene nuestra interfaz. La dividiremos en varias partes:

- Archivos a cargar
- Líneas de agua
- Dibujo y área de la sección
- Variables hidrostáticas
- Botones principales
- Barra de tareas

En los archivos a cargar será donde cargaremos nuestros archivos con toda la información, en cuanto a formas del barco, Figura 7.2.

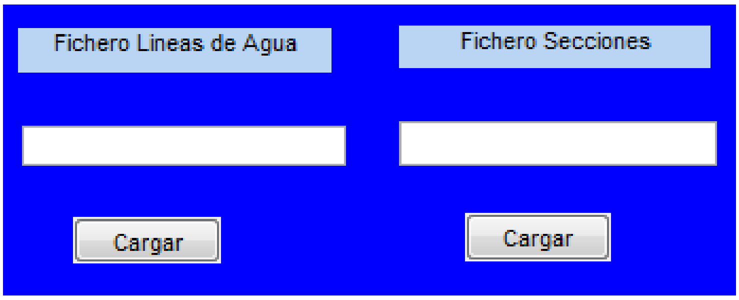


Figura 7.2 – Archivos a cargar

En el apartado líneas de agua, tendremos dos botones, uno será el de generar líneas de agua, el cual nos generará un archivo *xls, a partir de nuestro archivo de secciones. Diremos las líneas de agua que deseamos y hasta un cierto calado. El segundo botón nos dibujará las líneas de agua, Figura 7.3

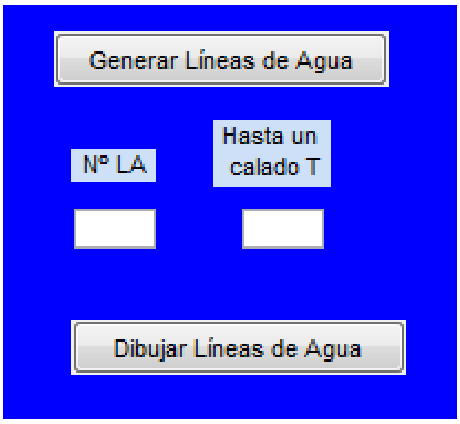


Figura 7.3 – Líneas de agua

Dibujo y área de la sección, seleccionaremos cómo queremos que nos dibuje las secciones, tenemos cinco opciones diferentes:

- Dibujar la sección que queramos indicando el número de sección deseada.
- Dibujar un rango de secciones, indicando la sección inicial y la sección final.
- Dibujar todas las secciones de manera ordenada y con un pause entre cada sección.
- Dibujar las secciones de manera agrupada en grupos de m x n (m filas, m columnas).
- Dibujar un rango de secciones superpuestas, indicando la sección inicial y la sección final.

El botón dibujar nos generará una imagen o varias con la selección de secciones que hayamos seleccionado. En la parte, área de secciones, nos calculará el área de sección que le indiquemos, Figura 7.4.

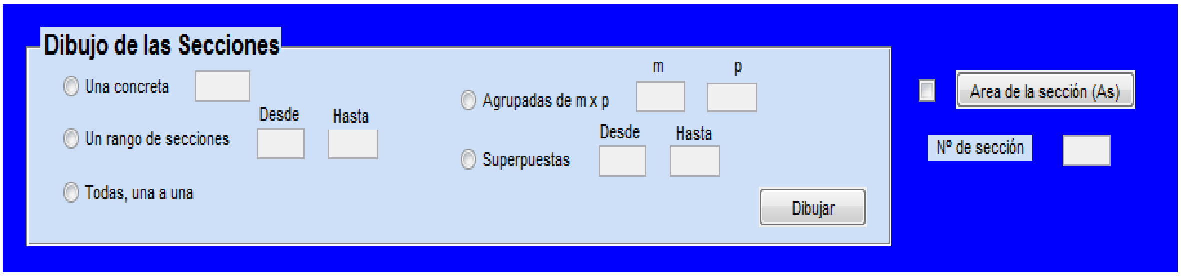


Figura 7.4 – Dibujo y área de la sección

En el panel, variables hidrostáticas, podremos seleccionar que variable queremos calcular, tanto de forma gráfica hasta un cierto calado T, como numéricamente, figura 7.5.

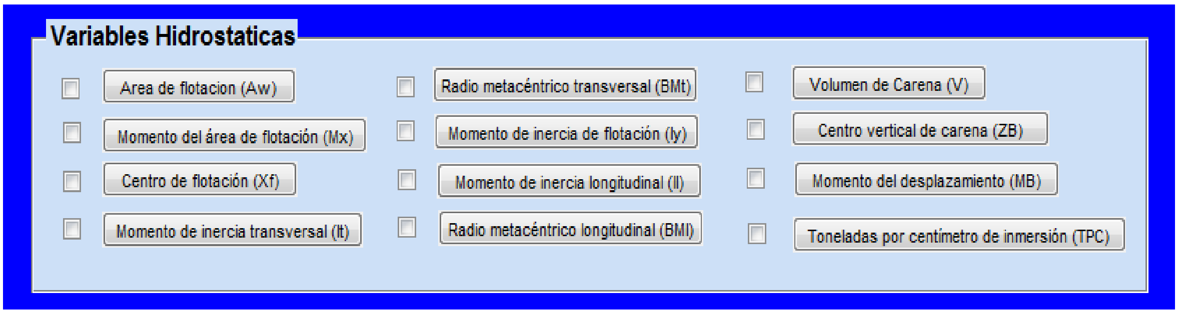


Figura 7.5 – Variables hidrostáticas

Botones principales, en la parte inferior de la interfaz gráfica, tendremos cuatro botones principales: (Figura 7.6)

- Aplicar: Nos realizará los cálculos, tanto gráficamente, como numéricamente de las variables hidrostáticas seleccionadas.
- Resetear: Pulsando este botón nos borrará todo lo introducido y volveremos a la condición inicial.
- Cerrar gráficas: Cerrará todas las gráficas abiertas.
- Salir: Cerrará el programa por completo.

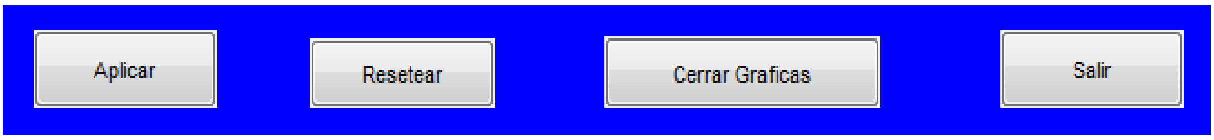


Figura 7.6 – Botones principales

Barra de tareas, tendremos un solo botón llamado información, el cual desplegará un menú con cuatro botones diferentes que serán los siguientes: (Figura 7.7)

- Acerca De
- Tutorial de la interfaz
- Pdf del proyecto
- Salir

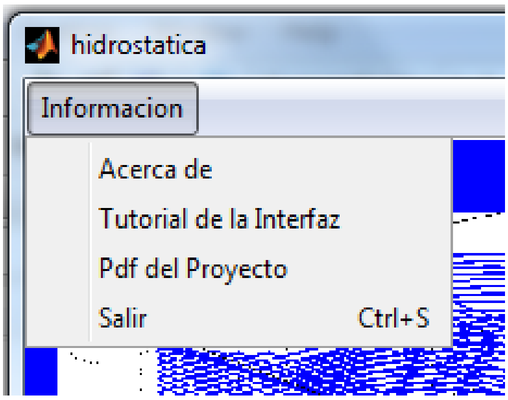


Figura 7.7 – Barra de tareas

7.8.3-Cómo utilizar la interfaz gráfica

Una vez ejecutado el programa y explicado por partes cada uno de los botones de la interfaz, comenzaremos a explicar de una forma detallada cómo utilizar el programa.

Lo principal y primera cosa que necesitamos para hacer funcionar el programa es, el archivo *gf con toda la información de las formas del buque, en cuanto a secciones. Como hemos visto anteriormente, este archivo podremos obtenerlo con el software Rhinoceros, utilizando la herramienta “Asociar datos GHS”.

Lo primero que haremos será cargar este archivo, en la parte “Fichero de Secciones”, pulsaremos el botón cargar, se nos abrirá una ventana, dónde deberemos buscar nuestro archivo y pulsar en abrir. En tipo de archivo, solo tendremos una opción que será “Archivo que contiene las secciones del barco (*gf)”, figura 7.8.

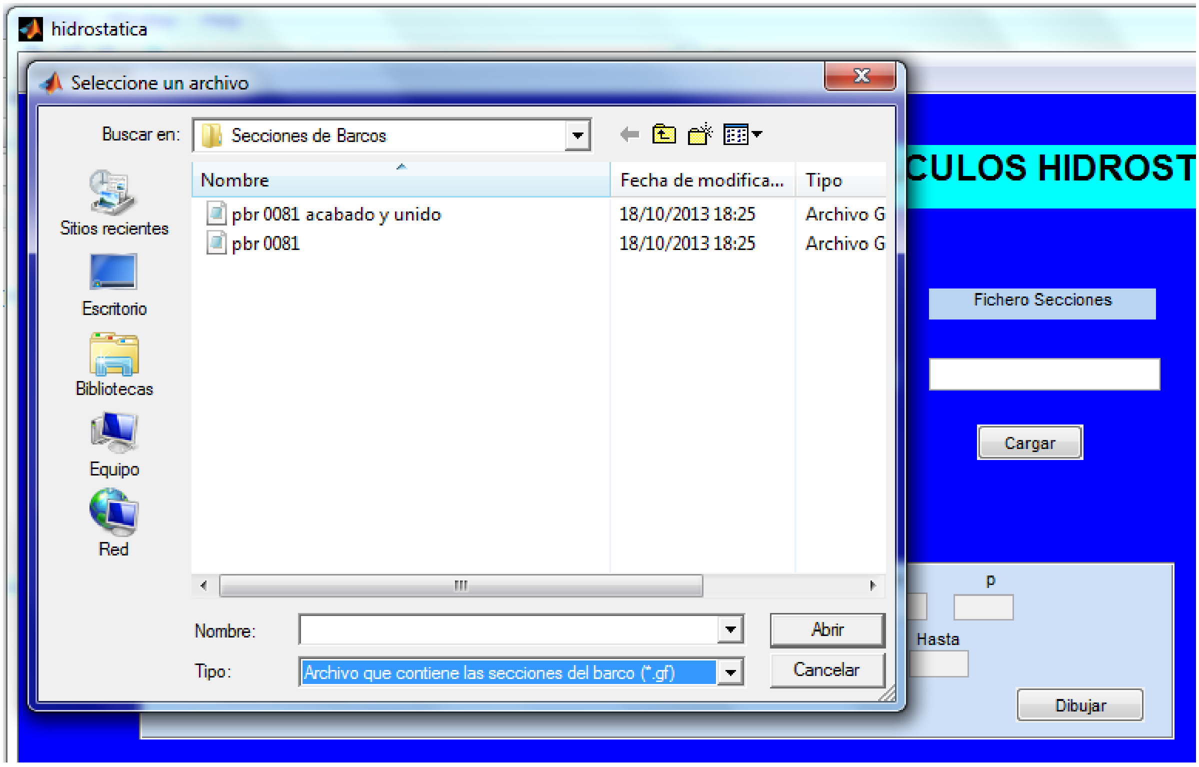


Figura 7.8 – Cargar archivo secciones de barcos (*gf)

En caso de no cargar ningún archivo y pulsar algún botón, como por ejemplo, generar líneas de agua, dibujar secciones o seleccionar alguna variable hidrostática y pulsar aplicar, nos saldrá un mensaje de error, el cual nos pedirá lo que se necesita para realizar los pertinentes cálculos, es decir, por ejemplo, queremos realizar el cálculo del volumen de carena de nuestro barco, entonces, necesitaremos el archivo de líneas de agua, por lo que nos saldrá el siguiente mensaje de error (figura 7.9).

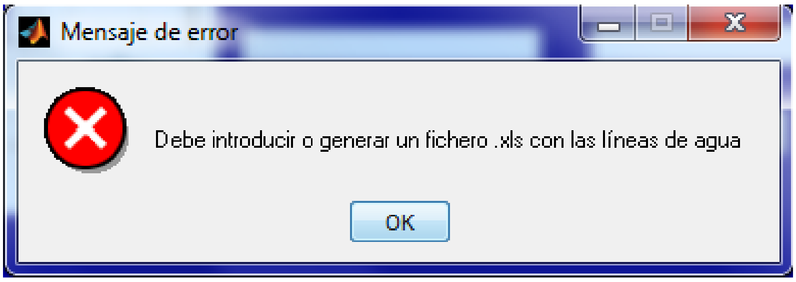


Figura 7.9 – Mensaje de error

Si por consiguiente, pulsamos el botón generar líneas de agua para generar dicho archivo, nos saldrá el siguiente mensaje de error, Figura 7.10.

Figura 7.10 – Mensaje de error

En conclusión, lo primero e indispensable que debemos hacer es cargar nuestro archivo *gf con las secciones del buque. En este tutorial/explicación de cómo utilizar nuestro programa, cargaremos como ejemplo el archivo de nuestro barco con nombre "pbr 0081.gf"

Con el archivo cargado, seguidamente podremos generar nuestro archivo de líneas de agua, para ello debemos indicar el número de líneas de agua y hasta un calado T, el cual indica hasta donde queremos generar nuestras líneas de agua. En caso de introducir algún dato erróneo, el programa nos informará del error, por ejemplo, para 20 líneas de agua introducidas, introducimos un calado erróneo o ningún calado, entonces aparecerá, (figura 7.11)

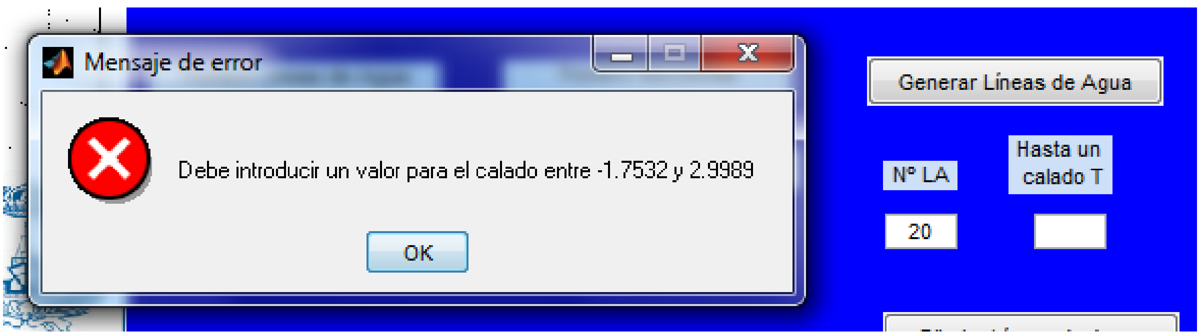


Figura 7.11 – Mensaje de error

El programa automáticamente nos calculará el calado mínimo y el calado máximo y nos dirá que introduzcamos un valor del calado entre esos valores.

Indicando correctamente las líneas de agua deseadas y el calado, pulsamos el botón "Generar líneas de agua", el programa nos generará un archivo *xls con las líneas de agua deseadas y automáticamente lo cargará en "Fichero Líneas de Agua". El archivo se guardará en nuestro directorio en la carpeta “Líneas de Agua” con el mismo nombre que el archivo *gf, pero con la extensión *xls.

Con el fichero de las líneas de agua generado y cargado, podremos obtener nuestra figura de las líneas de agua en 3D, pulsando del botón “Dibujar Líneas de Agua”, Figura 7.12.

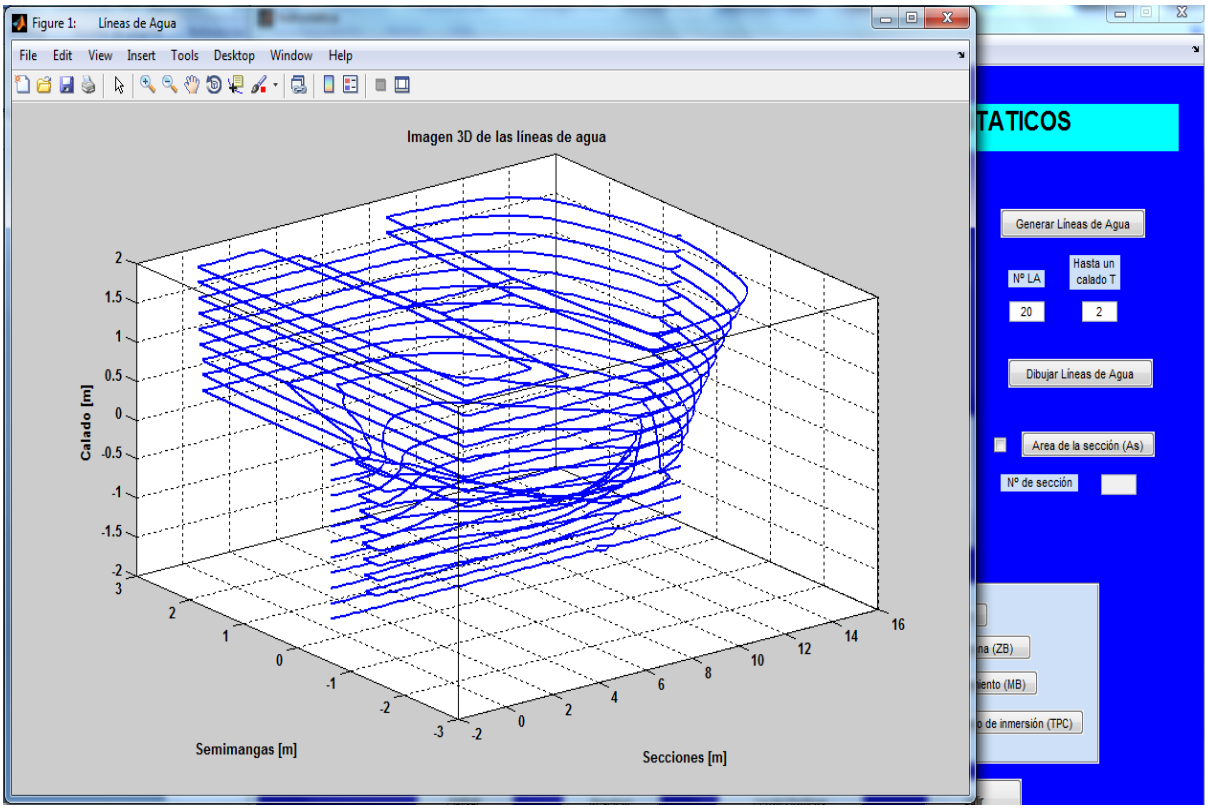


Figura 7.12 – Líneas de Agua en 3D

En el panel “Dibujo de las Secciones” podremos elegir la forma de dibujar nuestras secciones, para este apartado no necesitaremos generar o cargar nuestro fichero de las líneas de agua. Activamos la opción o forma con la que queremos dibujar las secciones, por ejemplo, superpuestas, e indicaremos desde que sección hasta que sección queremos dibujar, después pulsamos el botón “Dibujar”. En caso de elegir una sección errónea, nos saldrá un mensaje de error, indicándonos que el valor introducido no es correcto y que dicho valor deberá estar entre un cierto número, dependiendo del número de secciones que tengamos. A continuación mostraremos la imagen de las secciones superpuestas desde la sección 40 hasta la 50, Figura 7.13.

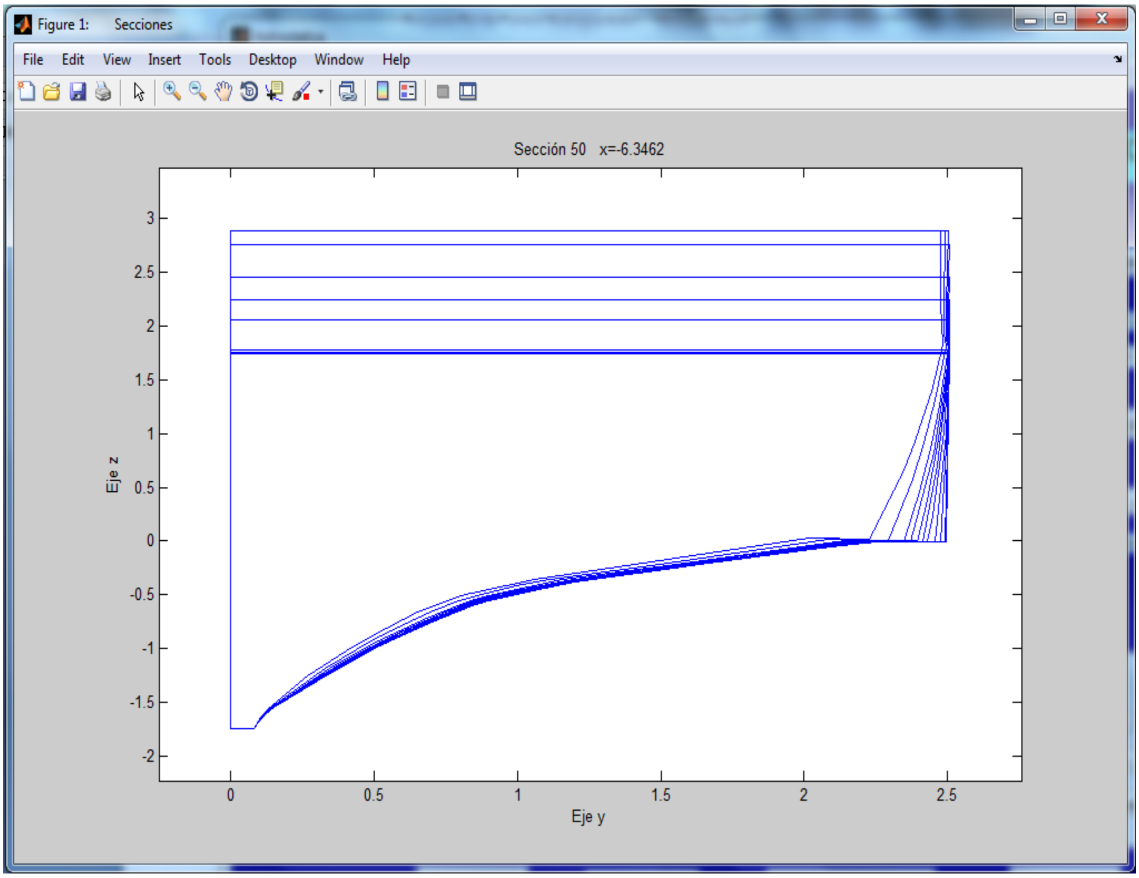


Figura 7.13 – Dibujo de las secciones

En el panel “Variables Hidrostáticas” marcaremos las variables que queremos calcular, estos cálculos irán en función del calado, por lo que será necesario haber generado antes el “Fichero Líneas de Agua” y por consiguiente las variables serán calculadas hasta el calado T que hayamos indicado al generar el “Fichero Líneas de Agua”. Una vez seleccionadas las variables a calcular, pincharemos en el botón “Aplicar” y obtendremos los resultados, tanto gráficamente, como numéricamente.

Otra función de nuestro programa consistirá, en que al pulsar uno de los botones, en el panel “Variables Hidrostáticas”, se nos abrirá un archivo *pdf con la descripción detallada de la variable hidrostática pulsada.

A continuación calcularemos el volumen de carena de nuestra embarcación, con el archivo de líneas de agua generado anteriormente, obteniendo la siguiente figura, Figura 7.14.

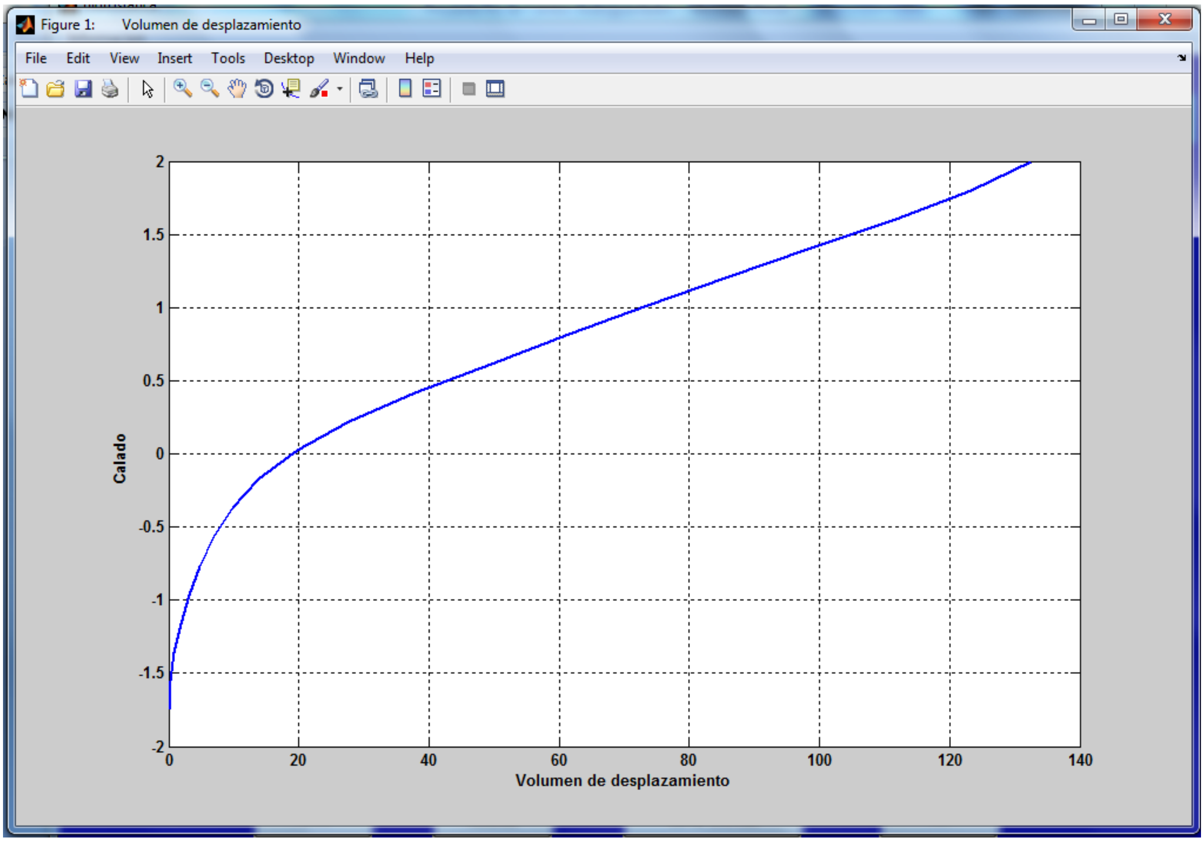
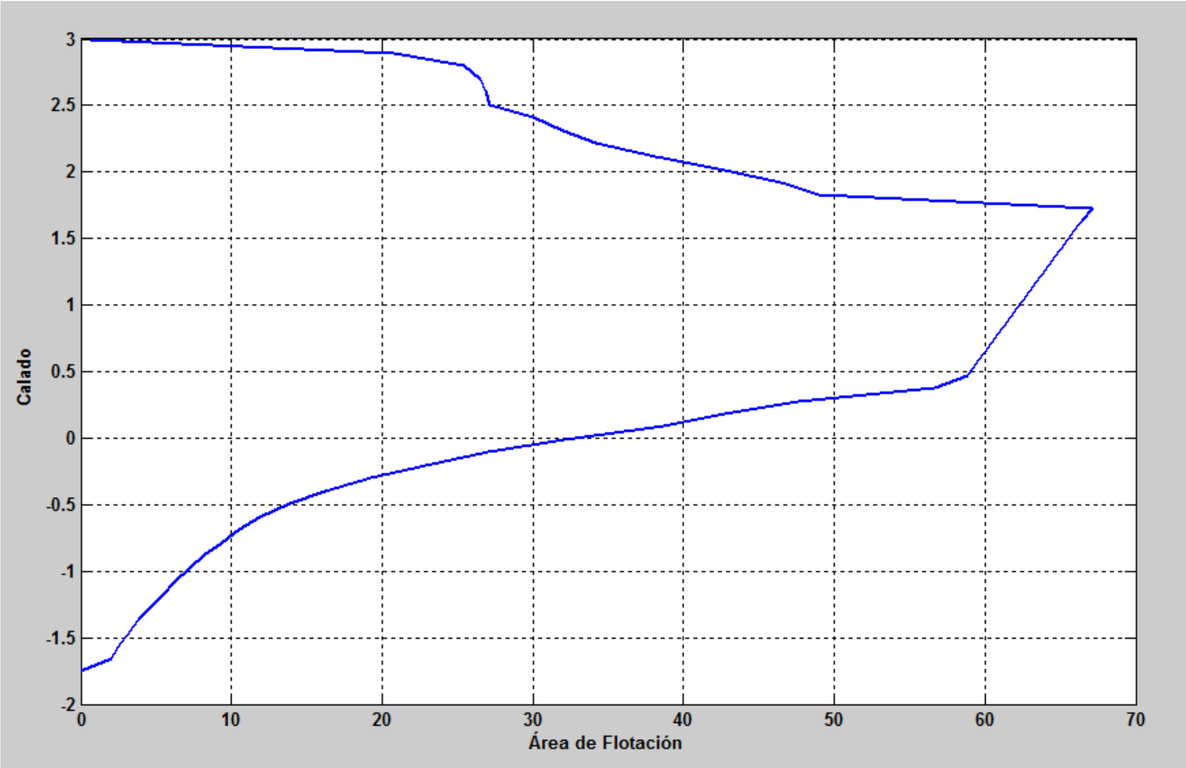


Figura 7.14 – Curva Volumen de Carena

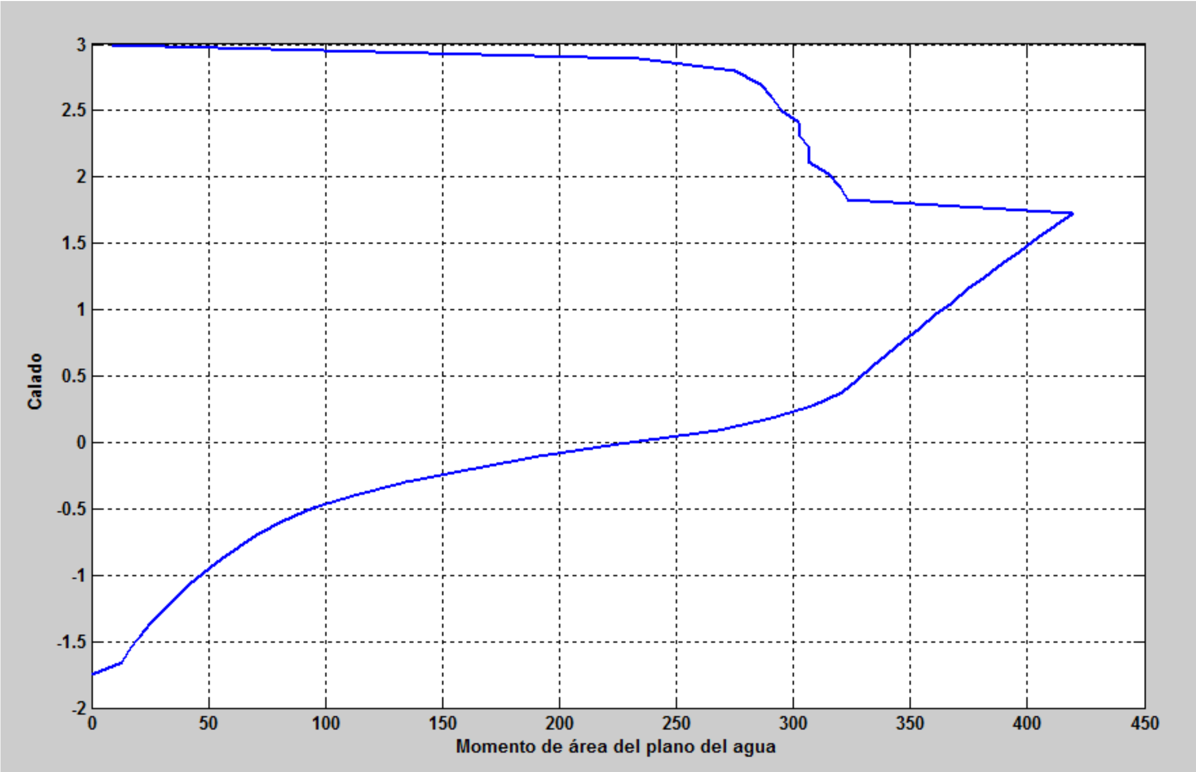
La estructura de la interfaz es sencilla, fácil de usar y con funciones básicas y claras, con botones explicativos y un tutorial/ayuda de cómo utilizar el programa.

8- Valoración de los resultados

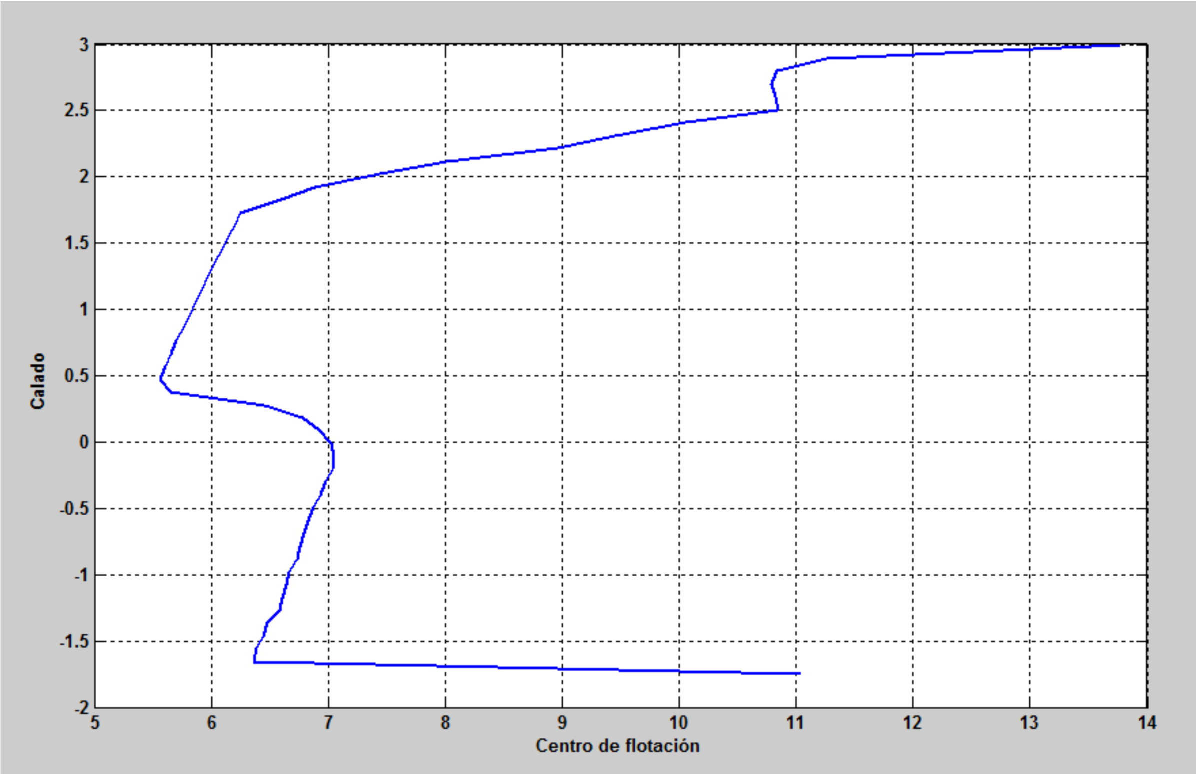
El objetivo de este capítulo es mostrar lo que verdaderamente importa a la hora de la verdad, es decir, a la hora de trabajar, y son los resultados. Mostraremos cada una de las gráficas de las propiedades geométricas de nuestro barco, en su condición de adrizado, ya que las líneas de agua creadas son totalmente horizontales y para un calado máximo, haciendo que el buque quede totalmente sumergido.



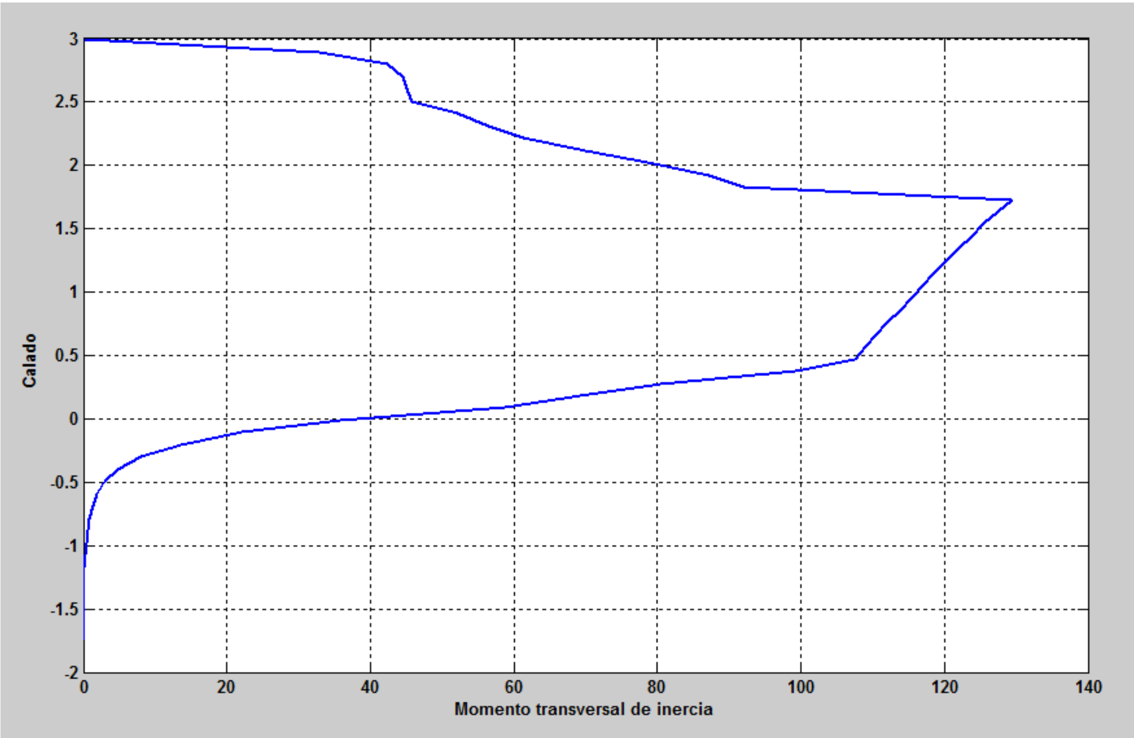
Área de la flotación



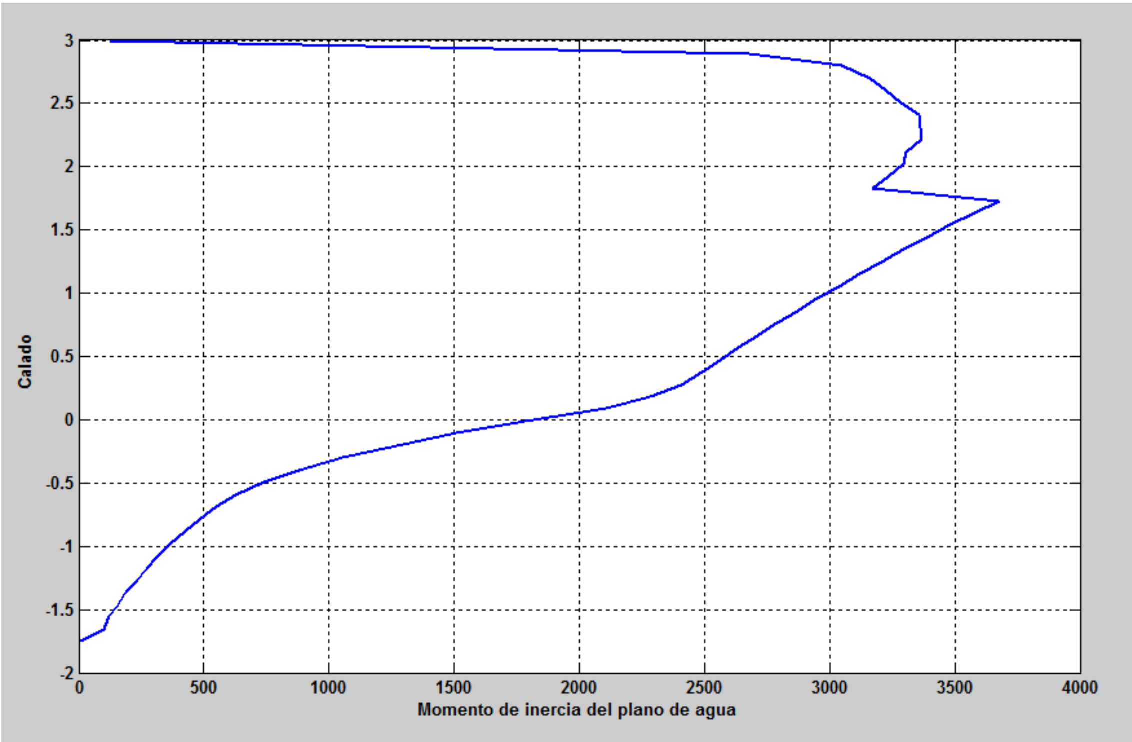
Momento del área de la flotación



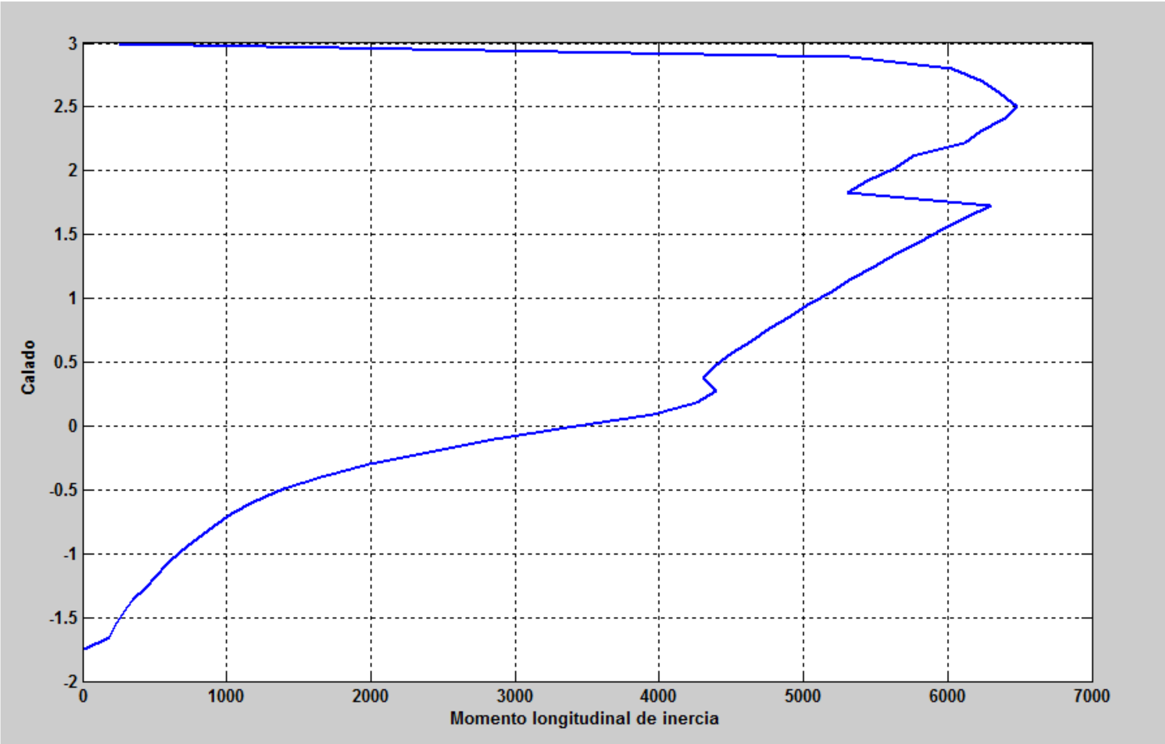
Centro de flotación



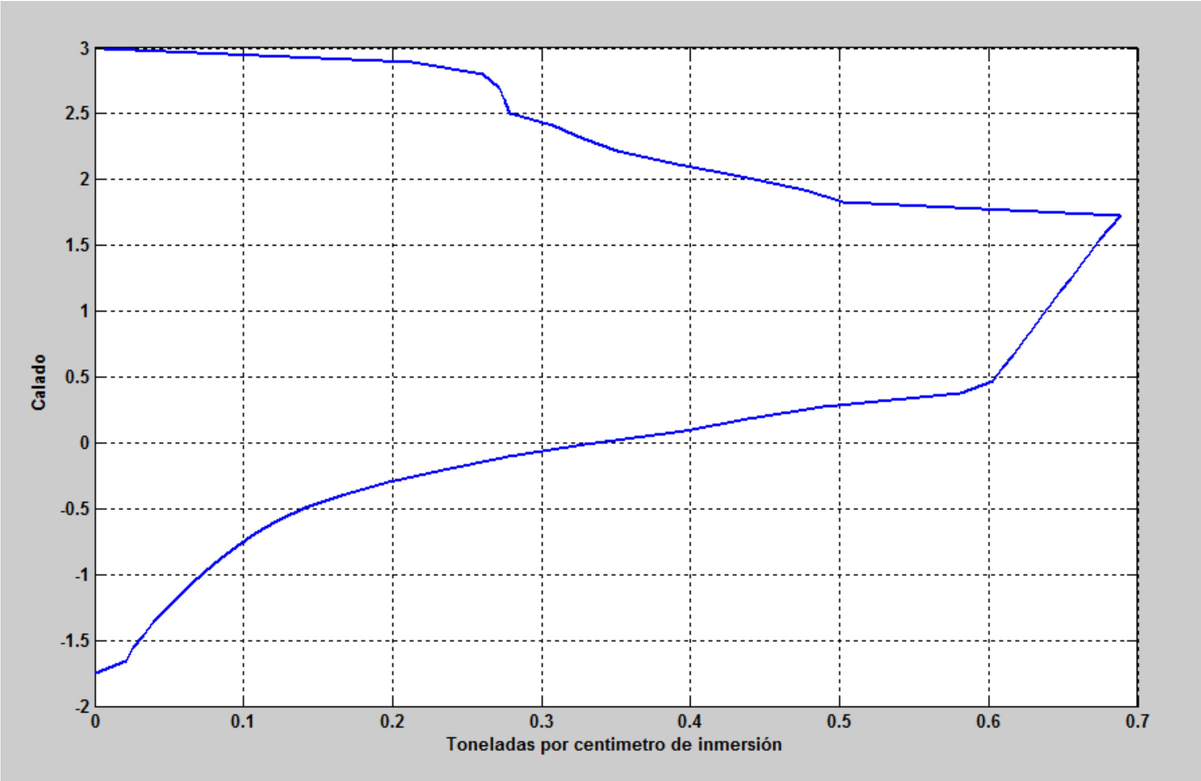
Momento de inercia transversal



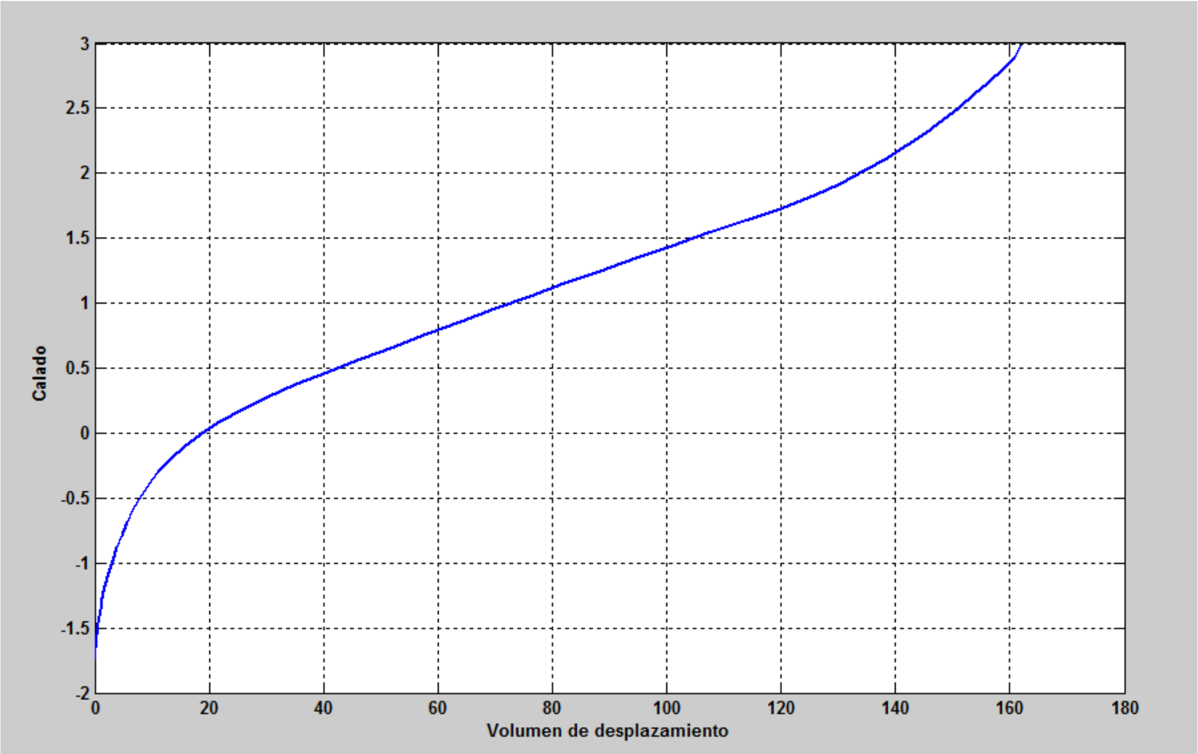
Momento de inercia de flotación



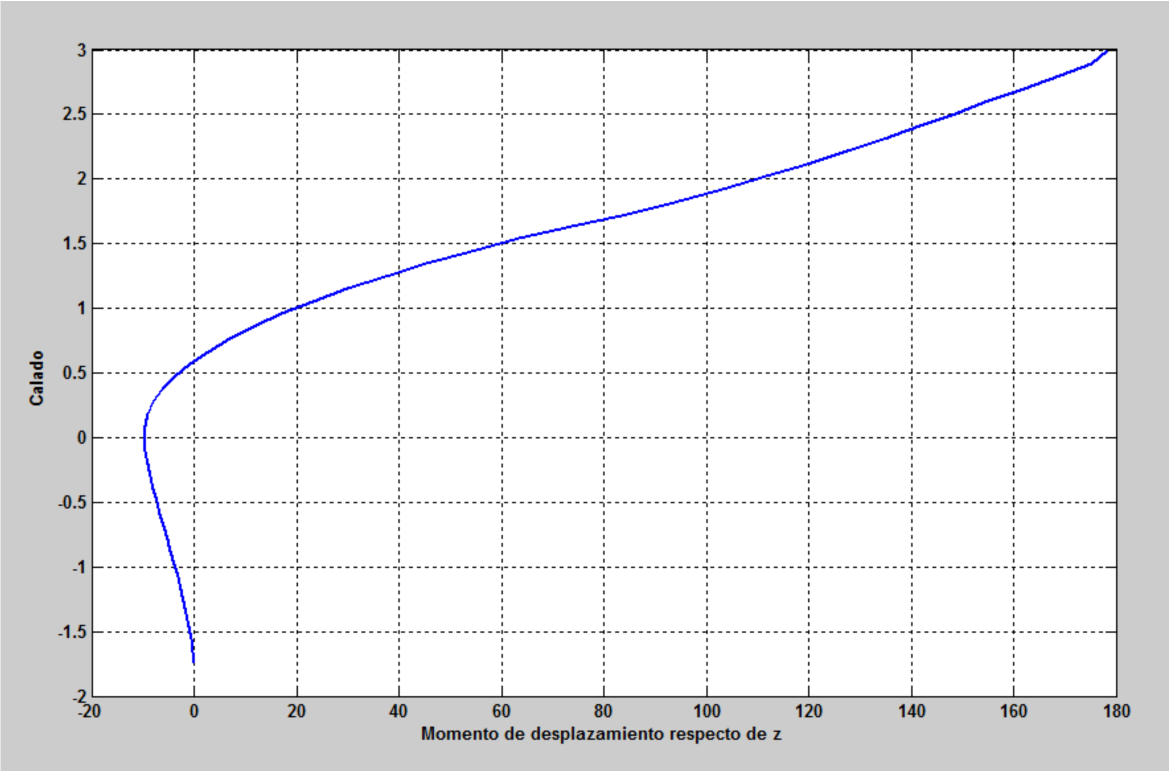
Momento de inercia longitudinal



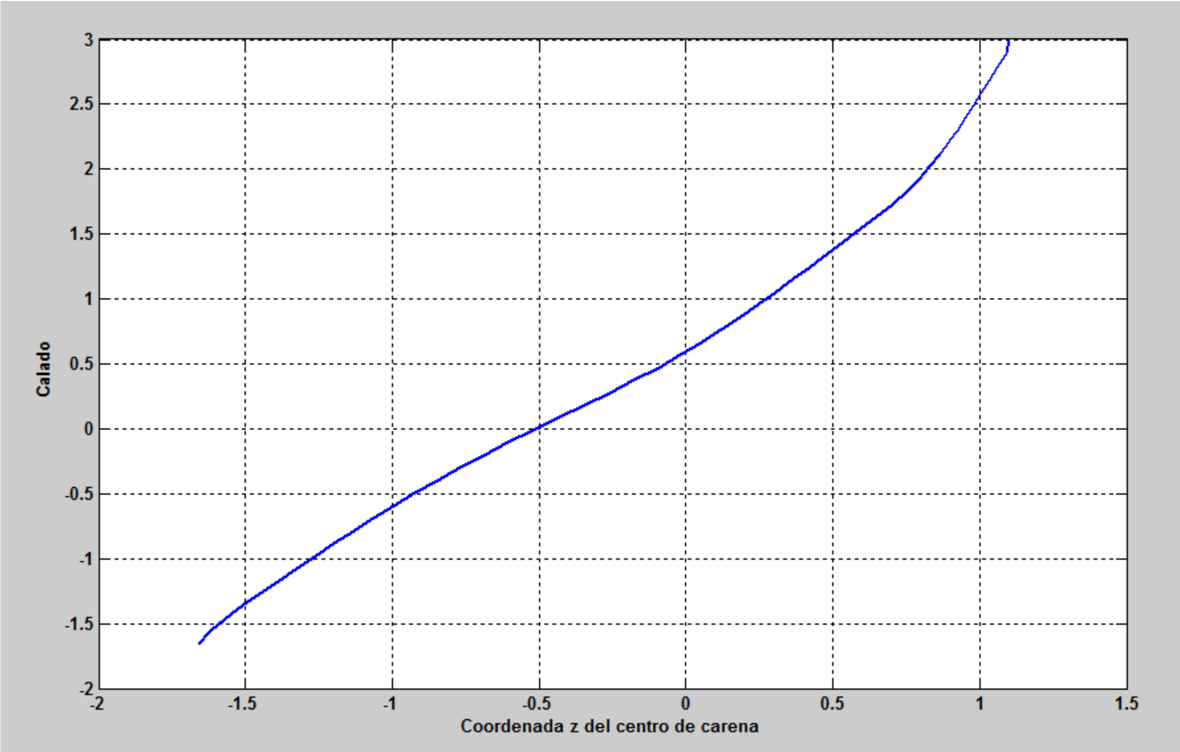
Toneladas por centímetro de inmersión



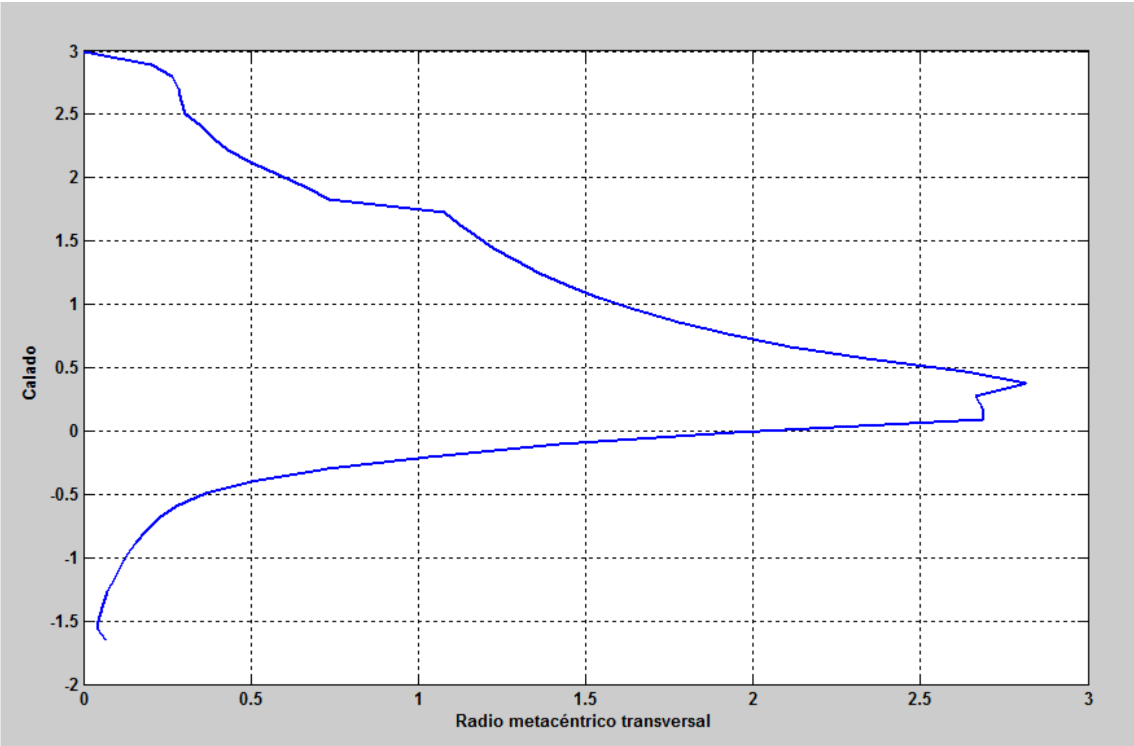
Volumen de carena



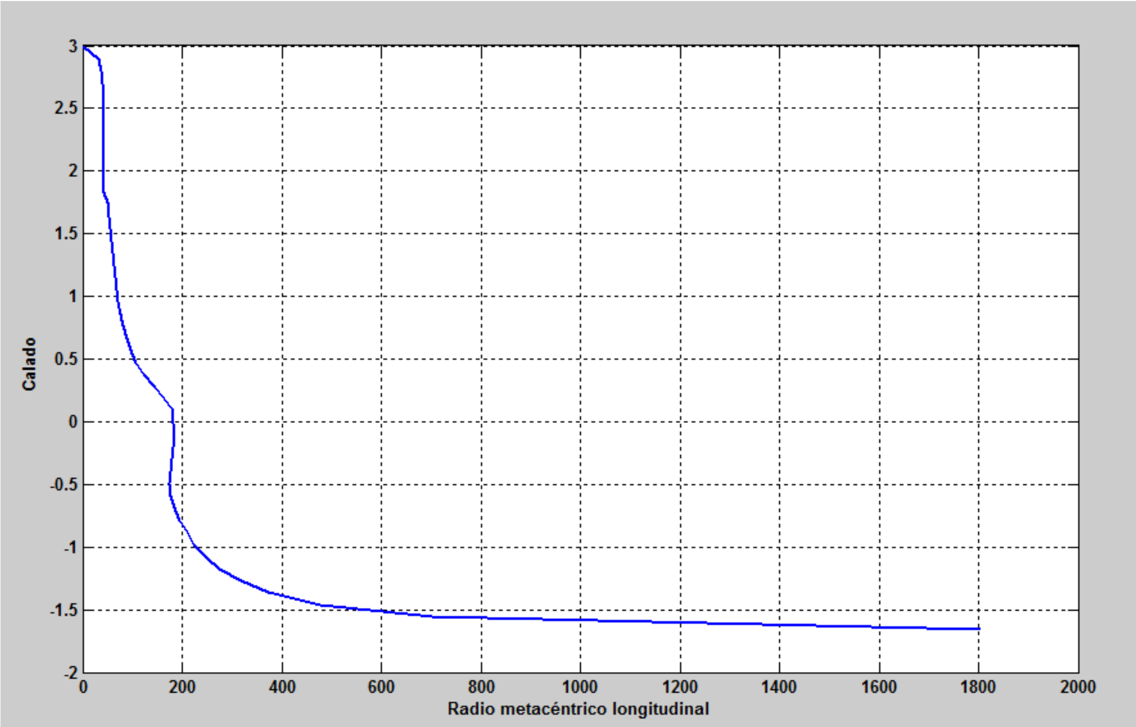
Momento de desplazamiento respecto de z



Posición vertical del centro de carena



Radio metacéntrico transversal



Radio metacéntrico longitudinal

9- Conclusiones

En el presente Proyecto Fin de Carrera se ha realizado un estudio de estabilidad para un buque pesquero, en su condición de adrizado, con el fin de poder saber datos importantes del barco, utilizando los resultados gráficos en cualquier momento que deseemos, es decir, entrando con un calado T en dichas gráficas, podremos saber cada uno de los datos que hemos calculado, sin necesidad de realizar ya ningún cálculo.

También el objetivo era realizar estos cálculos de manera automática y que pudieran servirnos para cualquier otro barco deseado, por lo que el programa realizado se ha generalizado para que así sea. Pues obteniendo un único archivo de Rhinoceros con las formas de nuestro barco, obtendremos todo un estudio de estabilidad, eso sí, en su condición de adrizado.

Se ha diseñado una Interfaz Gráfica con la que se consigue una mayor simplicidad a la hora de trabajar con dicho programa, es totalmente simple e intuitiva.

10- Bibliografía

[1] J.A. Alaez. *Teoría del buque*. ETSIN-UPM, 1980.

[2] S. Amat y S. Busquier. *Métodos Numéricos*. ETSIA-UPCT, 2005.

[3] A. Biran. *Ship hydrostatics and stability*. Butterworth-Heinemann, 2003.

[4] O. Palomo. Hidrostática y estabilidad. Apuntes 2010-2011, ETSINO-UPCT.

[5] Ayuda de Rhinoceros 4.0.

[6] Ayuda de MATLAB R2012a.

11- ANEXO I

DECLARACIÓN DE CONFORMIDAD

Capitanía Marítima en Almería · Ministerio de Fomento · Inspección Marítima

Los abajo firmantes;

ARMADOR : Antonio Fernández Ayala, D.N.I. nº 23.801.103-J

PATRON : Antonio Fernández Ayala

I. Naval que realiza el libro de estabilidad : D. Juan Dominguez Caparros.

declaran su conformidad a este cuaderno de estabilidad, del pesquero de nueva construcción, MACHIN.

Dicha conformidad se refiere a las condiciones de carga en cuanto al peso y distribución de los efectos de pesca, pertrechos, hielo en bodega, cajas de pescado, víveres, equipos, combustible y carga en bodega, así como a las instrucciones generales y particulares contenidas en el apartado del libro de estabilidad " Instrucciones al patrón"

En Adra, a 22 de Agosto de 2006

Armador	**Patrón**	**I. N. que realiza el libro de estabilidad**
		INGENIERO NAVAL Juan Domínguez Caparrós Cº n.º 1.194

INDICE

1.- INFORMACION GENERAL DEL BUQUE 2

2.- EXPERIENCIA DE ESTABILIDAD 3

- 2.1. a) - Determinación del coef. de la cna. maestra.
- 2.1. b) - Determinación del coef. de la flotación.
- 2.1. c) - Determinación del coef. de bloque.
- 2.2.- Determinación del peso escorante.
- 2.3.- Criterio de estabilidad sin personas a bordo.
- 2.4.- Cálculos suplementarios.

3.- INSTRUCCIONES AL PATRON 5

4.- PLANOS 6

5.- ANEXO 7

- Tabla III, valores C_1 en función de Cw

Capitanía Marítima en Almería · Ministerio de Fomento · Inspección Marítima

INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

itp	INNOVACIONES TECNOLOGICAS PESQUERAS ✱ MADRID ✱	Buque **MACHIN** Hoja nº **1**

1.- INFORMACIÓN GENERAL DEL BUQUE

NOMBRE DEL BUQUE: **MACHIN**

CONSTRUCTOR : **NICOLAS CASAS, S.L. ADRA (Almería)**

MATERIAL DE CONSTRUCCION: **P.R.F.V.**

MODALIDAD : **CERCO**

ESLORA TOTAL ....14,99 m

ESLORA (L) (80% Eslora Total)....................................11,99 m

MANGA FUERA FORROS...5,00 m

PUNTAL DE TRAZADO ....0,71 m

PUNTAL C/B QUILLA a cubta. parte elevada ....2,86 m

ALTURA QUILLA ....0,20 m

ASIENTO DE PROYECTO...0,00 m

ESPESOR DE LA CUBTA...14,00 mm

ESPESOR DEL FORRO ORDINARIO ....15,55 mm

Capitanía Marítima en Almería · Ministerio de Fomento · Inspección Marítima

INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

itp	INNOVACIONES TECNOLOGICAS PESQUERAS ✱ MADRID ✱	Buque **MACHIN** Hoja nº 2

2.- EXPERIENCIA DE ESTABILIDAD

LUGAR DE LA EXPERIENCIA : **ADRA**
FECHA DE LA EXPERIENCIA: **16-ago-06**

Situacion de la embarcación : Máximo desplazamiento

Calado en marcas de popa	**2,70 m**
Calado en marcas de proa	**1,98 m**

Eslora en la flotación; LBP	**13,19 m**
Manga flot; B	**4,96 m**
Puntal de cons.; D	**2,66 m**
Calado medio(ref. cb/q); d	**2,34 m**
Francobordo; f	**0,52 m**
Brusca bao maestro; c	**0,15 m**
Registro Bruto; R.B.	**19,87 tons. Moorsom**

Año y lugar de construccion : **2006-ADRA**

PRUEBA DE ESTABILIDAD
Entidad que realizó la experiencia: **INN. TEC. PESQUERAS S.L. (Madrid)**

2.1. a) Determinacion de coef. de la maestra; Cm

b1 (m) =	**0,76**	
b2 (m) =	**1,64**	
b3 (m) =	**2,02**	
b4 (m) =	**0,09**	
Am = Area de la maestra	**5,06**	m^2
Cm =	**0,44**	

Capitanía Marítima en Almería
Ministerio de Fomento
Inspección Marítima

2.1. b) Determinacion del coef. de la flotación; Cw

B1 (m) =	**4,20**	
B (m) =	**4,96**	
B3 (m) =	**4,91**	
Aw = Area de la flotación	**50,96**	m^2
Cw =	**0,78**	

2.1. c) Determinacion del coef. de bloque; Cb = 0,86*Cm*Cw

Δ = Desplazamiento = 1,026 * Cb*LBP*B*d

Cb = **0,29**

Δ = **45,85 t**

2.2. Determinacion del peso escorante; Δh = 0,065*Aw

Δh = **3,31 t**

INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

INNOVACIONES TECNOLOGICAS PESQUERAS ✱ MADRID ✱		Buque **MACHIN** Hoja nº **3**

2.3. Criterio de estabilidad sin personas a bordo.

φ = escora media en grados **4,69 º**

$\Delta\varphi$ = lastre solido colocado para obtener una escora no superior a 14º,o para que no se sumerja la tapa de regala o el trancanil.

$GH = (B/3) * (\Delta h/\Delta) * \text{cotang } \varphi =$ **1,46 m**

$GZ = GH * \text{sen } \varphi =$ **0,12 m**

2,4 Cálculos suplementarios

a) Determinación del C. de Carena respecto ALEFRIZ = KB_o

$KB_o = d*(0{,}833-0{,}287 * Cm) =$ **1,65 m**

b) Determinación de la altura met. Transversal por encima del KB_o; B_oM_{ot}

C_1 = Cof. Inercia tabla III: **0,05264**

i = iner. flot. $= C_1 * LBP * B^3 =$ **84,72 m⁴**

$B_oM_{ot} = i * 1{,}026 / \Delta =$ **1,90 m**

c) Determinación de la altura del C.D.G., sobre ALEFRIZ (KG), a plena carga sin pasajeros.

W_i = pesos utilizados en la exp.; $\Delta h/9 =$ **0,37 t**

l = Distancia horizontal entre pesos: **4,44 m**

φi = escora al trasladar uno de los pesos: **1,41 º**

GM_{ot} = r-a = altura metacentrica = $(W_i * l)/\Delta * \tan \varphi i$

$GM_{ot} =$ **1,46 m**

$KG = KB_o + B_oM_{ot} - GM_{ot} =$ **2,09 m**

Capitanía Marítima en Almería · Ministerio de Fomento · Inspección Marítima

INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

INNOVACIONES TECNOLOGICAS PESQUERAS
MADRID

Buque **MACHIN**

Hoja nº 4

3.- INSTRUCCIONES AL PATRÓN

- Se tendrá especial cuidado en mantener cerradas las puertas de acceso a espacios cerrados de habilitación desde la cubierta principal, cuando no sea en el momento de su utilización para acceso a dicho compartimento.
- Las cajas de pescado en bodega, una vez estibadas, habrán de sujetarse convenientemente de forma que se haga imposible su corrimiento.
- No se autoriza la navegación con carga sobre cubierta.
- El patrón deberá cuidar que no sumerja ningún punto de la cubierta por efecto de las fuerzas ejercidas por el sistema de pesca.
- La información sobre las características de estabilidad del buque deberá estar disponible a bordo, ser accesible para el personal de guardia y respetarse estrictamente.
- El Patrón deberá adoptar las medidas de precaución necesarias para el mantenimiento de la estabilidad del buque.

COLEGIO OFICIAL DE INGENIEROS NAVALES Y OCEANICOS
VISADO N.º 119402570 0
EXPEDIDO EL 29 AGO 2006
Por el Colegio,

Capitanía Marítima en Almería · Ministerio de Fomento · Inspección Marítima

INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

INNOVACIONES TECNOLOGICAS PESQUERAS
MADRID

Buque MACHIN

Hoja nº 5

4.- PLANOS

PLANO Nº	DESIGNACION
E-06-1301	- PLANO DE FORMAS CALADOS EN MARCAS Y PERPENDICULARES
E-06-1302	- DISPOSICION GENERAL ALZADO Y PLANTA

Capitanía Marítima en Almería
Ministerio de Fomento
Inspección Marítima

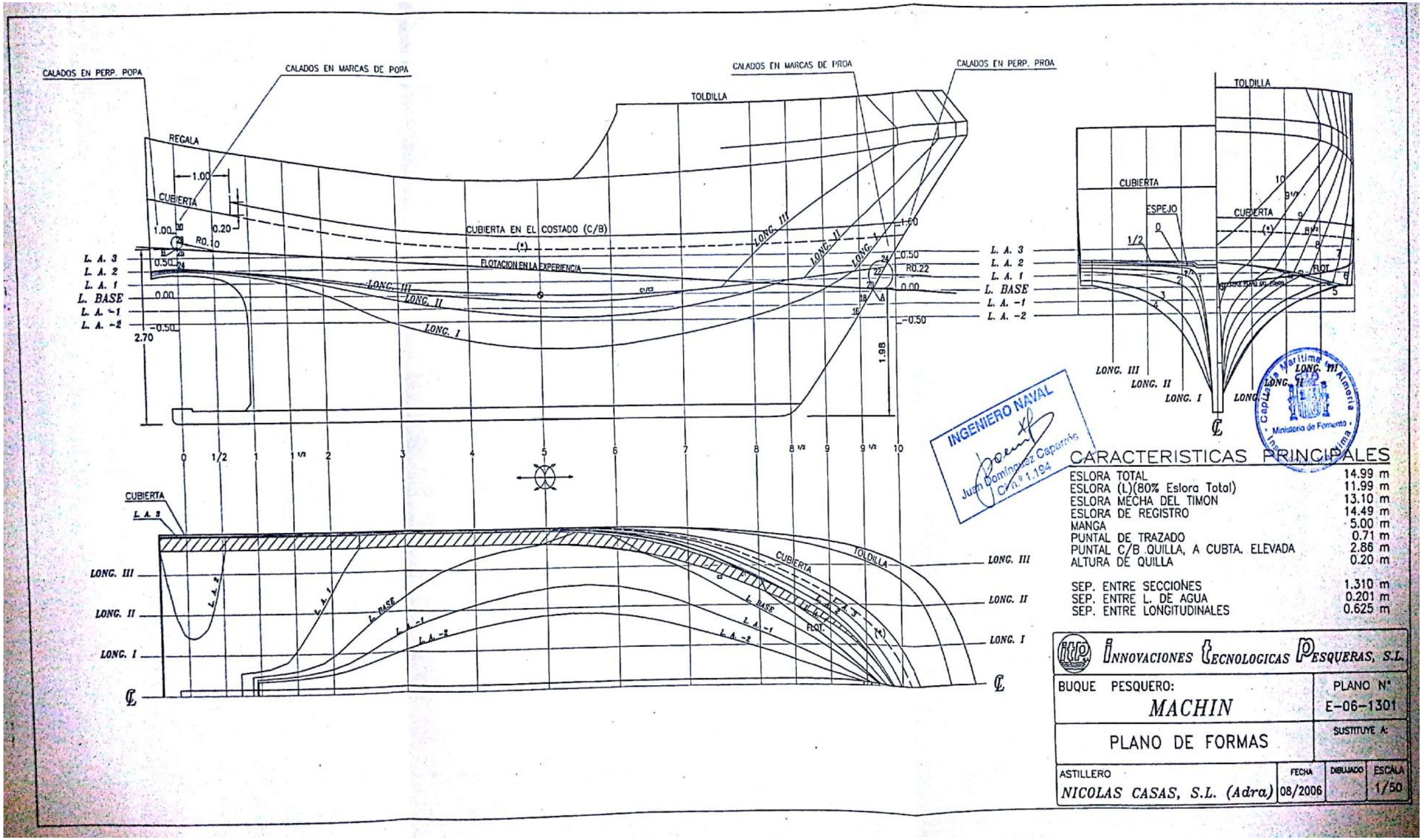
INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

itp

INNOVACIONES
TECNOLOGICAS
PESQUERAS
MADRID

Buque
MACHIN

Hoja nº 6


CALADOS EN PERP. POPA
CALADOS EN MARCAS DE POPA
CALADOS EN MARCAS DE PROA
CALADOS EN PERP. PROA
TOLDILLA
REGALA
CUBIERTA
CUBIERTA EN EL COSTADO (C/B)
FLOTACION EN LA EXPERIENCIA
L. A. 3
L. A. 2
L. A. 1
L. BASE
L. A. -1
L. A. -2
LONG. III
LONG. II
LONG. I
ESPEJO
FLOT.
CARACTERISTICAS PRINCIPALES
ESLORA TOTAL 14.99 m
ESLORA (L)(80% Eslora Total) 11.99 m
ESLORA MECHA DEL TIMON 13.10 m
ESLORA DE REGISTRO 14.49 m
MANGA 5.00 m
PUNTAL DE TRAZADO 0.71 m
PUNTAL C/B QUILLA, A CUBTA. ELEVADA 2.86 m
ALTURA DE QUILLA 0.20 m
SEP. ENTRE SECCIONES 1.310 m
SEP. ENTRE L. DE AGUA 0.201 m
SEP. ENTRE LONGITUDINALES 0.625 m
INGENIERO NAVAL
Juan Domínguez Capurrós
Cº Nº 1.194
Capitanía Marítima Almería
Ministerio de Fomento
INNOVACIONES TECNOLOGICAS PESQUERAS, S.L.
BUQUE PESQUERO:
MACHIN
PLANO Nº
E-06-1301
SUSTITUYE A:
PLANO DE FORMAS
ASTILLERO
NICOLAS CASAS, S.L. (Adra)
FECHA
08/2006
DIBUJADO
ESCALA
1/50

Capitanía Marítima en Almería · Ministerio de Fomento · Inspección Marítima

Tabla III, valores de C_1 en funcion de Cw

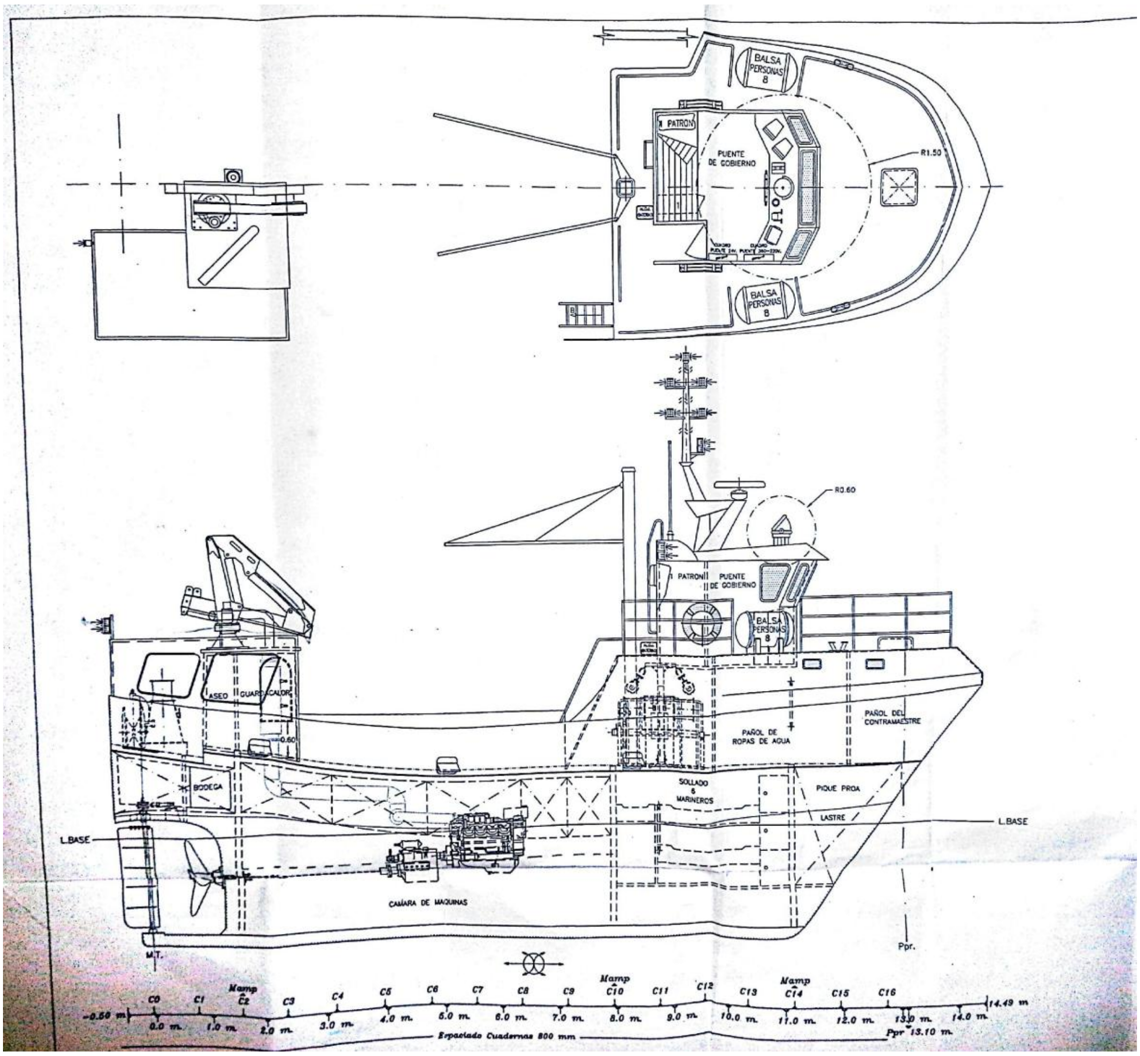
Cw	C_1
0,50	0,02250
0,51	0,02316
0,52	0,02383
0,53	0,02466
0,54	0,02540
0,55	0,02633
0,56	0,02710
0,57	0,02800
0,58	0,02910
0,59	0,03000
0,60	0,03100
0,61	0,03200
0,62	0,03300
0,63	0,03400
0,64	0,03500
0,65	0,03600
0,66	0,03733
0,67	0,03844
0,68	0,03955
0,69	0,04100
0,70	0,04200
0,71	0,04325
0,72	0,04500
0,73	0,04600
0,74	0,04700
0,75	0,04841
0,76	0,04982
0,77	0,05123
0,78	0,05264
0,79	0,05405
0,80	0,05546

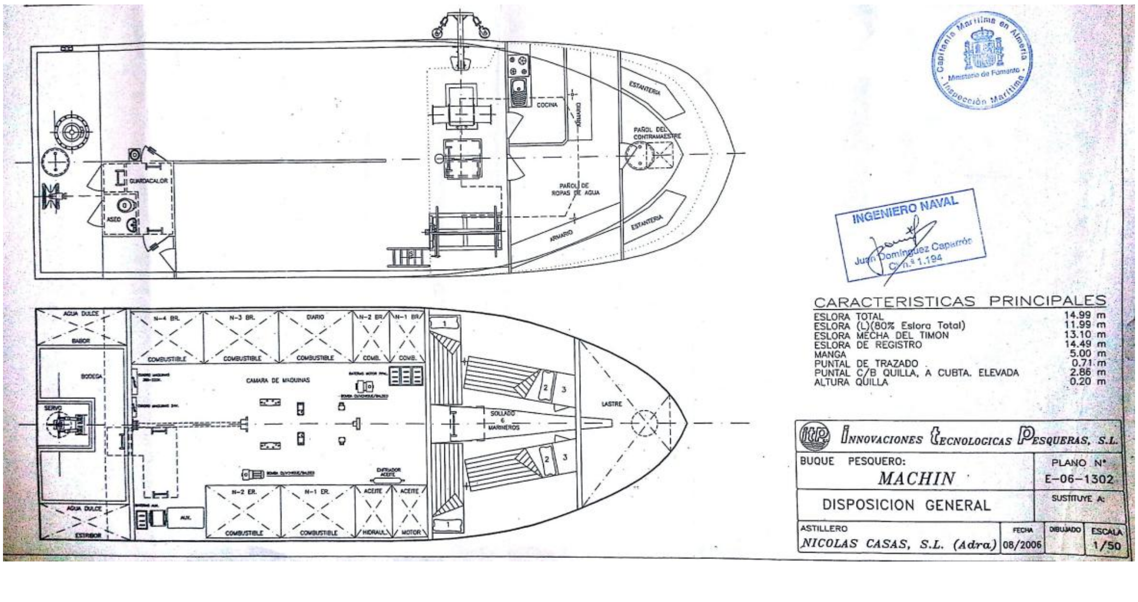
INGENIERO NAVAL
Juan Domínguez Caparrós
Cº n.º 1.194

itp INNOVACIONES TECNOLOGICAS PESQUERAS MADRID

Buque MACHIN

Hoja nº 8


BALSA PERSONAS 8
PATRON
PUENTE DE GOBIERNO
R1.50
BALSA PERSONAS 8
R0.60
PATRON
PUENTE DE GOBIERNO
ASEO
GUARDACALOR
PAÑOL DEL CONTRAMAESTRE
PAÑOL DE ROPAS DE AGUA
BODEGA
SOLLADO 6 MARINEROS
PIQUE PROA
LASTRE
L.BASE
L.BASE
CAMARA DE MAQUINAS
M.T.
Ppr.
Mamp
Mamp
Mamp
Espaciado Cuadernas 800 mm
Ppr 13.10 m
14.49 m


COCINA
ESTANTERIA
PAÑOL DEL CONTRAMAESTRE
PAÑOL DE ROPAS DE AGUA
ARMARIO
ESTANTERIA
GUARDACALOR
ASEO
AGUA DULCE
BABOR
BODEGA
SERVO
CAMARA DE MAQUINAS
COMBUSTIBLE
COMBUSTIBLE
COMBUSTIBLE
LASTRE
SOLLADO 6 MARINEROS
AGUA DULCE
ESTRIBOR
COMBUSTIBLE
COMBUSTIBLE
HIDRAUL.
MOTOR
INGENIERO NAVAL
Juan Domínguez Caparrós
C. n.º 1.194
CARACTERISTICAS PRINCIPALES
ESLORA TOTAL 14.99 m
ESLORA (L)(80% Eslora Total) 11.99 m
ESLORA MECHA DEL TIMON 13.10 m
ESLORA DE REGISTRO 14.49 m
MANGA 5.00 m
PUNTAL DE TRAZADO 0.71 m
PUNTAL C/B QUILLA, A CUBTA. ELEVADA 2.86 m
ALTURA QUILLA 0.20 m
INNOVACIONES TECNOLOGICAS PESQUERAS, S.L.
BUQUE PESQUERO:
MACHIN
PLANO Nº
E-06-1302
DISPOSICION GENERAL
SUSTITUYE A:
ASTILLERO
NICOLAS CASAS, S.L. (Adra)
FECHA
08/2006
DIBUJADO
ESCALA
1/50