

# Proyecto y construcción plataformas y artefactos Práctica nº3

**Realizado por:** Antonio Justo González

**DNI =** 74741742-E

**Asignatura:** Proyecto y construcción de plataformas y artefactos

**Profesor:** José Enrique Gutiérrez Romero

**Práctica nº:** 3



## INDICE

<b>INDICE.....</b>	<b>2</b>
<b>Índice de figuras .....</b>	<b>3</b>
<b>1º Cálculo de las fuerzas cuando la unidad soporta un oleaje armónico con un periodo de 10 segundos y una altura de ola de 8 m .....</b>	<b>6</b>
<b>2º Cálculo de las fuerzas cuando la unidad está sometido a un espectro Pierson – Moskowitz con un periodo de pico <math>T_p</math> de 12 segundos y una altura significativa de ola <math>H_s</math> de 4 m. ....</b>	<b>17</b>
<b>3º Programación previa para comparar los resultados de fuerzas globales del apartado “a” mediante cálculo analítico con la solución obtenida usando SeaFEM, mediante Matlab. ....</b>	<b>20</b>
3.1º Fuerzas en los pilares y en los refuerzos esbeltos teoría .....	20
3.2º Calculo iterativo para hallar la longitud de onda.....	21
3.3º Lectura de datos del problema.....	23
3.4º Programación para el cálculo de la fuerzas de Morrison. ....	24
3.5º Programación del cálculo de las fuerzas sobre el ponton/soft tank .....	31
3.6º Programación del cálculo de las fuerzas sobre las planchas de arfada.....	40
3.7º Programación del cálculo de las fuerzas sobre el Hard Tank o cilindro .....	44
<b>4º Comparación de los resultados obtenidos entre .....</b>	<b>48</b>

## Índice de figuras

INDICE.....	2
-------------	---

### 1º Cálculo de las fuerzas cuando la unidad soporta un oleaje armónico con un periodo de 10 segundos y una altura de ola de 8 m .....

Figura 1.1 Opción que nos permitirá “unir” los puntos iniciales y finales de la línea.	6
Figura 1.2 Puntos a utilizar para generar las líneas, y puntos medios tras eliminar las líneas. ....	7
Figura 1.3 Orientación tomada. Popa-Proa, eje y. Babor-Estribor, eje x. ....	7
Figura 1.4 Dirección de propagación de la ola, eje x. ....	8
Figura 1.5 Coordenadas de los puntos iniciales y finales de los pilares además de sus diámetros.....	8
Figura 1.6 Identificación de ejemplo para los pilares de popa .....	9
Figura 1.7 Significado de up/down.....	9
Figura 1.8 Coordenadas iniciales para los 3 grupos de refuerzos.....	10
Figura 1.9 Términos junto a su definición del apartado de definición del problema. Pág. 8-9 del manual. ....	11
Figura 1.10 Tamaño del mallado .....	12
Figura 1.11 Casillas correspondientes a los períodos de tiempo .....	12
Figura 1.12 Calculo de las fuerzas según Seafem .....	13
Figura 1.13 Valores de los coeficientes de masa añadidas para los refuerzos y los pilares de sección circular.....	14
Figura 1.14 Valor del coeficiente de arrastre para una sección circular.....	14
Figura 1.15 Time step de 0.247 s, que fue tomado igual que el output step, tras mallar y resolver la primera vez.....	15
Figura 1.16 Estructura ya mallada, según los tamaños dados en la Figura 1.10. ....	15
Figura 1.17 Introducción de los datos del enunciado .....	16
Figura 1.18 Fuerzas totales en las 3 direcciones x y z para una ola monocromática de A=4 m y T=10 s.....	16
Figura 1.19 Fuerzas totales en las 3 direcciones y z para una ola monocromática de A=4 m y T=10 .....	16

### 2º Cálculo de las fuerzas cuando la unidad está sometido a un espectro Pierson – Moskowitz con un periodo de pico $T_p$ de 12 segundos y una altura significativa de ola $H_s$ de 4 m. ....

Figura 2.1 Definición del espectro P-M para la resolución del segundo apartado ....	17
Figura 2.2 Términos que caracterizarán el espectro de JONSWAP. Pág. 15 del manual.....	18

Figura 2.3 Fuerzas x,y,z para un espectro P-M de 20 componentes para $H_s=4$ m y $T_p=12$ s .....	19
Figura 2.4 Fuerzas y,z para un espectro P-M de 20 componentes para $H_s=4$ m y $T_p=12$ .....	19

### **3º Programación previa para comparar los resultados de fuerzas globales del apartado “a” mediante cálculo analítico con la solución obtenida usando SeaFEM, mediante Matlab. .... 20**

Figura 3.1 Formulas de Morrison para cilindros esbeltos .....	20
Figura 3.2 Triple producto vectorial, resultado de multiplicar $\mathbf{a} \times \mathbf{b} \times \mathbf{a}$ da un vector “d” normal a “a”.....	21
Figura 3.3 Archivo de datos de entrada necesario para los cálculos .....	23
Figura 3.4 Coordenadas de los pilares. ....	24
Figura 3.5 Coordenada de los refuerzos .....	25
Figura 3.7 Explicación de cómo están organizados los puntos y como se deben introducir para su correcta lectura .....	25
Figura 3.8 Fórmulas con las que se hallarán las velocidades y las aceleraciones. Extraídas del DNV-RP-205 .....	28
Figura 3.9 Aguas profundas, Aguas limitadas, y aguas transitorias entre ambos límites .....	28
Figura 3.10 Determinamos la fuerza dominante.....	31
Figura 3.11 Expresiones del potencial de velocidad extraídas del DNV-cp-205.....	32
Figura 3.12 Coordenadas de los vértices de las 4 planchas verticales del pontón teniendo en cuenta la orientación dada en la Figura 1.6 y Figura 1.7 .....	33
Figura 3.13 Breve explicación de cómo se está realizando la integración .....	35
Figura 3.14 Vértices de las planchas superiores e inferiores del pontón.....	36
Figura 3.15 Esquema de cómo se deben introducir las coordenadas y los valores lógicos.....	37
Figura 3.16 Coordenadas de los vértices de las planchas verticales. Primera plancha antiarfada .....	40
Figura 3.17 Coordenadas de los vértices de las planchas horizontales. Primera plancha antiarfada.....	41
Figura 3.18 Coordenadas de los vértices de las planchas verticales. Segunda plancha antiarfada .....	41
Figura 3.19 Coordenadas de los vértices de las planchas horizontales. Segunda plancha antiarfada.....	41
Figura 3.20 Coordenadas de los vértices de las planchas verticales. Tercera plancha antiarfada .....	42
Figura 3.21 Coordenadas de los vértices de las planchas horizontales. Tercera plancha antiarfada .....	42
Figura 3.22 Coordenadas de los vértices de las planchas verticales. Cuarta plancha antiarfada .....	43

Figura 3.23 Coordenadas de los vértices de las planchas horizontales. Cuarta plancha antiarfada .....	43
Figura 3.24 Comprobación de la teoría a aplicar para el hard tank. ....	44
Figura 3.25 Resumen de aplicabilidad de las distintas teorías según el efecto de la difracción y de $K_c$ .....	45
Figura 3.26 Formulación de la teoría de la difracción para un cilindro compacto. ....	46
Figura 3.27 Solución resultado de la integración .....	46
<b>4º Comparación de los resultados obtenidos entre .....</b>	<b>48</b>
Figura 4.1 Pequeña guía resumen de que contiene cada variable tras ejecutar el programa .....	48
Figura 4.2 Fuerzas x y Fuerzas z para los pilares y refuerzos .....	49
Figura 4.3 Fuerzas x y Fuerzas z para el Soft Tank .....	49
Figura 4.4 Fuerzas x y Fuerzas z para las planchas de arfada .....	50
Figura 4.5 Fuerzas x para el Hard Tank .....	50
Figura 4.6 Fuerzas en x (N/m) para $t=140s$ a varias profundidades z (m) .....	51
Figura 4.7 Fuerza total en x (N) actuante en la Spar .....	52
Figura 4.8 Fuerza total en z (N) actuante en la Spar .....	52
Figura 4.9 Fuerzas x, z, totales, las fuerzas en y son cero .....	53
Figura 4.10 Fuerzas x,y,z para una ola monocromática de $H=8$ m, $T=10s$ .....	54
Figura 4.11 Curva de fuerzas totales x,z obtenidas analíticamente para aun rango de tiempo 143-153 s .....	54
Figura 4.12 Fuerzas x e y obtenidas con el programa seaFEM .....	55

## 1º Cálculo de las fuerzas cuando la unidad soporta un oleaje armónico con un periodo de 10 segundos y una altura de ola de 8 m

Dado a que la escala a aplicar es 0.9, el modelo de partida hay que primero prepararlo. Los pasos a seguir son los dados el video guía, y que se resumen a continuación:

- Movemos el modelo para que esté completamente sumergido.
- Aplicamos el factor de escala de 0.9.
- Creamos el volumen de control, para ello creamos un cilindro interno, con el doble del diámetro que el cilindro de la Spar,  $2 \times 31.5 = 56.7$  m.
- La altura del cilindro será mayor que la Spar, se tomará la altura de 250 m.
- Eliminamos el volumen, la tapa superior y la inferior así como la tapa de la Spar.
- Creamos otro cilindro exterior de igual altura, y con un diámetro aproximadamente tres veces el diámetro interior, 150 m.
- Eliminamos el volumen, y ambas tapas.
- Creamos las tapas, 2 para las superficies libres, una que va desde el cilindro exterior al interior y otra para el cilindro interior. Y 2 para el fondo, de la misma manera que las superficies libres.
- Creamos 2 volúmenes, una para el volumen entre el cilindro interior y el exterior, y otro entre la Spar y el cilindro interior.
- Ahora, tras rellenar las distintas casillas, hay que proceder a crear los elementos slender, para lo que necesitaremos las coordenadas iniciales y las coordenadas finales de cada uno de los elementos. Para ello crearemos una línea en cada elemento utilizando para ello los puntos de sus líneas. Para que el programa nos coja el punto, haremos uso de la opción “join”.

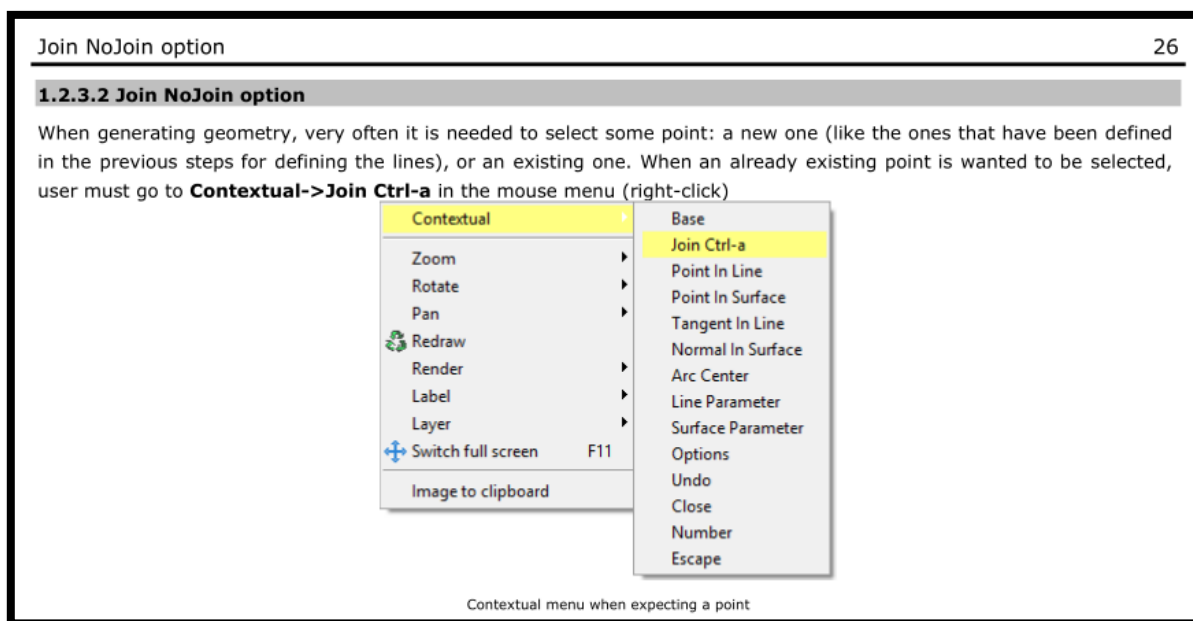


Figura 1.1 Opción que nos permitirá “unir” los puntos iniciales y finales de la línea.

Después procederemos a dividir en dos cada línea de forma que obtendremos el punto medio, y por último eliminaremos las líneas.

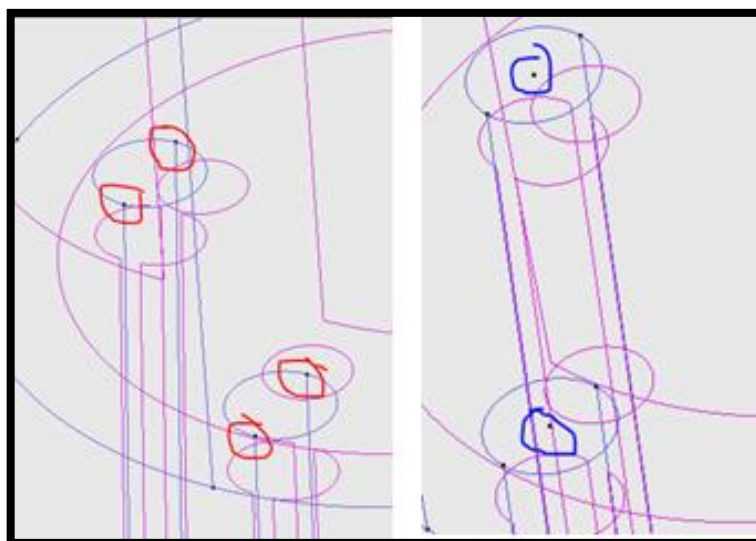


Figura 1.2 Puntos a utilizar para generar las líneas, y puntos medios tras eliminar las líneas.

- Tras obtener todos los puntos, vamos a crear los elementos slender en los que, necesitaremos alguna forma de nombrarlos y diferenciarlos uno de otro. Para los pilares y brazing se seguirá la orientación mostrada en la Figura 1.3

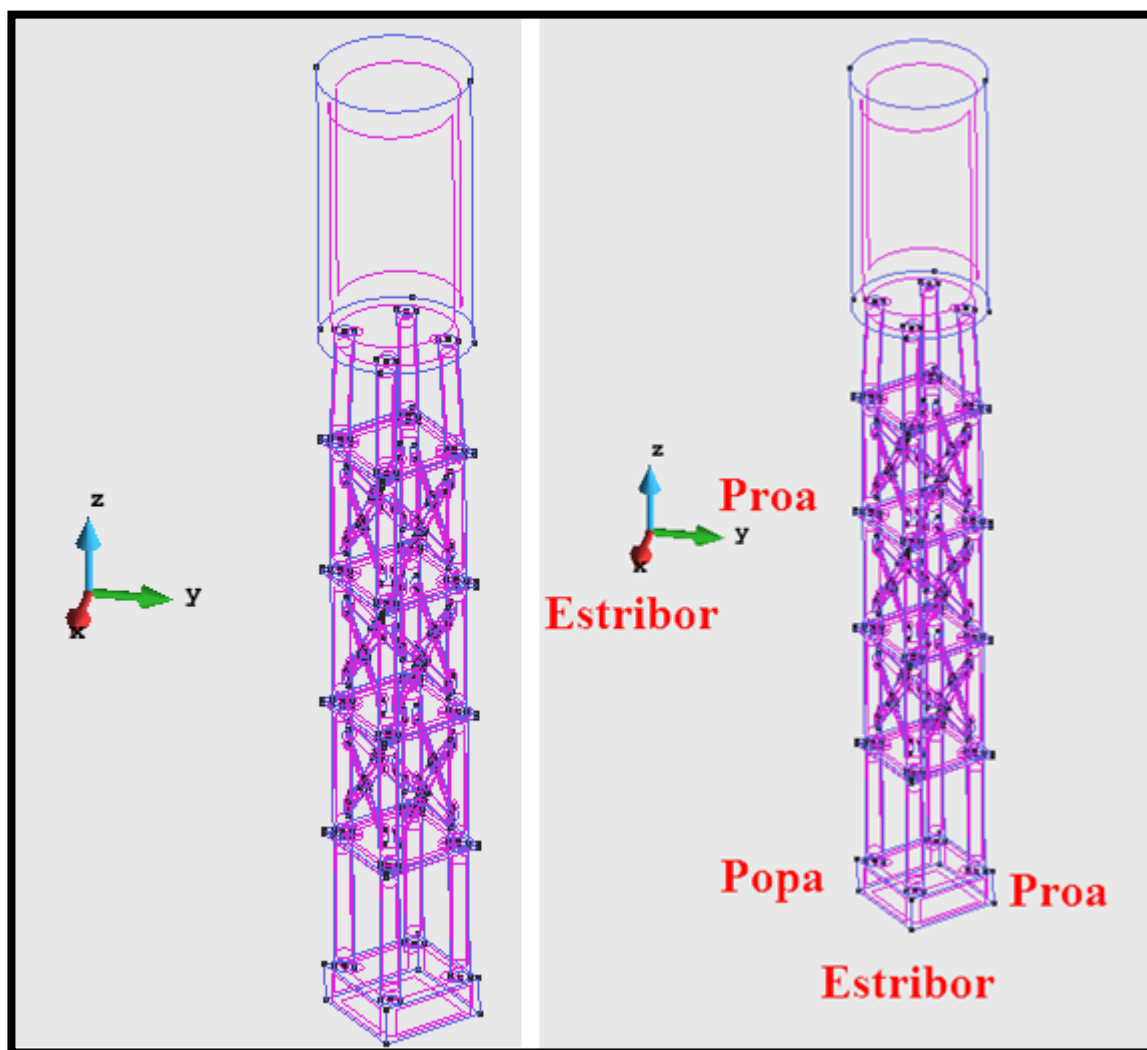


Figura 1.3 Orientación tomada. Popa-Proa, eje y. Babor-Estribor, eje x.

Tal y como se ve, la Figura 1.4, la ola se propagará de en la dirección x, que correspondería para nuestro sistema de coordenadas de babor a estribor, y ya que la estructura es simétrica, es lo mismo que si se propagase en la dirección y.

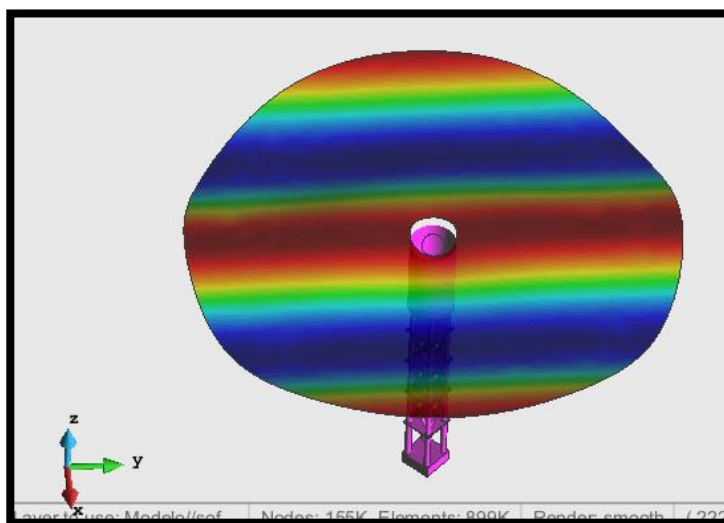


Figura 1.4 Dirección de propagación de la ola, eje x.

Por lo tanto teniendo en cuenta la orientación dada en la Figura 1.3, se tienen las siguientes coordenadas para crear los elementos slender.

	Nº	Punto inicial (abajo)			Punto final (arriba)			D (m)
		x	y	z	x	y	z	
pilar 1 popa	1	0	-10.6312	-76.5	0	-9.45	-54	3.6
pilar 1 proa	2	0	10.6312	-76.5	0	9.45	-54	3.6
pilar 1 babor	3	-10.6312	0	-76.5	-9.45	0	-54	3.6
pilar 1 estribor	4	10.6312	0	-76.5	9.45	0	-54	3.6
pilar 2 popa	5	0	-10.6312	-103.5	0	-10.6312	-77.4	3.6
pilar 2 proa	6	0	10.6312	-103.5	0	10.6312	-77.4	3.6
pilar 2 babor	7	-10.6312	0	-103.5	-10.6312	0	-77.4	3.6
pilar 2 estribor	8	10.6312	0	-103.5	10.6312	0	-77.4	3.6
pilar 3 popa	9	0	-10.6312	-130.5	0	-10.6312	-104.4	3.6
pilar 3 proa	10	0	10.6312	-130.5	0	10.6312	-104.4	3.6
pilar 3 babor	11	-10.6312	0	-130.5	-10.6312	0	-104.4	3.6
pilar 3 estribor	12	10.6312	0	-130.5	10.6312	0	-104.4	3.6
pilar 4 popa	13	0	-10.6312	-157.5	0	-10.6312	-131.4	3.6
pilar 4 proa	14	0	10.6312	-157.5	0	10.6312	-131.4	3.6
pilar 4 babor	15	-10.6312	0	-157.5	-10.6312	0	-131.4	3.6
pilar 4 estribor	16	10.6312	0	-157.5	10.6312	0	-131.4	3.6
pilar 5 popa	17	0	-10.6312	-185.4	0	-10.6312	-158.4	3.6
pilar 5 proa	18	0	10.6312	-185.4	0	10.6312	-158.4	3.6
pilar 5 babor	19	-10.6312	0	-185.4	-10.6312	0	-158.4	3.6
pilar 5 estribor	20	10.6312	0	-185.4	10.6312	0	-158.4	3.6

Figura 1.5 Coordenadas de los puntos iniciales y finales de los pilares además de sus diámetros.

Hay que mencionar que el punto inicial corresponderá siempre al elemento más bajo de los dos en la dirección z, y el punto final el punto más alto.



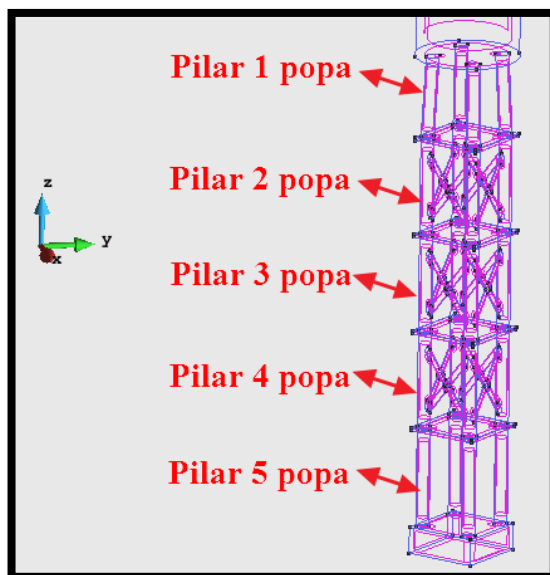


Figura 1.6 Identificación de ejemplo para los pilares de popa

Si obtenemos ahora los puntos iniciales y finales de los refuerzos de la misma forma que con los pilares obtendremos las coordenadas mostradas en la Figura 1.8, aunque para entender la nomenclatura se requiere ver los brazing localmente. Es decir, R1\_pp\_estribor\_up, hace referencia al primer grupo de 8 refuerzos, al subgrupo de 4 que se encuentra a popa, teniendo en cuenta el sistema explicado en la Figura 1.3 y que se ve en la Figura 1.6, al subgrupo de 2 que se encuentra a estribor (mirar Figura 1.3) y ahora para entender el up/down, Mirando la Figura 1.7.

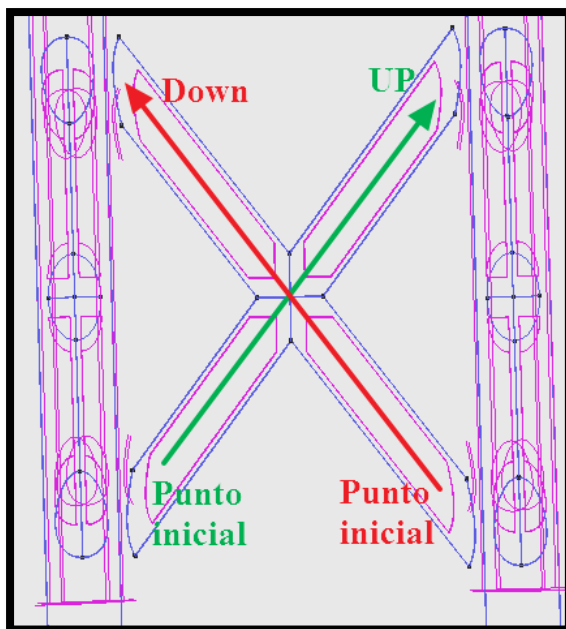


Figura 1.7 Significado de up/down

Aunque, para elegir correctamente el up/down, hay siempre que ver al refuerzo localmente, es decir indistintamente de su posición global, si nos situamos en frente de la cruz del refuerzo, el refuerzo up es el que crece hacia la derecha y el down el que crece hacia la izquierda.

Ahora para entender cuál será el punto inicial y cuál será el punto final, mirando a la cruz del refuerzo localmente, ver Figura 1.7, el punto inicial será para el refuerzo up, el que se encuentra más a la derecha de los dos, y para los refuerzos down, el punto inicial será el que se encuentra más a la izquierda de los dos.

Entonces tenemos que las coordenadas de los puntos iniciales para los refuerzos son:

	Nº	Punto inicial (abajo)			Punto final (arriba)			D (m)
		x	y	z	x	y	z	
R1_pp_estribor_up	21	1.27279	-9.35846	-97.9074	9.35846	-1.27279	-83.1051	1.8
R1_pp_estribor_down	22	9.35846	-1.27279	-98.5871	1.27279	-9.35846	-83.1003	1.8
R1_pp_babor_up	23	-9.35846	-1.27279	-97.8939	-1.27279	-9.35846	-83.0061	1.8
R1_pp_babor_down	24	-1.27279	-9.35846	-97.8939	-9.35846	-1.27279	-83.0061	1.8
R1_pr_estribor_up	25	9.35846	1.27279	-98.6007	1.27279	9.35846	-83.1994	1.8
R1_pr_estribor_down	26	1.27279	9.35846	-98.6007	9.35846	1.27279	-83.1994	1.8
R1_pr_babor_up	27	-1.27279	9.35846	-98.5871	-9.35846	1.27279	-83.1003	1.8
R1_pr_babor_down	28	-9.35846	1.27279	-97.9074	-1.27279	9.35846	-83.1051	1.8
R2_pp_estribor_up	29	1.27279	-9.35846	-124.907	9.35846	-1.27279	-110.105	1.8
R2_pp_estribor_down	30	9.35846	-1.27279	-125.587	1.27279	-9.35846	-110.1	1.8
R2_pp_babor_up	31	-9.35846	-1.27279	-124.894	-1.27279	-9.35846	-110.006	1.8
R2_pp_babor_down	32	-1.27279	-9.35846	-124.894	-9.35846	-1.27279	-110.006	1.8
R2_pr_estribor_up	33	9.35846	1.27279	-125.601	1.27279	9.35846	-110.199	1.8
R2_pr_estribor_down	34	1.27279	9.35846	-125.601	9.35846	1.27279	-110.199	1.8
R2_pr_babor_up	35	-1.27279	9.35846	-125.587	-9.35846	1.27279	-110.1	1.8
R2_pr_babor_down	36	-9.35846	1.27279	-124.907	-1.27279	9.35846	-110.105	1.8
R3_pp_estribor_up	37	1.27279	-9.35846	-151.907	9.35846	-1.27279	-137.105	1.8
R3_pp_estribor_down	38	9.35846	-1.27279	-152.587	1.27279	-9.35846	-137.1	1.8
R3_pp_babor_up	39	-9.35846	-1.27279	-151.894	-1.27279	-9.35846	-137.006	1.8
R3_pp_babor_down	40	-1.27279	-9.35846	-151.894	-9.35846	-1.27279	-137.006	1.8
R3_pr_estribor_up	41	9.35846	1.27279	-152.601	1.27279	9.35846	-137.199	1.8
R3_pr_estribor_down	42	1.27279	9.35846	-152.601	9.35846	1.27279	-137.199	1.8
R3_pr_babor_up	43	-1.27279	9.35846	-152.587	-9.35846	1.27279	-137.1	1.8
R3_pr_babor_down	44	-9.35846	1.27279	-151.907	-1.27279	9.35846	-137.105	1.8

Figura 1.8 Coordenadas iniciales para los 3 grupos de refuerzos

Una vez obtenidos los puntos iniciales y finales, procedemos a introducirlos en el programa Tdyn.

A la hora de definir que los elementos sean virtuales de forma que no se considere las fuerzas de flotación.

- A continuación se identificarán las condiciones de contorno, la primera será la superficie libre, que serán las tapas superiores, las superficies de salida del fluido que será las caras del cilindro exterior, y el fondo que serán las superficies inferiores de los cilindros.
- En la descripción del problema, habrá que garantizar que tenemos el caso de grandes profundidades. Y habrá además que identificar el punto a partir del cual se comenzará la disipación de las olas, es decir el beach. Este parámetro tendrá que ser mayor que el diámetro del cilindro interior, e inferior al exterior. Por lo que si este es de 56.7 m, la playa será de 65 m.

### Bathymetry

**Infinite depth:** to be used when the depth is much larger than the wave lengths. In this case, the depth of the domain is recommended to be at least equal to half the wave length of the largest incident wave. However, smaller computational depths can be used in combination with the *Bottom* boundary condition to simulate larger depths. This can be done when the characteristic length of the body under analysis is small compared to the computational depth.

\* **Constant depth:** to be used when the bottom is flat, and the depth is constant and smaller than half the wave lengths of the largest wave. Smaller computational depths can be used in combination with the *Bottom* boundary condition the same way it has been indicated previously.

\* **Depth:** only available if *Bathymetry=Constant depth* was selected. The real depth of the problem must be introduced in the box

**Wave absorption:** select if scattered waves generated by the presence of the body are to be absorbed in order to avoid reflection at the edge of the computational domain. The absorption area will start at a specific distance (*Beach*) from a reference point located on the free surface. Therefore, there will be no absorption in the circle with center the reference point, and radius the value introduced in "Beach". The area with no dissipation will be referred as the analysis area since no artificial dissipation is introduced on purpose to damp waves refracted and

**X absorption reference:** X coordinate of the reference point to determine the analysis and absorption area.

**Y absorption reference:** Y coordinate of the reference point to determine the analysis and absorption area.

**Absorption factor:** determines how strong the dissipation is (recommended value "1"). Large absorption factors might cause instabilities and/or wave reflection. Smaller values, while being less likely to cause instabilities, but will require larger computational domains to damp waves refracted and radiated waves.

**Beach:** determine how far, from the reference point, the free surface absorption starts.

**Sommerfeld radiation condition:** This option is only available in those cases where the body is subjected to waves, has no translational movement (turning circle/towing) and is in the absence of currents. In these cases, the largest waves can be hard to be dissipated in the absorption area unless a large computational domain is used. To avoid this situation, a Sommerfeld radiation condition can be used to allow the largest waves to leave the domain across the edge of the computational domain. Therefore, the combined action of the dissipation area and the Sommerfeld radiation condition is the best choice to avoid reflection of waves onto the edges.

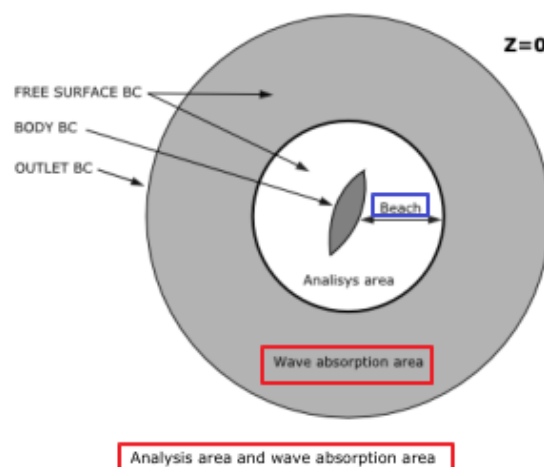


Figura 1.9 Términos junto a su definición del apartado de definición del problema. Pág. 8-9 del manual.

- Definiremos el body plan, selección el Spar completo, no habilitaremos ninguno de los grados de libertad, y como desplazamiento, se hará uso de la función volumen\*densidad, de forma que el volumen perdido tras aproximarlos con los elementos finitos, no haga que la Spar se hunda.
- Mallado, para obtener un número razonable de nodos de 150000, se tomaran los siguientes tamaños:
  1. Refuerzos y pilares, tamaño 0.6.
  2. Hard tank y Soft tank, tamaño 1.2.
  3. Superficie libre, 2.2.
  4. Fondo, 4.4.

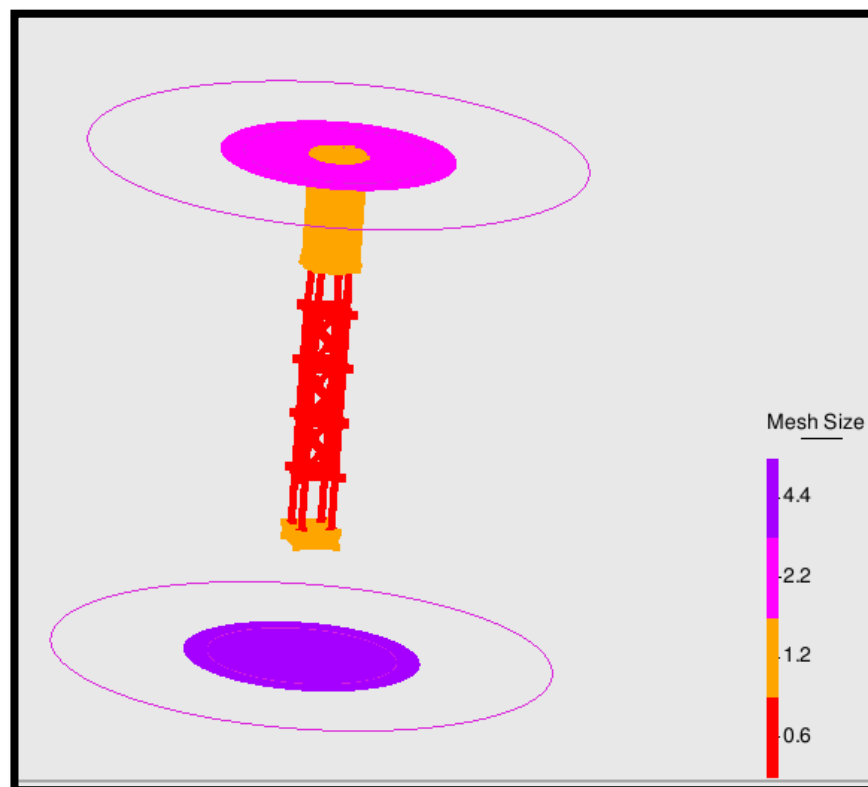


Figura 1.10 Tamaño del mallado

- En el subapartado de tiempo, se tomará un tiempo de simulación de 150 s, con una inicialización de 30 s, y el time step será de 0.1, que es inferior al que toma el programa.

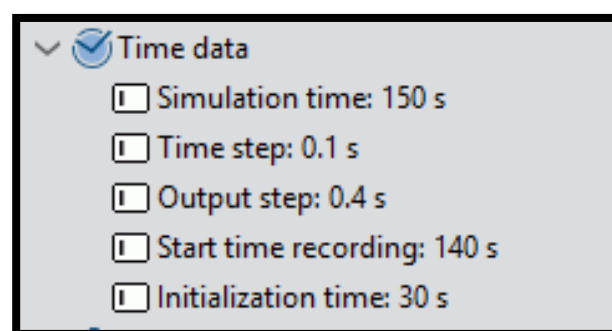


Figura 1.11 Casillas correspondientes a los períodos de tiempo

Cabe mencionar que antes de mallar la estructura, hay que primero introducir los coeficientes pertinentes, que se tendrán que introducir en los elementos esbeltos. Para ello primero vamos a entender como realiza SEAFem los cálculos, y que coeficientes habrá que introducir. Esto viene resumido en la Figura 1.12.

The force per unit length on a slender cylindrical element may be written as the sum of inertia, drag, friction and lift forces:

$$\mathbf{F} = \mathbf{F}_M + \mathbf{F}_D + \mathbf{F}_V + \mathbf{F}_F + \mathbf{F}_L \quad (10-3)$$

where the inertia force  $\mathbf{F}_M$  is oriented along the acceleration vector component normal to the element member, and its magnitude is proportional to the acceleration component. Lift force  $\mathbf{F}_L$  is oriented normal to the velocity vector and normal to the axis of the element, and its magnitude is proportional the velocity squared. Drag force  $\mathbf{F}_D$  is proportional to the squared velocity component normal to the element and normal to the lift force, while the linear drag force  $\mathbf{F}_V$  is proportional to the velocity component normal to the element. Finally, friction force  $\mathbf{F}_F$  is aligned along the axis of the element and proportional to the squared velocity components tangential to the element axis. All these are satisfied if the various force components are defined as follows:

$$\begin{aligned} \mathbf{F}_M &= (1 - \delta_v)(1 + C_M)\rho S(\mathbf{l} \times \mathbf{a}^w \times \mathbf{l}) - \rho S C_M(\mathbf{l} \times \mathbf{a}^b \times \mathbf{l}) \\ \mathbf{F}_D &= \frac{1}{2} C_D \rho D |\mathbf{l} \times \mathbf{v} \times \mathbf{l}| (\mathbf{l} \times \mathbf{v} \times \mathbf{l}) \\ \mathbf{F}_V &= \frac{1}{2} C_V \rho D (\mathbf{l} \times \mathbf{v} \times \mathbf{l}) \\ \mathbf{F}_F &= \frac{1}{2} C_F \rho \pi D |\mathbf{l} \cdot \mathbf{v}| (\mathbf{l} \cdot \mathbf{v}) \cdot \mathbf{l} \\ \mathbf{F}_L &= \frac{1}{2} C_L \rho D |\mathbf{l} \times \mathbf{v}| (\mathbf{l} \times \mathbf{v}) \end{aligned} \quad (10-4)$$

where  $D$  is a linear dimension (the diameter in the case of a cylinder),  $S$  is the cross section area,  $C_M$  is the added mass coefficient,  $C_D$  is the non-linear drag coefficient,  $C_V$  is the linear drag coefficient,  $C_F$  is the friction coefficient,  $C_L$  is the lift coefficient, and  $\delta_v$  takes the value 1 for virtual elements and 0 otherwise.

Remark: The first term in the right hand side of the  $\mathbf{F}_M$  equation, includes the Froude-Kriloff force (i.e. undisturbed wave pressure force) and the diffraction inertial force, while the second term represents the radiation inertial force.

Remark: Note that while  $C_M, C_D, C_F, C_L$  are non-dimensional coefficients,  $C_V$  has dimensions of velocity. As stated above, Eqs. (10-1) - (10-4) can estimate the different components of the force per unit length on a long (slender) structural element. Therefore, they can be integrated along the element axis, to obtain the additional forces and moments acting on the centre of gravity of the associated body.

Figura 1.12 Cálculo de las fuerzas según Seafem

Del sumatorio de fuerzas 10-3 que se ve en la Figura 1.12, se considerarán únicamente las fuerzas de arrastre  $F_d$  y las fuerzas de inercia, para ello tendremos que introducir los coeficientes  $C_m$  y  $C_d$ , para ello se hará uso de Anexo D y E, del documento DNV rp-c205, Entonces sabiendo que  $C_m=(1+C_a)$ , donde  $C_a$  es el coeficiente de masa añadida, se tiene que para elementos esbeltos:

APPENDIX D ADDED MASS COEFFICIENTS				
Table D-1 Analytical added mass coefficient for two-dimensional bodies, i.e. long cylinders in infinite fluid (far from boundaries). Added mass (per unit length) is $m_A = \rho C_A A_R$ [kg/m] where $A_R$ [m <sup>2</sup> ] is the reference area.				
Section through body	Direction of motion	$C_A$	$A_R$	Added mass moment of inertia [(kg/m) × m <sup>2</sup> ]
		1.0	$\pi a^2$	0

Figura 1.13 Valores de los coeficientes de masa añadidas para los refuerzos y los pilares de sección circular

Entonces para los refuerzos y para los pilares se les dará un  $C_m=1+1=2$ . Ahora para el coeficiente de arrastre para una sección circular.

APPENDIX E DRAG COEFFICIENTS		
Table E-1 Drag coefficient on non-circular cross-sections for steady flow $C_{DS}$ . Drag force per unit length of slender element is $f = \frac{1}{2}\rho C_{DS} Du^2$ . $D$ = characteristic width [m]. $R_e = uD/\nu$ = Reynolds number. Adopted from Blevins, R.D. (1984) <i>Applied Fluid Dynamics Handbook</i> . Krieger Publishing Co. Ref. is also made to Ch.5 for drag coefficients on I-profiles and to Ch.6 for drag coefficients on circular cylinders.		
Geometry	Drag coefficient, $C_D$	
14. Ellipse	$D/L$	$C_D (R_e \sim 10^5)$
	0.125	0.22
	0.25	0.3
	0.50	0.6
	1.00	1.0
	2.0	1.6

Figura 1.14 Valor del coeficiente de arrastre para una sección circular

Entonces para los refuerzos y los pilares se les dará un  $C_d=1$ . Y todos los otros coeficientes se dejaran en cero, de forma que las otras fuerzas (mostradas en 10-4 en la Figura 1.12 del sumatorio 10.3 de la Figura 1.12) no intervengan en el valor final.

Una vez introducidas todos los coeficientes, y tras elegir que sean elementos virtuales para todos los pilares, podemos pasar a generar el mallado según la discretización dada en la Figura 1.10.

El mallado generado tiene 155000 nodos 800000 elementos, tal y como se ve en la Figura 1.16.

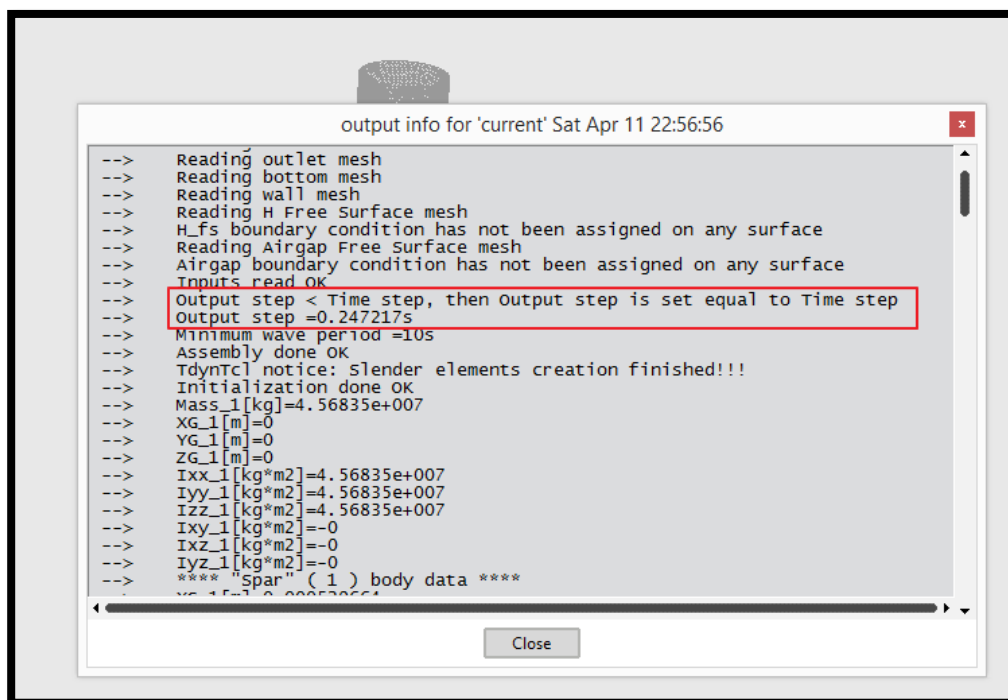


Figura 1.15 Time step de 0.247 s, que fue tomado igual que el output step, tras mallar y resolver la primera vez.

El mallado generado para la Spar es:

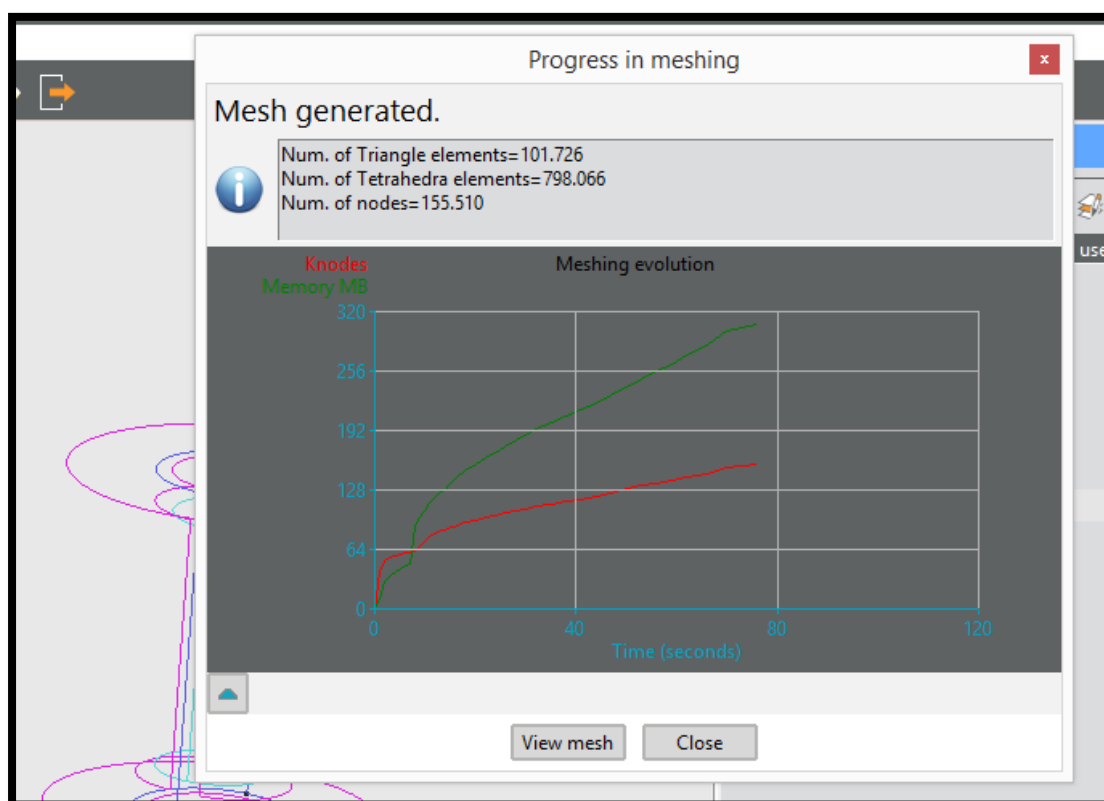


Figura 1.16 Estructura ya mallada, según los tamaños dados en la Figura 1.10.

Tras mallar la Spar, y tras introducir los datos de entrada del enunciado, periodo 10 s y altura de ola de 8, es decir amplitud de 4, podemos pasar a resolver el problema.



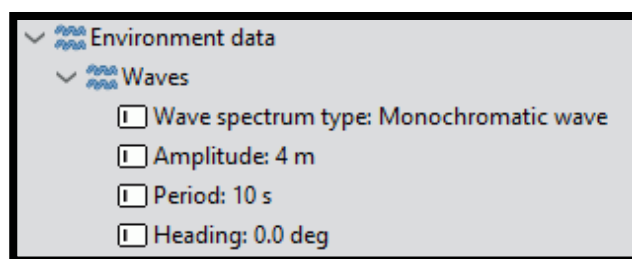


Figura 1.17 Introducción de los datos del enunciado

Tras realizar los cálculos, se han obtenido los siguientes resultados de las fuerzas totales en las 3 direcciones.

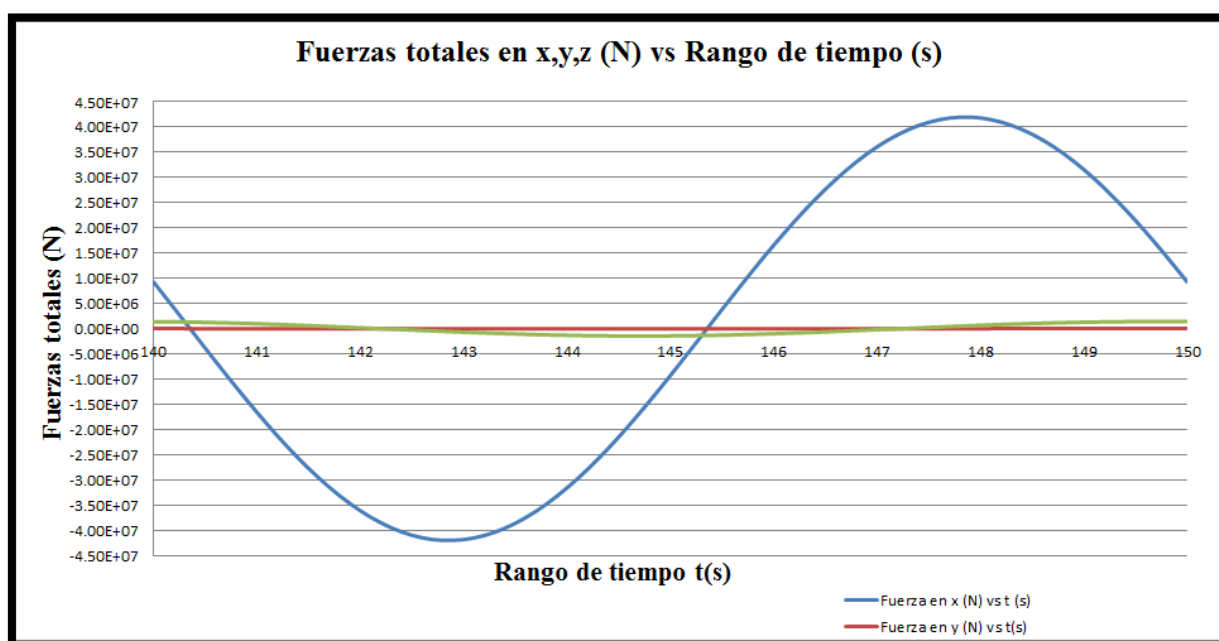


Figura 1.18 Fuerzas totales en las 3 direcciones x y z para una ola monocromática de  $A=4$  m y  $T=10$  s

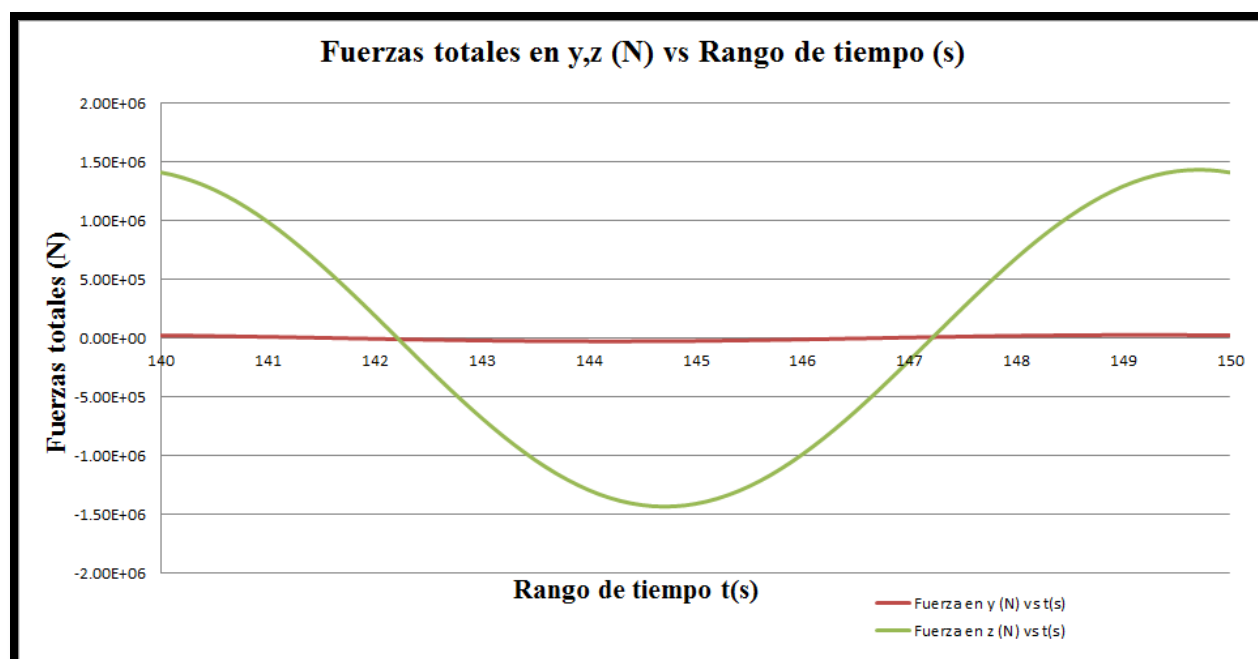


Figura 1.19 Fuerzas totales en las 3 direcciones y z para una ola monocromática de  $A=4$  m y  $T=10$  s



Como se ve en la Figura 1.18 para una ola monocromática que se propaga en la dirección x, la fuerza más grande es la fuerza x, teniendo un máximo aproximado de  $4.2 \cdot 10^7$ , y siendo un orden de magnitud mayor que las otras dos fuerzas z e y. Y si ahora comparamos las fuerzas en z e y, vemos que la fuerza z es a su vez varios ordenes de magnitud mayor que la Fuerza y, siendo esta despreciable en comparación con las otras fuerzas. Dado a que se trata de una ola bidimensional que se propaga en la dirección x, variando su elevación en el eje z, tiene sentido el no tener una componente de fuerzas en y, ya que la ola impacta a la Spar simétricamente, tal y como se ve en la Figura 1.4.

## 2º Cálculo de las fuerzas cuando la unidad está sometido a un espectro Pierson – Moskowitz con un periodo de pico $T_p$ de 12 segundos y una altura significativa de ola $H_s$ de 4 m.

Utilizando el mallado anterior, y adaptando las condiciones de las olas antes de resolver de nuevo tenemos la siguiente definición:

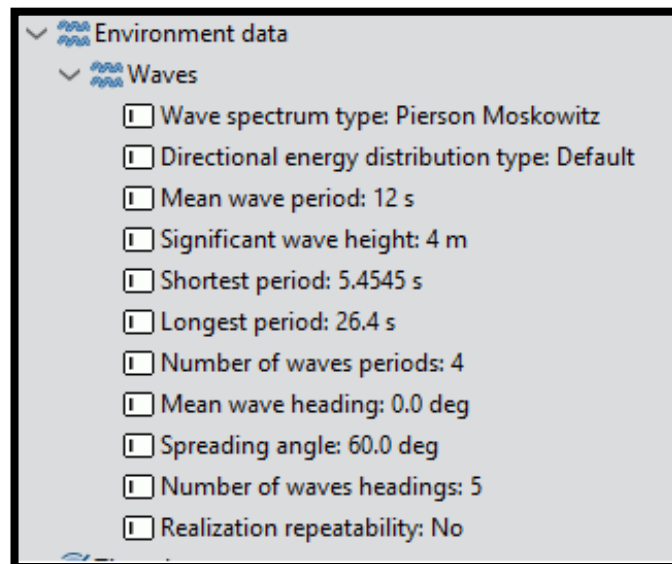


Figura 2.1 Definición del espectro P-M para la resolución del segundo apartado

Ahora bien para determinar los demás parámetros se hará uso de lo estipulado en el manual y que se muestra en la Figura 2.2. Entonces haciendo de los valores recomendados en el manual se tiene que el periodo de pico mínimo y máximo serán:

$$\text{Periodo de pico}, T_p = 12 \text{ s}$$

$$T_{min} = \frac{T_p}{2.2} = 5.4545 \text{ s}$$

$$T_{max} = T_p * 2.2 = 26.4 \text{ s}$$

Como no se piden un número mínimo de componentes en el enunciado, se tendrán 20 componentes, y tal y como nos indica al final de la Figura 2.2, el nº de olas

es el múltiplo entre el número de periodos de ola por el número de direcciones de propagación de las olas.

*Número de periodos de olas = 4*

*Número de direccion de olas = 5*

La amplitud del ángulo de propagación se mantendrá el de por defecto de 60°.

**Mean wave period:** mean wave period for wave spectrum such as Pearson Moskowitz, Jonswap, etc.

**Significant height:** significant wave period for wave spectrum such as Pearson Moskowitz, Jonswap, etc.

**Shortest period:** correspond to the wave with maximum frequency to be considered when discretizing a spectrum.  $T_{min}=T_m/2.2$  recommended.

**Longest period :** correspond to the wave with minimum frequency to be considered when discretizing a spectrum.  $T_{max}=2.2T_m$  recommended.

**Number of wave periods:** or number of wave frequencies to be used.

**Mean wave heading:** mean direction of wave propagation. It is provided in the form of an angle  $\theta$  measured with respect to the X global axis.

**Spreading angle:** angular sector  $\Delta\theta$  within which the waves propagate. Such an angular sector is always centered at the mean wave heading so that the waves propagate within the range  $[\theta - \Delta\theta/2, \theta + \Delta\theta/2]$

**Number of wave headings:** in case the waves propagate within an angular sector specified by the mean wave heading and the spreading angle, this parameter determines how many directions such an angular sector will be discretized into.

**Realization repeatability:** this option must be activated if the user wishes to run exactly the same spectrum realization in further simulations. By contrast, if such an option remains deactivated, random realizations of the same given spectrum will be used when running the simulation several times.

**Note:** the total number of waves used in the realization will be the "number of wave periods" times "Number of wave directions".

Figura 2.2 Términos que caracterizarán el espectro de JONSWAP. Pág. 15 del manual.

Los resultados obtenidos se muestran a continuación:

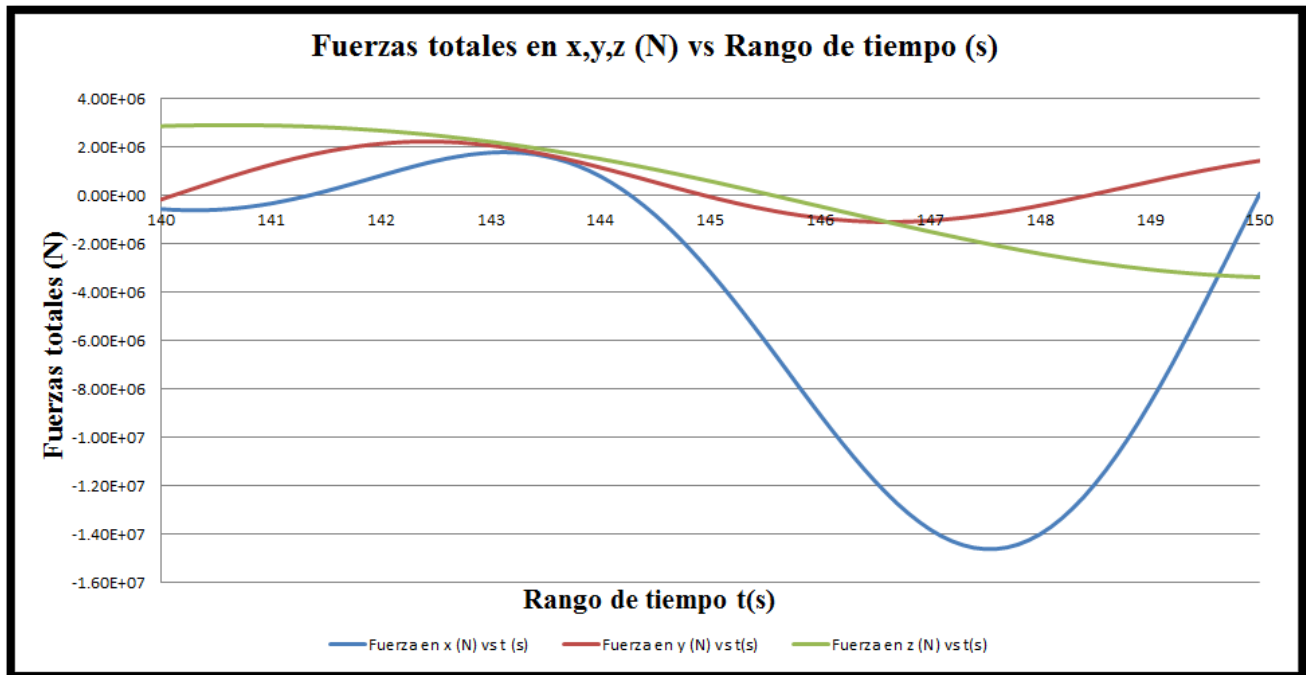


Figura 2.3 Fuerzas x,y,z para un espectro P-M de 20 componentes para  $H_s=4$  m y  $T_p=12$  s

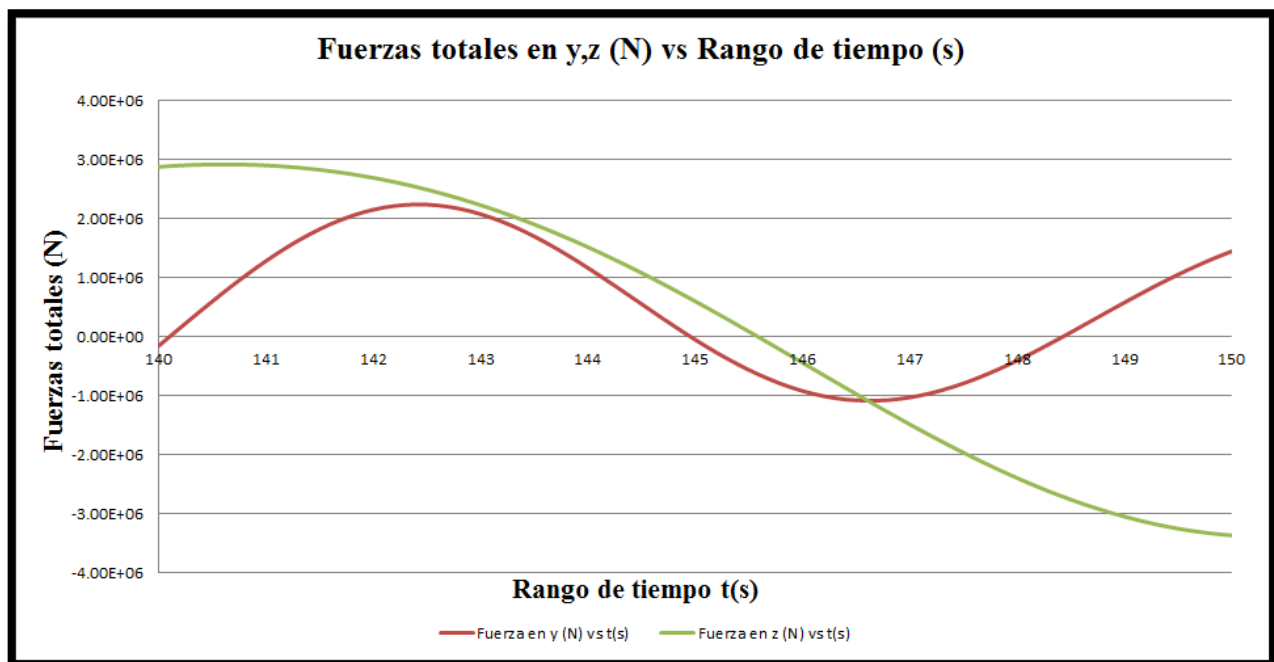


Figura 2.4 Fuerzas y,z para un espectro P-M de 20 componentes para  $H_s=4$  m y  $T_p=12$

Como se ve los resultados son significativamente inferiores a los obtenidos en el apartado anterior, que se debe principalmente a que ahora la altura de ola es 4 m, y para la ola monocromática su altura de ola era 8 m. Además de que ahora la dirección de propagación no es unidireccional por ello se obtienen valores de fuerzas  $F_x$ ,  $F_y$  y  $F_z$  de magnitudes más o menos similares.

### 3º Programación previa para comparar los resultados de fuerzas globales del apartado “a” mediante cálculo analítico con la solución obtenida usando SeaFEM, mediante Matlab.

En este apartado se programara la resolución de las fuerzas para la Spar en Matlab y se hará una comparación con los resultados obtenidos en el apartado 1º. Pero primero habrá que explicar que es lo que se han hecho los cálculos y como se han programado. Al final en un apartado aparte 4º. Se expondrá la comparación.

#### 3.1º Fuerzas en los pilares y en los refuerzos esbeltos teoría

Para hallar las fuerzas en los refuerzos y en los pilares se utilizarán las fórmulas de Morrison, que tienen en cuenta las fuerzas de inercia y de las fuerzas de arrastre. Esto es posible ya que la longitud de onda  $L=156$  m es mayor que 5 veces sus diámetros, 1.8 y 3.6 m respectivamente. El cálculo de la longitud de onda se explica más abajo, y la fórmulas de Morrison a aplicar se muestra en la Figura 3.1 siguiente.

The MOJS equation was subsequently written in vectorial form (Borgman 1958) for a vertical cylinder as

$$F = \begin{pmatrix} F_x \\ F_y \end{pmatrix} = 0.5\rho C_d D \begin{pmatrix} v_x \\ v_y \end{pmatrix} \sqrt{v_x^2 + v_y^2} + \frac{\pi}{4} \rho D^2 C_m \begin{pmatrix} a_x \\ a_y \end{pmatrix}$$

where  $v_x$  and  $v_y$  are the velocity components in the x-y plane which is normal to the cylinder axis. It is important to note that the velocity and acceleration vectors are not collinear.

Considering a segment of a long slender structural element submerged into water its local orientation is given by a unit vector

$$l = li + mj + nk$$

being  $l, m, n$  the directional cosines and  $(i, j, k)$  the unit vectors of the global coordinate system.

Similarly, the fluid velocity vector and the acceleration vector of the submerged body are given by:

$$v = v_x i + v_y j + v_z k \quad \text{Fuerza de inercia: } F_M = \rho S C_M (l \times a \times l)$$

$$a = a_x i + a_y j + a_z k \quad \text{Fuerza de arrastre: } F_D = \frac{1}{2} C_D \rho D |l \times v \times l| (l \times v \times l)$$

Figura 3.1 Formulas de Morrison para cilindros esbeltos

Para ello se hará uso de las coordenadas de los pilares mostradas en la Figura 1.5 y de los refuerzos mostrados en la Figura 1.8. Con los cuales haremos el triple producto vectorial que nos permitirá obtener los vectores de aceleración y velocidad normales al eje axial del elemento en cuestión. En el programa Matlab el vector se llama  $ci$ , y se han ido almacenado en la matriz  $mx3$   $c\_axil$ , en donde  $m$  es el número elementos y en donde las 3 columnas corresponden a las 3 coordenadas cartesianas  $x, y, z$ .

El motivo por el que el triple producto vectorial de 2 vectores “a” y “b”, es debido a la propia definición del producto vectorial, en donde el vector resultante es normal al plano que contiene a los vectores del producto, es decir “a” y “b”. Por lo  $(a \times b) \times a$ , nos da primero un vector  $a \times b = c$ , que es normal a los dos, y si volvemos a multiplicar esa resultante con el vector al que queremos hallarle la normal, es decir “a”, nos da por lo tanto ahora un cuarto vector “d”, que es normal a “c” y “a”.

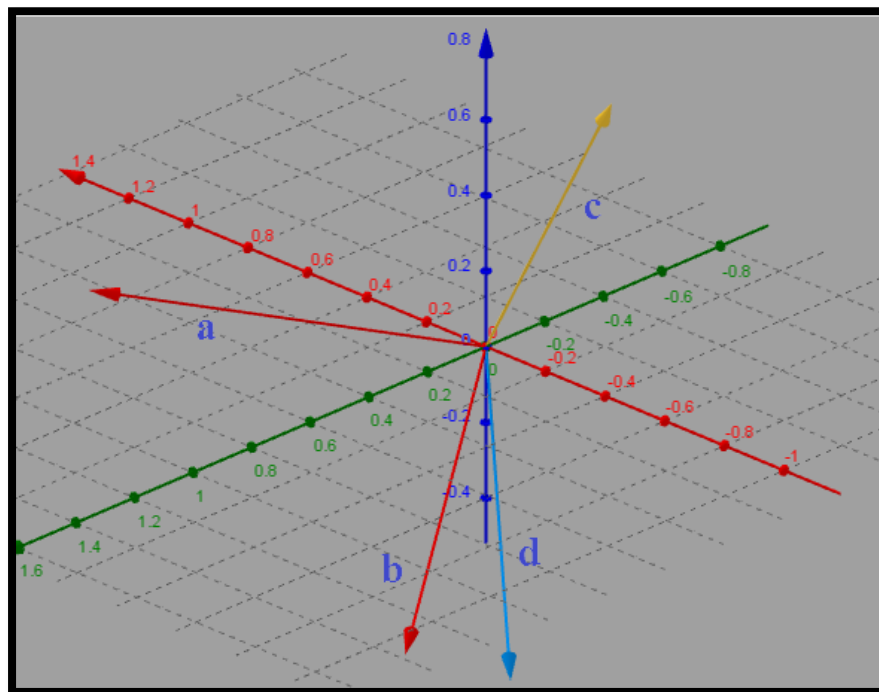


Figura 3.2 Triple producto vectorial, resultado de multiplicar  $a \times b \times a$  da un vector “d” normal a “a”

### 3.2° Calculo iterativo para hallar la longitud de onda

Ahora bien, antes de poder pasar a hacer los cálculos pertinentes nos tenemos que asegurar de que los elementos son esbeltos, para ello tendremos que hallar la longitud entre olas. Sabiendo que la celeridad de transmisión de movimiento C se puede expresar en función de L y del periodo T, o en función de L y de la profundidad d, podemos obtener L,

$$C = \frac{L}{T}$$

$$C = \sqrt{\frac{g * L}{2\pi} * \tanh\left(\frac{2\pi * d}{L}\right)}$$

$$C^2 = \frac{g * L}{2\pi} * \frac{T}{T} * \tanh\left(\frac{2\pi * d}{L}\right) = \frac{g * T}{2\pi} * C * \tanh\left(\frac{2\pi * d}{L}\right)$$

$$C = \frac{g * T}{2\pi} * \tanh\left(\frac{2\pi * d}{L}\right) = \frac{L}{T}$$

Entonces:

$$L = \frac{g * T^2}{2\pi} * \tanh\left(\frac{2\pi * d}{L}\right)$$

Pero como vemos tenemos el valor de la eslora en ambos lado de la ecuación por lo que vamos a hallar la solución iterativamente, en donde se buscará el valor de L donde el error cometido e es mínimo. Es decir:

$$e = \frac{g * T^2}{2\pi} * \tanh\left(\frac{2\pi * d}{L}\right) - L$$

La programación iterativa con Matlab se ha hecho en una subrutina llamada [L]=longitud\_iterativo(g,T,d), que solo entra si L no es un dato de entrada en el archivo datos.inp. de forma resumen se hace dos iteraciones en donde se obtiene una aproximación burda de Lmin y después se realiza otra más aumentado la precisión.

```
function [L]=longitud_iterativo(g,T,d)
%e=error=(g*(T^2)/(2*pi))*tanh(2*pi*d/L)-L

e=zeros(40,2); %Inicializamos la matriz donde almacenaremos los errores
%para poder despues buscar en el el valor mínimo
i=1;
for L=1:10:400 %olas de 1 a 400 m de eslora en saltos de 10 m
    error=(g*(T^2)/(2*pi))*tanh(2*pi*d/L)-L;
    e(i,1)=L;%Almacenamos la longitud L del error e
    e(i,2)=error;%Almacenamos el error para la longitud L de onda
    i=i+1;
end
emin=min(abs(e(:,2))); %Hallamos el valor más pequeño
%Ahora recorremos la segunda columna de errores de la matriz e donde
%buscamos la longitud L que le corresponde al error emin
for i=1:30
    if emin == abs(e(i,2))
        Lmin=e(i,1);
    end
end
%Ahora realizamos exactamente lo mismo que ante pero con mayor precisión
i=1;
Linf=Lmin-10;
Lsup=Lmin+10;
n=(Lsup-Linf)/40;
for L=Linf:n:Lsup %olas de 1 a 400 m de eslora en saltos de (Lsup-Linf)/40
    error=(g*(T^2)/(2*pi))*tanh(2*pi*d/L)-L;
    e(i,1)=L;
    e(i,2)=error;
    i=i+1;
end
emin=min(abs(e(:,2)));
for i=1:30
    if emin == abs(e(i,2))
        Lmin=e(i,1);
    end
end
end
```

```
L=Lmin; %Renonbramos la longitud de onda L al valor minimo Lmin
end
```

Conociendo que  $L=156$  m, se tiene que  $L>5*3.6$  y que  $L>5*1.8$ , que son los diámetros de los pilares y de los refuerzos.

### 3.3° Lectura de datos del problema

Para poder realizar los cálculos primero de todo hay que introducir los datos de entrada. Para ello en un archivo datos.inp se introducirán los datos tal y como se muestra en la Figura siguiente. Y según viene explicado en término de unidades.

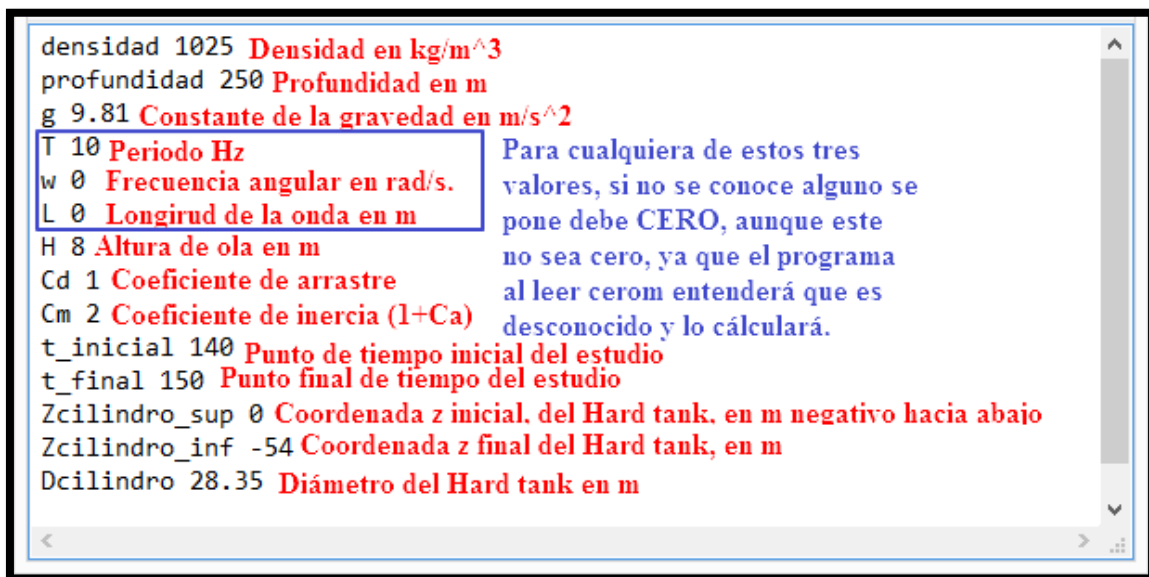


Figura 3.3 Archivo de datos de entrada necesario para los cálculos

Entonces, teniendo ya introducidos todos los datos en el archivo datos.inp podemos pasar a cargarlos en el programa Matlab. Así que primero cargamos el archivo, y obtenemos cualquier dato que no se conozca. Y en donde se entrará o no a la subrutina antes menciona en el subapartado 3.2 para hallar la longitud de onda.

```
DATOS = importdata('datos.inp'); %cargamos el archivo datos.inp
DATOS = table(DATOS.textdata, DATOS.data); %creamos una tabla con nuestros datos
DATOS1 = table2array(DATOS(:,2)); %extraemos la columna de la tabla y creamos una matriz
```

### Extraemos los datos del documento datos

```
densidad=DATOS1(1,1); %Densidad, kg*103/m3
d=DATOS1(2,1); %Profundidad, m
g=DATOS1(3,1); %Constante gravedad, m/s2
T=DATOS1(4,1); %Perido, s
w=DATOS1(5,1); %Frecuencia, Hz
if T~=0 && w==0
    w=2*pi/T;
end
```



```

L=DATOS1(6,1);           %Longitud de onda, m
if L~=0
    k=2*pi/L;           %Número de onda rad/m
end
if w==0
    w=sqrt(g*k*tanh(k*d));
end
if T==0
    T=2*pi/w;
end
if L==0
    [L]=longitud_iterativo(g,T,d);
end
k=2*pi/L;           %Número de onda rad/m

H=DATOS1(7,1);           %altura de ola, m
Cd=DATOS1(8,1);
Cm=DATOS1(9,1);
    
```

### 3.4° Programación para el cálculo de la fuerzas de Morrison.

Ahora bien a parte de la lectura de los datos del problema, es necesario para hallar las fuerzas horizontales, introducir las coordenadas iniciales y finales de todos los elementos de estudio. Estas coordenadas no son otras que las mostradas en la Figura 1.5 y la Figura 1.8. Y que se mostrará de nuevo por conveniencia.

	Nº	Punto inicial (abajo)			Punto final (arriba)			D (m)
		x	y	z	x	y	z	
pilar 1 popa	1	0	-10.6312	-76.5	0	-9.45	-54	3.6
pilar 1 proa	2	0	10.6312	-76.5	0	9.45	-54	3.6
pilar 1 babor	3	-10.6312	0	-76.5	-9.45	0	-54	3.6
pilar 1 estribor	4	10.6312	0	-76.5	9.45	0	-54	3.6
pilar 2 popa	5	0	-10.6312	-103.5	0	-10.6312	-77.4	3.6
pilar 2 proa	6	0	10.6312	-103.5	0	10.6312	-77.4	3.6
pilar 2 babor	7	-10.6312	0	-103.5	-10.6312	0	-77.4	3.6
pilar 2 estribor	8	10.6312	0	-103.5	10.6312	0	-77.4	3.6
pilar 3 popa	9	0	-10.6312	-130.5	0	-10.6312	-104.4	3.6
pilar 3 proa	10	0	10.6312	-130.5	0	10.6312	-104.4	3.6
pilar 3 babor	11	-10.6312	0	-130.5	-10.6312	0	-104.4	3.6
pilar 3 estribor	12	10.6312	0	-130.5	10.6312	0	-104.4	3.6
pilar 4 popa	13	0	-10.6312	-157.5	0	-10.6312	-131.4	3.6
pilar 4 proa	14	0	10.6312	-157.5	0	10.6312	-131.4	3.6
pilar 4 babor	15	-10.6312	0	-157.5	-10.6312	0	-131.4	3.6
pilar 4 estribor	16	10.6312	0	-157.5	10.6312	0	-131.4	3.6
pilar 5 popa	17	0	-10.6312	-185.4	0	-10.6312	-158.4	3.6
pilar 5 proa	18	0	10.6312	-185.4	0	10.6312	-158.4	3.6
pilar 5 babor	19	-10.6312	0	-185.4	-10.6312	0	-158.4	3.6
pilar 5 estribor	20	10.6312	0	-185.4	10.6312	0	-158.4	3.6

Figura 3.4 Coordenadas de los pilares.



	Nº	Punto inicial (abajo)			Punto final (arriba)			D (m)
		x	y	z	x	y	z	
R1_pp_estribor_up	21	1.27279	-9.35846	-97.9074	9.35846	-1.27279	-83.1051	1.8
R1_pp_estribor_down	22	9.35846	-1.27279	-98.5871	1.27279	-9.35846	-83.1003	1.8
R1_pp_babor_up	23	-9.35846	-1.27279	-97.8939	-1.27279	-9.35846	-83.0061	1.8
R1_pp_babor_down	24	-1.27279	-9.35846	-97.8939	-9.35846	-1.27279	-83.0061	1.8
R1_pr_estribor_up	25	9.35846	1.27279	-98.6007	1.27279	9.35846	-83.1994	1.8
R1_pr_estribor_down	26	1.27279	9.35846	-98.6007	9.35846	1.27279	-83.1994	1.8
R1_pr_babor_up	27	-1.27279	9.35846	-98.5871	-9.35846	1.27279	-83.1003	1.8
R1_pr_babor_down	28	-9.35846	1.27279	-97.9074	-1.27279	9.35846	-83.1051	1.8
R2_pp_estribor_up	29	1.27279	-9.35846	-124.907	9.35846	-1.27279	-110.105	1.8
R2_pp_estribor_down	30	9.35846	-1.27279	-125.587	1.27279	-9.35846	-110.1	1.8
R2_pp_babor_up	31	-9.35846	-1.27279	-124.894	-1.27279	-9.35846	-110.006	1.8
R2_pp_babor_down	32	-1.27279	-9.35846	-124.894	-9.35846	-1.27279	-110.006	1.8
R2_pr_estribor_up	33	9.35846	1.27279	-125.601	1.27279	9.35846	-110.199	1.8
R2_pr_estribor_down	34	1.27279	9.35846	-125.601	9.35846	1.27279	-110.199	1.8
R2_pr_babor_up	35	-1.27279	9.35846	-125.587	-9.35846	1.27279	-110.1	1.8
R2_pr_babor_down	36	-9.35846	1.27279	-124.907	-1.27279	9.35846	-110.105	1.8
R3_pp_estribor_up	37	1.27279	-9.35846	-151.907	9.35846	-1.27279	-137.105	1.8
R3_pp_estribor_down	38	9.35846	-1.27279	-152.587	1.27279	-9.35846	-137.1	1.8
R3_pp_babor_up	39	-9.35846	-1.27279	-151.894	-1.27279	-9.35846	-137.006	1.8
R3_pp_babor_down	40	-1.27279	-9.35846	-151.894	-9.35846	-1.27279	-137.006	1.8
R3_pr_estribor_up	41	9.35846	1.27279	-152.601	1.27279	9.35846	-137.199	1.8
R3_pr_estribor_down	42	1.27279	9.35846	-152.601	9.35846	1.27279	-137.199	1.8
R3_pr_babor_up	43	-1.27279	9.35846	-152.587	-9.35846	1.27279	-137.1	1.8
R3_pr_babor_down	44	-9.35846	1.27279	-151.907	-1.27279	9.35846	-137.105	1.8

Figura 3.5 Coordenada de los refuerzos

Punto inicial del elemento, corresponde al punto más bajo de este, o más a la izquierda del refuerzo si es del subgrupo "up" o más a la derecha si pertenece al subgrupo "down". Viendo la cruz localmente, indistintamente de su orientación global. Ver Figura 1.7

Punto final del elemento, corresponde al punto más alto de este, o más a la derecha del refuerzo si es del subgrupo "up" o más a la izquierda si pertenece al subgrupo "down". Viendo la cruz localmente, indistintamente de su orientación global. Ver Figura 1.7

Coordenada y				Coordenada y				Diámetro m
Coordenada x	Coordenada z	Coordenada x	Coordenada z	Coordenada x	Coordenada z	Coordenada x	Coordenada z	
1	0	-10.6312	-76.5	0	-9.45	-54	3.6	Coordenadas del primer pilar, Pilar1popa
2	0	10.6312	-76.5	0	9.45	-54	3.6	
3	-10.6312	0	-76.5	0	-9.45	0	-54	
4	10.6312	0	-76.5	0	9.45	0	-54	
5	0	-10.6312	-103.5	0	-10.6312	-77.4	3.6	Coordenadas del primer pilar Pilar2popa del grupo P2, es decir el siguiente nivel
6	0	10.6312	-103.5	0	10.6312	-77.4	3.6	
7	-10.6312	0	-103.5	-10.6312	0	-77.4	3.6	
8	10.6312	0	-103.5	10.6312	0	-77.4	3.6	
9	0	-10.6312	-130.5	0	-10.6312	-104.4	3.6	
10	0	10.6312	-130.5	0	10.6312	-104.4	3.6	
11	-10.6312	0	-130.5	-10.6312	0	-104.4	3.6	
12	10.6312	0	-130.5	10.6312	0	-104.4	3.6	
13	0	-10.6312	-157.5	0	-10.6312	-131.4	3.6	
14	0	10.6312	-157.5	0	10.6312	-131.4	3.6	
15	-10.6312	0	-157.5	-10.6312	0	-131.4	3.6	
16	10.6312	0	-157.5	10.6312	0	-131.4	3.6	
17	0	-10.6312	-185.4	0	-10.6312	-158.4	3.6	
18	0	10.6312	-185.4	0	10.6312	-158.4	3.6	
19	-10.6312	0	-185.4	-10.6312	0	-158.4	3.6	
20	10.6312	0	-185.4	10.6312	0	-158.4	3.6	
21	1.27279	-9.35846	-97.9074	9.35846	-1.27279	-83.1051	1.8	Coordenadas del primer subgrupo R1 de refuerzos
22	9.35846	-1.27279	-98.5871	1.27279	-9.35846	-83.1003	1.8	
23	-9.35846	-1.27279	-97.8939	-1.27279	-9.35846	-83.0061	1.8	
24	-1.27279	-9.35846	-97.8939	-9.35846	-1.27279	-83.0061	1.8	
25	9.35846	1.27279	-98.6007	1.27279	9.35846	-83.1994	1.8	
26	1.27279	9.35846	-98.6007	9.35846	1.27279	-83.1994	1.8	
27	-1.27279	9.35846	-98.5871	-9.35846	1.27279	-83.1003	1.8	
28	-9.35846	1.27279	-97.9074	-1.27279	9.35846	-83.1051	1.8	
29	1.27279	-9.35846	-124.907	9.35846	-1.27279	-110.105	1.8	

Figura 3.7 Explicación de cómo están organizados los puntos y como se deben introducir para su correcta lectura

El primer paso será extraer los datos necesarios tras cargar el archivo nodos.list, y inicializar todas aquellos vectores donde almacenaremos los datos.

### Inicializamos los vectores donde almacenaremos los resultados

```
m=size(nodos,1); %Obtenemos el número de elementos que tenemos a partir del
                    %tamaño de la matriz nodos

Txyz_refuerzos=zeros(m,6); %En este vector extraeremos del archivo nodos las coordenadas
                    %finales e iniciales de cada elemento
Diam=zeros(m,1); %En este vector almacenaremos los valores del diámetro de
                    %cada elemento
for i=1:m

    Txyz_refuerzos(i,:)=nodos(i,2:7);%Extraemos las coordenadas almacenadas en las
    columnas
    %2 a la 7, en donde de 2-4 se encuentran las coordenadas del
    %punto inicial, y de 5-7 se encuentran las coordenadas del último punto

    Diam(i,1)=nodos(i,8); %y se utilizara para introducir la z, este vector
    %son las coordenadas de los vectores de posicion final e inicial
end

%F=C*abs(Vn)*Vnx+K*anx, (kg/m^3)*(m^2)*(m/s^2)
Vnormal_total=zeros(3*m,3); %Es 3*m filas ya que cada elemento tendra 1 fila para el
                    %punto inicial, 1 fila para el punto
                    %intermedio, y una tercera fila para el punto
                    %final. Las columnas son vx,vy,z
anormal_total=zeros(3*m,3); %Igual que para las velocidades

vv=0;%variable auxiliar que me permitirá almacenar las velocidades y aceleraciones
%normales en sus vectores

%Ahora inicializaremos tanto el vector donde almacenaremos las fuerzas
%distribuidas, FNxm, en donde la primera columna corresponde a la fuerza
%distribuida en el primer punto del elemento, que corresponde al punto más
%"bajo" o mas "a popa", la segunda columna corresponderá al punto intermdio
%y la tercera columna corresponderá al punto final del elemento que se
%encuentra mas a proa o mas arriba del elemento en cuestion.
FNxm_braising=zeros(m,3);%Vector que almacena las fuerzas distribuidas en x, 1ºcolumna
                    %corresponde al primer punto, la 2º para el intermedio y
                    %la 3º para el ultimo punto
FNzm=zeros(m,3); %Igual que para FNxm, pero para la dirección z
FNx_braising=zeros(m,1); %Vector donde almacenaremos los valores totales de las fuerzss
en x
FNz_braising=zeros(m,1); %Vector donde almacenaremos los valores totales de las fuerzss
en z
c_axial=zeros(m,3); %Vector donde almacenaremos los vectores directores de cada
elemento
```

Entonces, ahora deberemos especificar el rango de tiempo de nuestro estudio, para ello se extraerán los datos de tiempo cargados en el archivos datos.inp y mostrado en la Figura 3.3.

## Intervalo de tiempo de estudio

```
t_inicial=DATOS1(10,1); %Punto de inicio del calculo
t_final=DATOS1(11,1); %Punto final del calculo
nt=1; %Numero de pasos donde haremos los calculos que se van a evaluar, es decir
      %hallaremos las fuerzas cada 2 segundos si es de 0-150s el rango.
nwt=1+(t_final-t_inicial)/nt;%Se le añade el 1, para que me tenga en cuenta los
extremos
F_refuerzosx_rangot=zeros(nwt,m);%En este vector se almacenará los valores de las
fuerzas

      %en los distintos puntos de tiempo t. en donde las
      %columnas corresponden a distintos elementos brazing y las
      %filas a los distinto valoroes de tiempo t
```

Entonces ahora ya podemos pasar a explicar la programación del ejercicio. Para un punto de tiempo  $t$ , y dentro del for que nos recorre todos los elementos, podemos agrupar para las fuerzas de Morrison que se muestran en la Figura 3.1 todos los valores constantes en dos variables  $C$  y  $K$ .

Tras ello se procede a hallar los vectores normales axiales al eje del elemento, para ello sólo se necesitan las coordenadas finales e iniciales del elemento, y que se han extraído del archivo nodos y almacenados ahora en una matriz  $Txyz\_refuerzos$  de tamaño  $m \times 9$ . En donde las 3 primeras columnas corresponden al punto inicial, las tres siguientes al punto intermedio y las 3 últimas al punto final. Haciendo uso de los puntos iniciales y finales también se puede hallar la longitud del elemento mediante Pitágoras, que será necesaria conocer para poder realizar la integración numérica.

```
%Vector direccional del elemento i. Lo obtendremos haciendo uso de las
      %coordenas iniciales y finales del elemento en cuestión
ci=zeros(1,3);
e1=Txyz_refuerzos(i,1:3); %punto inicial del elemento, (punto más bajo)
e2=Txyz_refuerzos(i,4:6); %punto final del elemento, (punto más alto)
ci(1,:)=(e2-e1)/norm(e2-e1); %vector unitario axial a la barra
c_axial(i,:)=ci(1,:); %almacenamos loa vectores directores
long=norm(e2-e1); %Tambien podemos hallar la longitud del elemento
```

Ahora bien, para obtener la fuerza total que se aplica sobre el elemento es necesario integrar la fuerza distribuida obtenida por Morrison. Para ello se utilizará la primera regla de Simpson y que hará uso de 3 puntos de integración, los dos extremos y uno intermedio. Los tres puntos de integración se hallarán mediante un for en donde la variable  $j$  variará de 1 a 3.

Por ello se calcula el punto intermedio y se almacena en sus columnas correspondientes de  $Txyz\_refuerzos$ .

```
%Obtenemos las coordenadas del punto intermedio
eintermedio=(e1+e2)/2; %Nos da el punto intermedio del brazing
%Vector coordenadas del elemento i
xyz=zeros(1,9);
xyz(1,1:3)=Txyz_refuerzos(i,1:3); %almacenamos las coordenadas del punto inicial
```

```
xyz(1,4:6)=eintermedio(1,:);%almacenamos las coordenadas del punto intermedio
xyz(1,7:9)=Txyz_refuerzos(i,4:6);%almacenamos las coordenadas del punto final
```

Ahora ya podemos pasar a hallar las velocidades y las aceleraciones, según las fórmulas siguientes extraídas del DNV:

Recommended Practice DNV-RP-C205, October 2010 Page 32		
Table 3-1 Gravity wave theory		
Parameter	Airy wave theory	
	General water depth	Deep water
Horizontal particle velocity, $u$	$\frac{\pi H}{T} \frac{\cosh[k(z+d)]}{\sinh(kd)} \cos \theta$	$\frac{\pi H}{T} e^{kz} \cos \theta$
Vertical particle velocity, $w$	$\frac{\pi H}{T} \frac{\sinh[k(z+d)]}{\sinh(kd)} \sin \theta$	$\frac{\pi H}{T} e^{kz} \sin \theta$
Horizontal particle acceleration, $\dot{u}$	$\frac{2\pi^2 H}{T^2} \frac{\cosh[k(z+d)]}{\sinh(kd)} \sin \theta$	$\frac{2\pi^2 H}{T^2} e^{kz} \sin \theta$
Vertical particle acceleration, $\dot{w}$	$-\frac{2\pi^2 H}{T^2} \frac{\sinh[k(z+d)]}{\sinh(kd)} \cos \theta$	$-\frac{2\pi^2 H}{T^2} e^{kz} \cos \theta$

Figura 3.8 Fórmulas con las que se hallarán las velocidades y las aceleraciones. Extraídas del DNV-RP-205

Entonces, teniendo en cuenta de la la relación  $L/d$ , o longitud de ola/profundidad podemos saber si nos encontramos en aguas profundas o aguas poco profundas. Para el caso de este problema,  $d/L=250/156 = 1.624 > 0.5$ , por lo que nos encontramos en aguas profundas. Se ha tomado  $d=250$ , porque esa es la altura que se le ha dado al cilindro interior y al cilindro exterior en la modelización de GiD.

## B.6 Wave Classification after Water Depth

$h/L < \frac{1}{20}$  , shallow water waves

$h/L > \frac{1}{2}$  , deep water waves

Figura 3.9 Aguas profundas, Aguas limitadas, y aguas transitorias entre ambos limites

Entonces las velocidades son:

```
%Dependiendo de si tenemos aguas profundas o aguas limitas se
%utilizará una u otra expresión para la reducción de la amplitud
%con la profundidad
if d/L < 0.5 %Aguas transitorias, general water depth
    %La velocidad en la dirección x es:
    vx=(pi *H/T)*(cosh(k*(d+z))/sinh(k*d))*cos(k*x-wt);
    %La velocidad en la dirección z es:
    vz=(pi *H/T)*(sinh(k*(d+z))/sinh(k*d))*sin(k*x-wt);
else %d/L > 0.5 1/2 aguas profundas
    %La velocidad en la dirección x es:
    vx=(pi *H/T)*(exp(k*z))*cos(k*x-wt);
    %La velocidad en la dirección z es:
    vz=(pi *H/T)*(exp(k*z))*sin(k*x-wt);
end
```

Ahora bien, tal y como se exige en la Figura 3.1, esta velocidad no es la normal, por lo que tendremos que realizar el triple producto de  $\mathbf{ci} \times (\mathbf{v}_x \text{ o } \mathbf{v}_z) \times \mathbf{ci}$ , para encontrar la velocidad normal al eje del elemento, tal y como se ve en la Figura 3.2.

```
%Almacenamos el vector de velocidades en un vector columna
vi_sin_normalizar=zeros(3,1);
vi_sin_normalizar(1,1)=vx;
vi_sin_normalizar(3,1)=vz;

%Obtenemos el triple producto para la fuerza arrastre c1xvixc1, de
%forma que obtenemos la velocidad normal al elemento
vnormal=cross(cross(c1,vi_sin_normalizar),c1);
vnormal=vnormal'; %Pasamos de un vector fila a un vector columna
%Calculamos la magnitud del vector normal
vnormal_abs=norm(vnormal);
```

Lo mismo que se hizo para la velocidad, se realiza de nuevo para la aceleración haciendo uso de las expresiones de la aceleración dadas en la Figura 3.8, y obteniendo después y almacenando la aceleración normal en  $\mathbf{anormal}$ .

Entonces ya podemos hallar el valor de la fuerza en el punto de integración  $j=1$ , como:

```
%Calculamos las fuerzas de arrastre e inercia
FD=C*vnormal_abs*vnormal; %Valor de la fuerza de arrastre en N/m
FI=K*anormal; %Valor de la fuerza de inercia en N/m
FNmi=FD+FI; %Valor de la fuerza total en N/m
```

Si almacenamos el vector  $\mathbf{FNmi}$  dentro de los vectores  $\mathbf{FNxm\_brazing}$  y  $\mathbf{FNzm\_brazing}$ , ya que la componente  $\mathbf{FNym\_brazing}$  es igual a cero en todos los casos. Si hacemos lo mismo para los puntos de integración  $j=2$  y  $j=3$ , que corresponden al punto intermedio y punto final, podemos hacer la integración numérica utilizando la primera regla de Simpson.

Realizamos las integraciones numéricas y obtenemos las fuerzas totales en la dirección x y la dirección z.

```
%El espaciado de la integración es, para el caso de los brazing la
%mitad de la longitud
espaciado=(long/2);
%Obtenemos la fuerza en x haciendo uso de la fuerza distribuida FNm,
%para ello se aplicará la primera regla de simpson
FNxi_brazing=espaciado*(1/3)*(1*FNxm_brazing(i,1)+4*FNxm_brazing(i,2) ...
+1*FNxm_brazing(i,3));
FNx_brazing(i,1)=FNxi_brazing;

FNzi_brazing=espaciado*(1/3)*(1*FNzm_brazing(i,1)+4*FNzm_brazing(i,2) ...
+1*FNzm_brazing(i,3));
FNz_brazing(i,1)=FNzi_brazing;
```

Una vez hallado todas las fuerzas en todos los elemento m para el punto de tiempo t almacenamos las fuerzas en otra matriz, en donde la fuerzas de los distintos elementos m estarán dispuesto en las columnas, y las filas corresponderán a los distintos puntos de tiempo del estudio.

```
%Almacenamos las fuerzas de todos los elementos en las distintas
%columna, i, para el mismo valor de tiempo. Ponemos t-(t_inicial-1)
%de forma que si t_inicial no se parte de cero, no nos ponga las
%demas casillas como cero y al final los resultados
F_refuerzosx_rangot(t-(t_inicial-1),i)=FNx_brazing(i,1);
F_refuerzosz_rangot(t-(t_inicial-1),i)=FNz_brazing(i,1);
```

Ahora para poder obtener el valor de las fuerzas en conjunto de todos los elementos, solo hay que sumar la fuerza horizontal final de cada elemento.

```
% Hallamos las fuerzas total de todos los refuerzos, por lo que hacemos el
% sumatorio de todas las fuerzas para el mismo valor de tiempo t
Ftotalx_refuerzos_rangot=zeros(nwt,1);
t=t_inicial;
while t <= t_final
    for i=1:m
        Ftotalx_refuerzos_rangot(t-(t_inicial-1),1)=...
            Ftotalx_refuerzos_rangot(t-(t_inicial-1),1)...
            +F_refuerzosx_rangot(t-(t_inicial-1),i);
    end
    t=t+nt;
end

plot(linspace(t_inicial,t_final,nwt),Ftotalx_refuerzos_rangot,'r-');
xlabel('Rango de tiempo, t (s)');
ylabel('Fuerzas total x en los pilares y refuerzos (N)');
title('Fuerzas total x en los pilares y refuerzos (N) vs t (s)');
```

### 3.5° Programación del cálculo de las fuerzas sobre el ponton/soft tank

Ahora, se va a proceder a hallar las fuerzas sobre el Soft tank, al que nos referiremos como pontón, por comodidad. Para el caso del pontón, al tratarse de una estructura compacta, las fuerzas dominantes son las fuerzas de Froude krylov

Tomando como variable característica, el valor medio de sus lados, tenemos que es  $(a+a)/2 = a$ , siendo “a” el lado del volumen compacto que es igual a 20.0465 m, esto no da un  $H/D = 8/20.0465 = 0.3991$  y un parámetro de difracción  $\pi D/L = \pi \cdot 20.0465/156 = 0.4037$

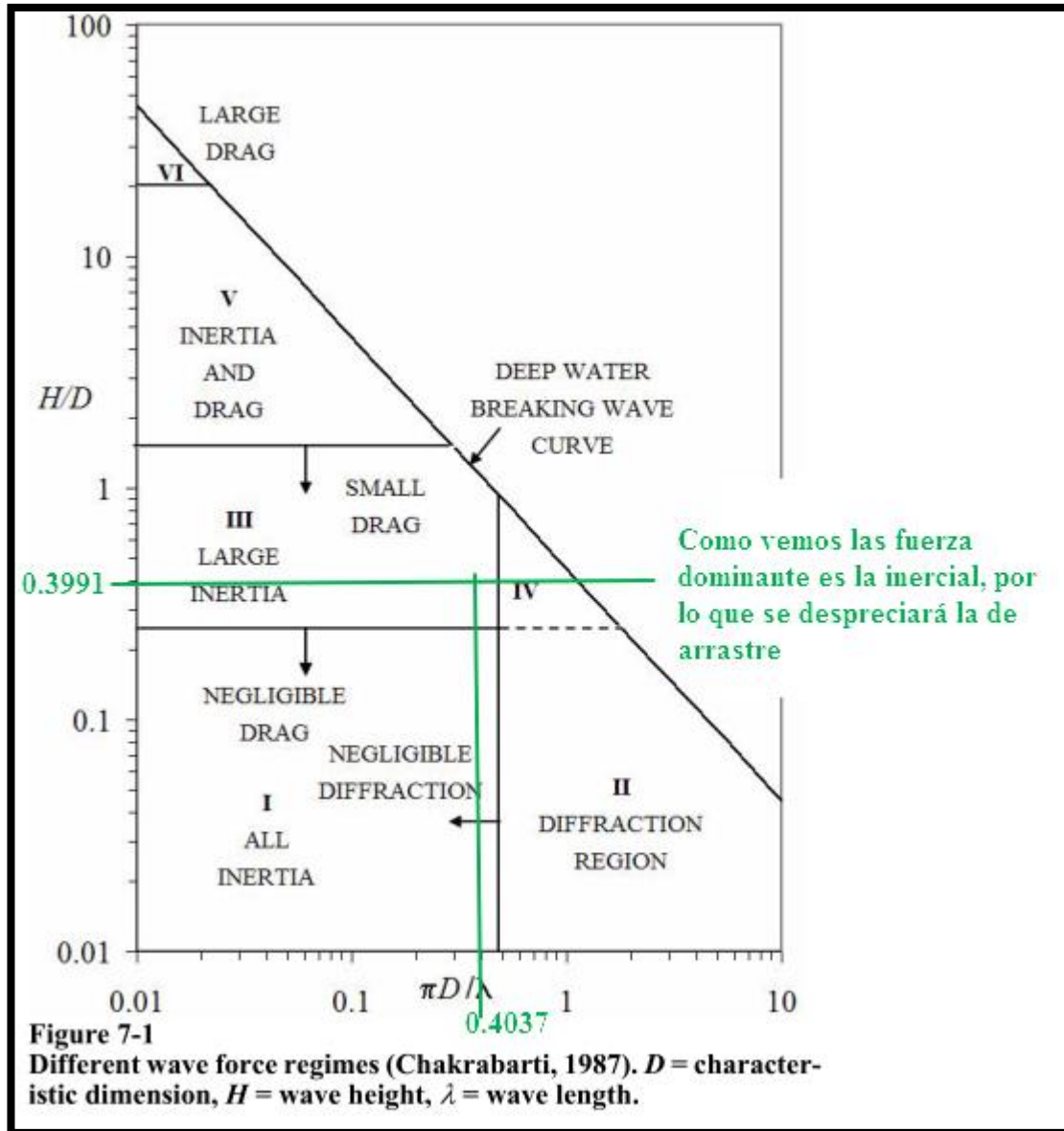


Figura 3.10 Determinamos la fuerza dominante

Entonces podemos determinar la fuerza mediante la integración de la presión aplicada en el cuerpo sumergido como:



$$F_{FK} = - \iint p * \hat{n} * ds = - \iint \rho * \frac{\partial \phi}{\partial t} * \hat{n} * ds = - \iint \rho * \frac{\partial \phi}{\partial t} * \hat{n} * dxdz$$

Por lo que haremos es el sumatorio de la fuerza alrededor de toda la superficie mojada, las cuatro caras verticales y la cara superior del pontón. Del DNV tenemos que:

Table 3-1 Gravity wave theory		
Parameter	Airy wave theory	
	General water depth	Deep water
Velocity potential, $\phi$	$\frac{\pi H}{kT} \frac{\cosh[k(z+d)]}{\sinh(kd)} \sin \theta$ $=$ $\frac{gH}{2\omega} \frac{\cosh[k(z+d)]}{\cosh(kd)} \sin \theta$	$\frac{\pi H}{kT} e^{kz} \sin \theta$ $=$ $\frac{gH}{2\omega} e^{kz} \sin \theta$

Notation:  $d$  = mean water depth,  $g$  = acceleration of gravity,  $H$  = trough-to-crest wave height,  $k = 2\pi/\lambda$  = wave number,  $\lambda$  = wave length,  $T$  = wave period;  $t$  = time;  $x$  = distance of propagation;  $z$  = distance from mean free surface positive upward;  $\theta = kx - \omega t = k(x - ct)$ ;  $\omega = 2\pi/T$  = angular wave frequency. Subscript  $l$  denotes linear small-amplitude theory.

Figura 3.11 Expresiones del potencial de velocidad extraídas del DNV-cp-205

$$\frac{\partial \phi}{\partial t} = (-w) * \frac{\pi * H}{k * T} * \frac{\cosh(k * (z + d))}{\sinh(k * d)} * \sin(k * x - w * t)$$

$$F_{FK} = (-\rho) * (-w) * \frac{\pi * H}{k * T} * \frac{\sin(k * x - w * t)}{\sinh(k * d)} * Lplancha * \int \cosh(k * (z + d)) dz$$

Esta expresión se evaluará numéricamente en tres puntos en su altura de la plancha.

En el programa primero se han modelado las caras verticales y después las cara superior e inferior. Si abrimos un while para recorrer todos los pasos de tiempo  $t$ , y un for después para recorrer todos los elementos  $m$ , que en este caso son sólo 4. Tenemos el siguiente código en donde se halla primero el vector director haciendo uso de 2 de las aristas de la cara, después hallamos el punto intermedio ya que vamos a hacer la integración primera de Simpson y por lo tanto ponemos un for que va de 1 a 3 para hallar los valores de la fuerza en los distintos puntos y los almacenamos.

Antes de poder proceder a realizar los cálculos hay que extraer los vértices de de las caras verticales en las que vamos a hacer la integración, utilizaremos para ello la herramienta SeaFEM.

Los valores obtenidos tras la extracción y almacenados en el Excel son:



Ponton inferior / Soft tank			
plancha popa estribor	x	y	z
popa_down	0	-14.175	-192.6
popa_up	0	-14.175	-185.4
estribor_down	14.175	0	-192.6
estribor_up	14.175	0	-185.4
plancha proa estribor	x	y	z
estribor_down	14.175	0	-192.6
estribor_up	14.175	0	-185.4
proa_down	0	14.175	-192.6
proa_up	0	14.175	-185.4
plancha proa babor	x	y	z
proa_down	0	14.175	-192.6
proa_up	0	14.175	-185.4
babor_down	-14.175	0	-192.6
babor_up	-14.175	0	-185.4
plancha popa babor	x	y	z
babor_down	-14.175	0	-192.6
babor_up	-14.175	0	-185.4
popa_down	0	-14.175	-192.6
popa_up	0	-14.175	-185.4

Figura 3.12 Coordenadas de los vértices de las 4 planchas verticales del pontón teniendo en cuenta la orientación dada en la Figura 1.6 y Figura 1.7

Entonces para las coordenadas dadas en 3.12, el código que nos resolverá el problema es:

```
load ponton_vertical.list

m=size(ponton_vertical,1)/4; %Números de planchas, se divide por 4 ya que cada plancha
tienen 4 puntos
%que definen sus vértices y que están almacenados
%en la matriz ponton
n_directores_ponton=zeros(m,4); %vector donde almacenaremos los vectores directores
%de las planchas

Txyz_ponton=zeros(4*m,3);
Txyz_ponton(:,:)=ponton_vertical(:,2:4);
FNxm_ponton=zeros(m,3);
FNx_ponton=zeros(m,1);
L_plancha_ponton=zeros(m,1);
F_ponton_vertical_x_rangot=zeros(nwt,m); %En este vector se almacenará los valores de las
fuerzas
```

```

%en los distintos puntos de tiempo t. en donde las
%columnas corresponden a distintos elementos plancha y las
%filas a los distinto valoroes de tiempo t

t=t_inicial;
while t <= t_final

    wt=w*t; %Hallamos el valor del producto w*t
    v=0; %variable auxiliar que se utilizará en el bucle para coger y saltar de una
        %plancha a otra
    for i=1:m
        %Extraemos los vertices de la planchas
        e1(1,:)=Txyz_ponton(1+v,:);
        e2(1,:)=Txyz_ponton(2+v,:);
        e3(1,:)=Txyz_ponton(3+v,:);
        e4(1,:)=Txyz_ponton(4+v,:);

        %Primero, haciendo uso de las vertices del ponton calcularemos los vectores
directores
        %c1 y c2 son vectores unitarios contenidos en el plano de las planchas
        c1=(e1-e2)/norm(e1-e2); %vector unitario axial al lado corto de la plancha
        c2=(e4-e2)/norm(e4-e2); %vector unitario axial al lado largo de la plancha
        n_director=cross(c1,c2);%vector normal unitario a la superficie

        n_directores_ponton(i,1)=i; %Almacenamos el elemento
        n_directores_ponton(i,2:4)=n_director(1,:); %Almacenamos sus vecto n
correspondiente
        %Hallamos la longitud de la plancha, del lado más largo para la
        %integral de froude krylov
        L_plancha=norm(e4-e2);
        L_plancha_ponton(i,1)=L_plancha;
        %Obtenemos las coordenadas del punto intermedio de la plancha,
        %punto medio vertical para la integración numérica. Para ello
        %haremos uso de los puntos up and down en popa, que tienen iguales
        %coordenadas x e y, pero distinta altura z.
        eintermedio=(e1+e2)/2; %Nos da el punto intermedio
        %Para poder hallar el angulo de desfase, kx-wt necesitamos hallar el valor
        %de x en los puntos de integración, que al ser una integración vertical,
        %se aproximará por la posción intermedia de x en la plancha, ya que al estar
        %inclinada es cierto que la ola no es igual en todos sus puntos.
        %Se hallara en el punto intermedio de la plancha longitudinalmente.
        x=(e1(1,1)+e3(1,1))/2;
        %Creamos un vector zzz donde almacenaremos los puntos verticales de
        %integración, y que se utilizará en el siguiente loop
        zzz=zeros(1,3);
        zzz(1,1)=e1(1,3); %Punto inicial, que es el punto mas bajo de la plancha,
integraremos
        zzz(1,2)=eintermedio(1,3); %hacia arriba la fuerza FNx en la plancha
        zzz(1,3)=e2(1,3); %Punto final, punto mas alto de la plancha

        for j=1:3 %Tres puntos de integración, j=1,2,3

            z=zzz(1,j);
            if d/L < 0.5 %Aguas transitorias, general water depth
                FNmi=(-densidad)*(-
w*pi*H/(k*T))*L_plancha*(cosh(k*(d+z))/sinh(k*d))*sin(k*x-wt);
            else %d/L > 0.5 1/2 aguas profundas
                FNmi=(-densidad)*(-w*pi*H/(k*T))*L_plancha*exp(k*z)*sin(k*x-wt);
            end
        end
    end
end

```

```

end
FNxm_ponton(i,j)=FNmi;%Almacenamos los valores de las fuerzas
                        %en los puntos de integración

end
%Para poder hacer la integración numérica, hay que hayar el
%espaciado
espaciado=abs(zzz(1,1)-zzz(1,2));
%Obtenemos la fuerza en x haciendo uso de la fuerza distribuida FNM,
%para ello se aplicará la primera regla de simpson
FNxi_ponton=espaciado*(1/3)*(1*FNxm_ponton(i,1)+4*FNxm_ponton(i,2)+1*FNxm_ponton(i,3));
FNx_ponton(i,1)=FNxi_ponton;
%Almacenamos las fuerzas de todos los elementos en las distintas
%columna, i, para el mismo valor de tiempo. Ponemos t-(t_inicial-1)
%de forma que si t_inicial no se parte de cero, no nos ponga las
%demas casillas como cero y al final los resultados
F_ponton_verticalx_rangot(t-(t_inicial-1),i)=FNx_ponton(i,1);

v=v+4;%Pasamos al siguiente set de vertices de la siguiente plancha
end
t=t+nt;%Pasamos al siguiente paso de tiempo
end

```

Ahora bien, para las plancha superior/inferior el código Matlab será programado de la siguiente manera:

$$F_{FK} = - \iint p * \hat{n} * ds = - \iint \rho * \frac{\partial \phi}{\partial t} * \hat{n} * ds = - \iint \rho * \frac{\partial \phi}{\partial t} * \hat{n} * dydx$$

$$F_{FK} = (-\rho) * (-w) * \frac{\pi * H}{k * T} * \frac{\cosh(k * (z + d))}{\sinh(k * d)} * (2 * y) * \int \sin(k * x - w * t) dx$$

El motivo por el que es  $2*y$ , es porque al resolver la integral numéricamente, necesitamos conocer la anchura del rombo en los puntos de integración numérica  $x$ . Y esta anchura es  $2*y$ , en donde  $y$ , es la coordenada  $y$  del borde de la plancha y que podremos obtener haciendo la media entre los valores conocidos. Esto se ve más claramente en la Figura siguiente:

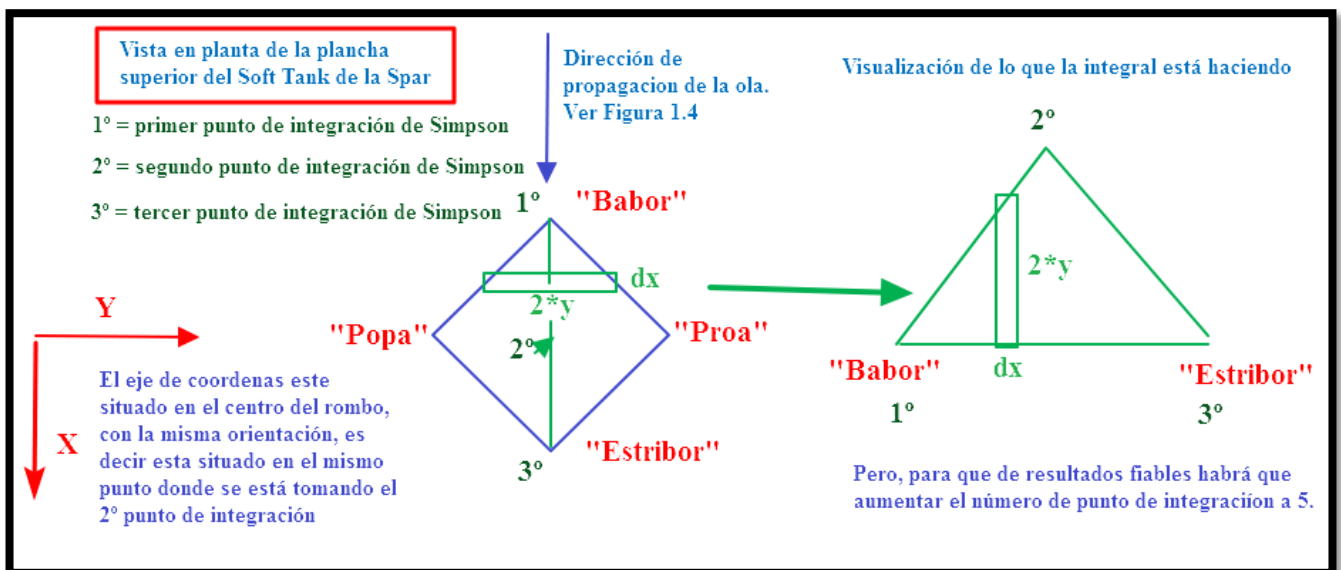


Figura 3.13 Breve explicación de cómo se está realizando la integración

Ahora bien, para que nos dé resultados fiables, habrá que realizar al menos 5 puntos de integración, este se ve claramente en un pequeño ejemplo en donde lo que buscamos es hallar el área de un rombo de lados 2. Por lo que su área sabemos que será  $4 u^2$ .

```
a=[-sqrt(2) 0 0];
ab=(a+b)/2;
b=[0 sqrt(2) 0];
bc=(b+c)/2;
c=[sqrt(2) 0 0];
A=zeros(1,5);

yyy=zeros(1,5);
yyy(1,1)=a(1,2);
yyy(1,2)=ab(1,2);
yyy(1,3)=b(1,2);
yyy(1,4)=bc(1,2);
yyy(1,5)=c(1,2);

for j=1:5
    aa=2*yyy(1,j)*1;
    A(1,j)=aa;
end

A_3puntos=(sqrt(2))*(1/3)*(1*A(1,1)+4*A(1,3)+1*A(1,5))
A_5puntos=(sqrt(2)/2)*(1/3)*(1*A(1,1)+4*A(1,2)+2*A(1,3)+4*A(1,4)+1*A(1,5))

error=((A_3puntos-A_5puntos)/A_5puntos)*100 %Con 3 puntos se comete un error de 33%
```

Si ejecutamos el código vemos que con 3 puntos obtenemos un 33% de error sobre el valor real del área, al obtener  $5.33 4 u^2$ , pero con 5 puntos obtenemos el valor exacto de  $4 u^2$ .

Las coordenadas de los vértices de las planchas horizontales del pontón son:

Ponton inferior / Soft tank			
plancha inferior	x	y	z
popa_down	0	-14.175	-192.6
estribor_down	14.175	0	-192.6
babor_down	-14.175	0	-192.6
proa_down	0	14.175	-192.6
plancha superior	x	y	z
popa_up	0	-14.175	-185.4
estribor_up	14.175	0	-185.4
babor_up	-14.175	0	-185.4
proa_up	0	14.175	-185.4

Figura 3.14 Vértices de las planchas superiores e inferiores del pontón

Ahora bien antes de pasar a mostrar la parte del código que nos hará los cálculos hay que mencionar como se deben introducir los datos de entrada en archivo `ponton_horizontal.list`. Para las caras horizontales podemos tener la cara superior o la cara inferior, en donde su vector normal es opuesto, para que el código sepa cuando invertir el signo al vector `n_director`, se tendrá que, aparte de añadir las coordenada de los vértices de las planchas inferior y superior, un valor 1 o 0 dependiendo si se trata de una plancha superior o de una plancha inferior. Esto se ve claramente en la siguiente figura:

nº elemento.	coordena x.	coordenada y.	coordenada z.	valor lógico
1	0	-14.175	-185.4	1
1	14.175	0	-185.4	1
1	-14.175	0	-185.4	1
1	0	14.175	-185.4	1
2	0	-14.175	-192.6	0
2	14.175	0	-192.6	0
2	-14.175	0	-192.6	0
2	0	14.175	-192.6	0

Al haber puesto para los cuatro primeros valores de la plancha horizontal un valor de 1, tenemos que es una plancha superior y su

Al haber puesto a la plancha horizontal un valor de 0, tenemos que es una plancha inferior y su normal será negativa

Figura 3.15 Esquema de cómo se deben introducir las coordenadas y los valores lógicos

Entonces sabiendo que se van a hacer 5 puntos de integración. Tras haber introducido los valores pertinentes de los vértices mostrados en la Figura 3.14, así como haber puesto su valor lógico correctamente según viene mostrado en la Figura 3.15. Podemos mostrar el código Matlab que nos resolverá el problema, tenemos:

```
load ponton_horizontal.list

m=size(ponton_horizontal,1)/4; %Números de planchas, se divide por 4 ya que cada plancha
tienen 4 puntos
%que definen sus vertices y que estan almacenados
%en la matriz ponton
valores_logicos_ponton_horizontal=zeros(m,1); %PARA saber si es una plancha horizontal
superior o inferior
%necesitamos decirselo al programa para que alterne el vector director de
%la superficie y esto se hara utilizando los valores lógicos introducidos
%en mismo archivo de los vertices, en la última columna, si tenemos 1,
%entonces se trata de una plancha superior y su vector normal apunta hacia
%arriba. si su valor es 0, es una plancha inferior y su vector normal
%apunta hacia abajo

valores_logicos_ponton_horizontal=ponton_horizontal(:,5); %Estan almacenados en la
columna 5
```

```
n_directores_ponton_horizontal=zeros(m,3); %vector donde almacenaremos los vectores
directores de las planchas
Txyz_ponton_horizontal=zeros(4*m,3);
Txyz_ponton_horizontal(:,:)=ponton_horizontal(:,2:4);
FNzm_ponton_horizontal=zeros(m,3);
FNz_ponton_horizontal=zeros(m,1);
F_ponton_horizontal_z_rangot=zeros(nwt,m);%En este vector se almacenará los valores de
las fuerzas

                %en los distintos puntos de tiempo t. en donde las
                %columnas corresponden a distintos elementos plancha y las
                %filas a los distintos valores de tiempo t

t=t_inicial;
while t <= t_final

    wt=w*t; %Hallamos el valor del producto w*t
    v=0;    %variable auxiliar que se utilizará en el bucle para coger y saltar de una
            %plancha a otra
    vv=1;   %Con esta variable saltaremos de 4 en 4 los coeficientes logicos
    for i=1:m

        %Extraemos los vertices de la planchas
        e1(1,:)=Txyz_ponton_horizontal(1+v,:);
        e2(1,:)=Txyz_ponton_horizontal(2+v,:);
        e3(1,:)=Txyz_ponton_horizontal(3+v,:);
        e4(1,:)=Txyz_ponton_horizontal(4+v,:);

        %Primero, haciendo uso de las vertices del ponton calcularemos los vectores
        directores
        %c1 y c2 son vectores unitarios contenidos en el plano de las planchas
        c1=(e2-e1)/norm(e2-e1); %vector unitario axial al lado corto de la plancha
        c2=(e3-e1)/norm(e3-e1); %vector unitario axial al lado largo de la plancha
        n_director=cross(c1,c2);%vector normal unitario a la superficie

        %Ahora comprobamos si la plancha es la superior o la inferior
        if valores_logicos_ponton_horizontal(vv,1)==0
            n_director=n_director*(-1);
        end

        n_directores_ponton_horizontal(i,:)=n_director(1,:);

        %Obtenemos las coordenadas del punto intermedio de la plancha, para
        %ello haremos uso de las coordenadas de los puntos extremos que se
        %encuentran en popa y en proa
        eintermedio=(e2+e3)/2;
        %AUNQUE para que nos de resultados fiables hay que aumentar el
        %número de puntos de integración. Esto se ve claramente en el
        %ejemplo añadido para el cálculo del área para un rombo.

        %Para poder hallar el angulo de desfase, k*x-w*t, necesitamos hallar el valor
        %de x, que al ser ahora la planchahorizontal, no nos vale considerar una
        %aproximación como si hicimos para la planchas verticales
        z=(e1(1,3)+e4(1,3))/2;
        %Extraemos las coordenadas de x, ya que ahora vamos integrar en la
        %dirección de x y no en la dirección vertical de z
        xxx=zeros(1,5);
        xxx(1,1)=e3(1,1); %Punto inicial,corresponde al punto x de babor, punto inicial
        xintermedio1=(e1(1,1)+e3(1,1))/2;
```

```

xxx(1,2)=xintermedio1; %coordenada entre el punto de popa x y el de babor x
xxx(1,3)=eintermedio(1,1); %Coordenada del punto intermedio
xintermedio2=(e1(1,1)+e2(1,1))/2;%coordenada x entre el punto x de estribor y el
de popa x
xxx(1,4)=xintermedio2;
xxx(1,5)=e2(1,1); %Punto final, corresponde al punto x de estribor, punto final

%Ya que estamos integrando un rombo, la doble integral dydx, la vamos a
%reducir a una sola integral dx, ya que ninguno de sus terminos
%depende de y, por lo tenemos que obtener los valores de la
%"altura" de la barra de integración, que va desde la "popa a proa"
%de la estructura offshore, y nosotros nos movemos integrando de
%"babor a estribor" en la direccion x. Ya que la ola viene de
%"costado" segun nuestra nomenclatura.
yyy=zeros(1,5);
yyy(1,1)=e3(1,2); %Punto inicial,corresponde al punto y de babor, punto inicial
y_para_xintermdio1=abs((e1(1,2)+e3(1,2))/2);
yyy(1,2)=y_para_xintermdio1; %coordenada entre el punto de popa y y el de babor
y
yyy(1,3)=abs(e1(1,2)); %Coordenada y, del el punto intermedio o central del
rombo
y_para_xintermedio2=abs((e1(1,2)+e2(1,2))/2);
yyy(1,4)=y_para_xintermedio2;%coordenada y entre el punto y de estribor y el de
popa y
yyy(1,5)=e2(1,2); %valor de y en la coordenada de estribor

for j=1:5 %Cinco puntos de integración, j=1,2,3,4,5 para la integración de
Simpson
    x=xxx(1,j);
    y=yyy(1,j);
    if d/L < 0.5 %Aguas transitorias,general water depth.
        %Se multiplica por 2 el valor de las y porque, este valor
        %va desde el punto de popa al eje de referencia, y nosotros
        %queremos que vaya hasta el otro extremo de proa del rombo,
        %de forma que estamos haciendo de la integral de golpe
        FNmi=(-densidad)*(-
w*pi*H/(k*T))*(2*y)*(cosh(k*(d+z))/sinh(k*d))*sin(k*x-wt);
    else %d/L > 0.5 1/2 aguas profundas
        FNmi=(-densidad)*(-w*pi*H/(k*T))*(2*y)*exp(k*z)*sin(k*x-wt);
    end
    FNzm_ponton_horizontal(i,j)=FNmi;%Almacenamos los valores de las fuerzas
    %en los puntos de integración
end

%Para poder hacer la integración numérica, hay que hayar el
%espaciado. Para ello hallamos la distancia entre el punto inicial
%más a popa y el punto intermedio
espaciado=abs(xxx(1,1)-xxx(1,2));
%Obtenemos la fuerza en x haciendo uso de la fuerza distribuida FNm,
%para ello se aplicará la primera regla de simpson
FNzi_ponton_horizontal=espaciado*(1/3)*(1*FNzm_ponton_horizontal(i,1)+...
4*FNzm_ponton_horizontal(i,2)+2*FNzm_ponton_horizontal(i,3)+...
4*FNzm_ponton_horizontal(i,4)+1*FNzm_ponton_horizontal(i,5));
FNz_ponton_horizontal(i,1)=FNzi_ponton_horizontal;
%Almacenamos las fuerzas de todos los elementos en las distintas
%columna, i, para el mismo valor de tiempo. Ponemos t-(t_inicial-1)
%de forma que si t_inicial no se parte de cero, no nos ponga las
%demas casillas como cero y al final los resultados

```

```
F_ponton_horizontalz_rangot(t-(t_inicial-1),i)=FNz_ponton_horizontal(i,1);

v=v+4;%Pasamos al siguiente set de vertices de la siguiente plancha
vv=vv+4;%Pasamos al siguiente set de valores lógicos
end
t=t+nt;%Pasamos al siguiente paso de tiempo
end
```

### 3.6° Programación del cálculo de las fuerzas sobre las planchas de arfada

La resolución de las planchas de arfada, tanto las fuerzas sobre las planchas verticales como las horizontales se hará exactamente igual que en el apartado anterior 3.5 donde se hizo la resolución para el pontón. El código de Matlab será exactamente igual, pero alterando los nombres, y los datos de entrada. Los vértices de las 16 caras verticales, 4 caras por bloque arfada de arfada, y las 8 planchas horizontales de arfada, 4 superiores, y 4 inferiores, 1 de cada por bloque de arfada.

Al ser las dimensiones iguales que las del pontón (a excepción de la altura), se sabe que podremos utilizar las mismas fórmulas de froude krylov para obtener las fuerzas en x y las fuerzas en z.

Primera plancha antiarfada			
plancha popa estribor	x	y	z
popa_down	0	-14.175	-77.4
popa_up	0	-14.175	-76.5
estribor_down	14.175	0	-77.4
estribor_up	14.175	0	-76.5
plancha proa estribor	x	y	z
estribor_down	14.175	0	-77.4
estribor_up	14.175	0	-76.5
proa_down	0	14.175	-77.4
proa_up	0	14.175	-76.5
plancha proa babor	x	y	z
proa_down	0	14.175	-77.4
proa_up	0	14.175	-76.5
babor_down	-14.175	0	-77.4
babor_up	-14.175	0	-76.5
plancha popa babor	x	y	z
babor_down	-14.175	0	-77.4
babor_up	-14.175	0	-76.5
popa_down	0	-14.175	-77.4
popa_up	0	-14.175	-76.5

Figura 3.16 Coordenadas de los vértices de las planchas verticales. Primera plancha antiarfada



Primera plancha antiarfada			
plancha inferior	x	y	z
popa_down	0	-14.175	-77.4
estribor_down	14.175	0	-77.4
babor_down	-14.175	0	-77.4
proa_down	0	14.175	-77.4
plancha superior	x	y	z
popa_up	0	-14.175	-76.5
estribor_up	14.175	0	-76.5
babor_up	-14.175	0	-76.5
proa_up	0	14.175	-76.5

Figura 3.17 Coordenadas de los vértices de las planchas horizontales. Primera plancha antiarfada

Segunda plancha antiarfada			
plancha popa estribor	x	y	z
popa_down	0	-14.175	-104.4
popa_up	0	-14.175	-103.5
estribor_down	14.175	0	-104.4
estribor_up	14.175	0	-103.5
plancha proa estribor	x	y	z
estribor_down	14.175	0	-104.4
estribor_up	14.175	0	-103.5
proa_down	0	14.175	-104.4
proa_up	0	14.175	-103.5
plancha proa babor	x	y	z
proa_down	0	14.175	-104.4
proa_up	0	14.175	-103.5
babor_down	-14.175	0	-104.4
babor_up	-14.175	0	-103.5
plancha popa babor	x	y	z
babor_down	-14.175	0	-104.4
babor_up	-14.175	0	-103.5
popa_down	0	-14.175	-104.4
popa_up	0	-14.175	-103.5

Figura 3.18 Coordenadas de los vértices de las planchas verticales. Segunda plancha antiarfada

Segunda plancha antiarfada			
plancha inferior	x	y	z
popa_down	0	-14.175	-104.4
estribor_down	14.175	0	-104.4
babor_down	-14.175	0	-104.4
proa_down	0	14.175	-104.4
plancha superior	x	y	z
popa_up	0	-14.175	-103.5
estribor_up	14.175	0	-103.5
babor_up	-14.175	0	-103.5
proa_up	0	14.175	-103.5

Figura 3.19 Coordenadas de los vértices de las planchas horizontales. Segunda plancha antiarfada

Tercera plancha antiarfada			
plancha popa estribor	x	y	z
popa_down	0	-14.175	-131.4
popa_up	0	-14.175	-130.5
estribor_down	14.175	0	-131.4
estribor_up	14.175	0	-130.5
plancha proa estribor	x	y	z
estribor_down	14.175	0	-131.4
estribor_up	14.175	0	-130.5
proa_down	0	14.175	-131.4
proa_up	0	14.175	-130.5
plancha proa babor	x	y	z
proa_down	0	14.175	-131.4
proa_up	0	14.175	-130.5
babor_down	-14.175	0	-131.4
babor_up	-14.175	0	-130.5
plancha popa babor	x	y	z
babor_down	-14.175	0	-131.4
babor_up	-14.175	0	-130.5
popa_down	0	-14.175	-131.4
popa_up	0	-14.175	-130.5

Figura 3.20 Coordenadas de los vértices de las planchas verticales. Tercera plancha antiarfada

Tercera plancha antiarfada			
plancha inferior	x	y	z
popa_down	0	-14.175	-131.4
estribor_down	14.175	0	-131.4
babor_down	-14.175	0	-131.4
proa_down	0	14.175	-131.4
plancha superior	x	y	z
popa_up	0	-14.175	-130.5
estribor_up	14.175	0	-130.5
babor_up	-14.175	0	-130.5
proa_up	0	14.175	-130.5

Figura 3.21 Coordenadas de los vértices de las planchas horizontales. Tercera plancha antiarfada

Cuarta plancha antiarfada			
plancha popa estribor	x	y	z
popa_down	0	-14.175	-158.4
popa_up	0	-14.175	-157.5
estribor_down	14.175	0	-158.4
estribor_up	14.175	0	-157.5
plancha proa estribor	x	y	z
estribor_down	14.175	0	-158.4
estribor_up	14.175	0	-157.5
proa_down	0	14.175	-158.4
proa_up	0	14.175	-157.5
plancha proa babor	x	y	z
proa_down	0	14.175	-158.4
proa_up	0	14.175	-157.5
babor_down	-14.175	0	-158.4
babor_up	-14.175	0	-157.5
plancha popa babor	x	y	z
babor_down	-14.175	0	-158.4
babor_up	-14.175	0	-157.5
popa_down	0	-14.175	-158.4
popa_up	0	-14.175	-157.5

Figura 3.22 Coordenadas de los vértices de las planchas verticales. Cuarta plancha antiarfada

Cuarta plancha antiarfada			
plancha inferior	x	y	z
popa_down	0	-14.175	-158.4
estribor_down	14.175	0	-158.4
babor_down	-14.175	0	-158.4
proa_down	0	14.175	-158.4
plancha superior	x	y	z
popa_up	0	-14.175	-157.5
estribor_up	14.175	0	-157.5
babor_up	-14.175	0	-157.5
proa_up	0	14.175	-157.5

Figura 3.23 Coordenadas de los vértices de las planchas horizontales. Cuarta plancha antiarfada

Todas las coordenadas anteriores habrá que meterlas correctamente en sus .list correspondientes y de la misma manera que con el pontón.

### 3.7º Programación del cálculo de las fuerzas sobre el Hard Tank o cilindro

Las dimensiones del hard tank, coordenada inicial y coordenada final así como su diámetro ya se introdujeron en el archivo datos.list y que se muestra en la Figura 3.3.

Lo primero que hay que hacer antes de programar nada es ver que fuerzas son las dominantes, por lo que haciendo uso del diámetro del hard tank,  $D=28.35$ , hacemos lo mismo que se hizo con el soft tank en la Figura 3.10.

Tenemos que  $H/D=8/28.35 = 0.2822$ , y que  $\pi D/L=\pi \cdot 28.35/156 = 0.57$ . Además si  $L/6 = 26$  que es menor que  $D$ , por lo estamos tratando con una estructura compacta.

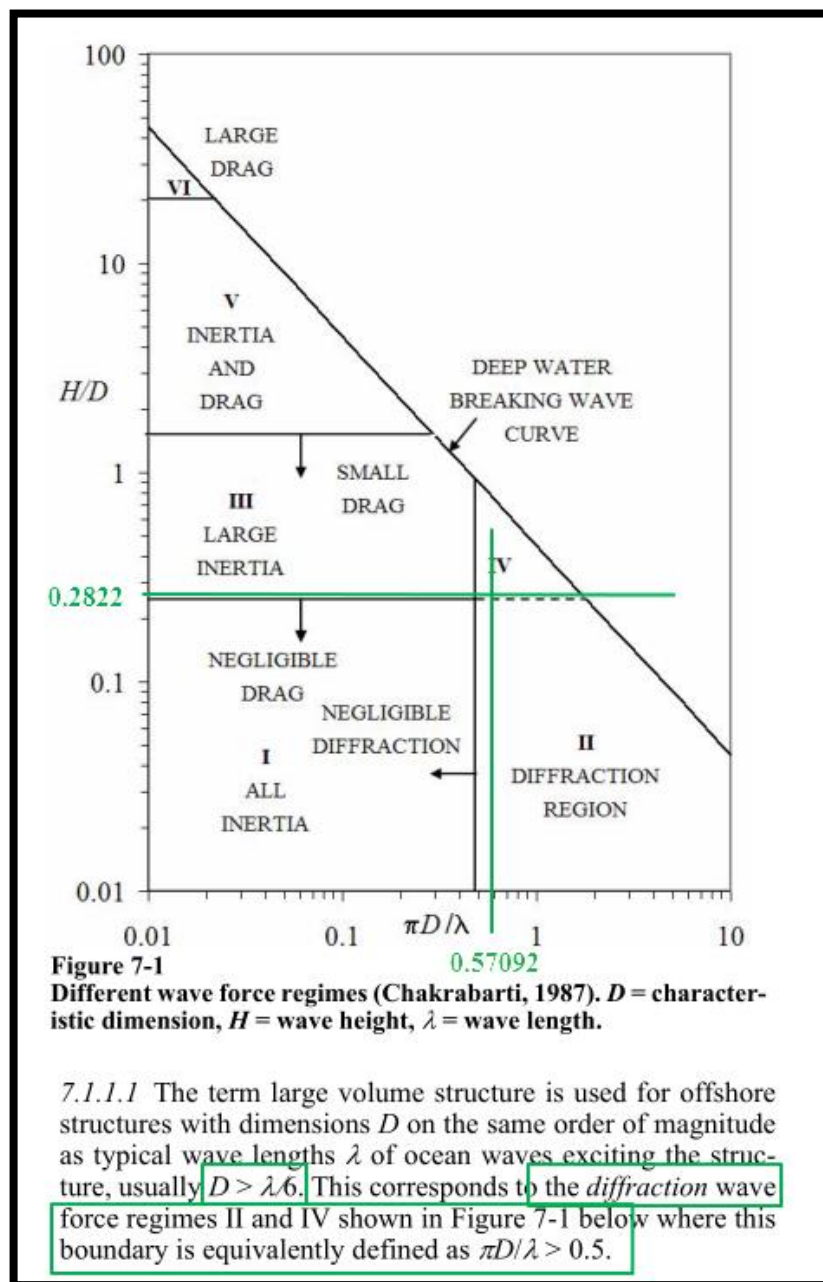


Figura 3.24 Comprobación de la teoría a aplicar para el hard tank.

Como vemos, el tamaño del cilindro es lo suficientemente grande en comparación a la longitud de la ola como para empezar a difractar, por lo que las teorías utilizadas hasta ahora ya no son de aplicación.

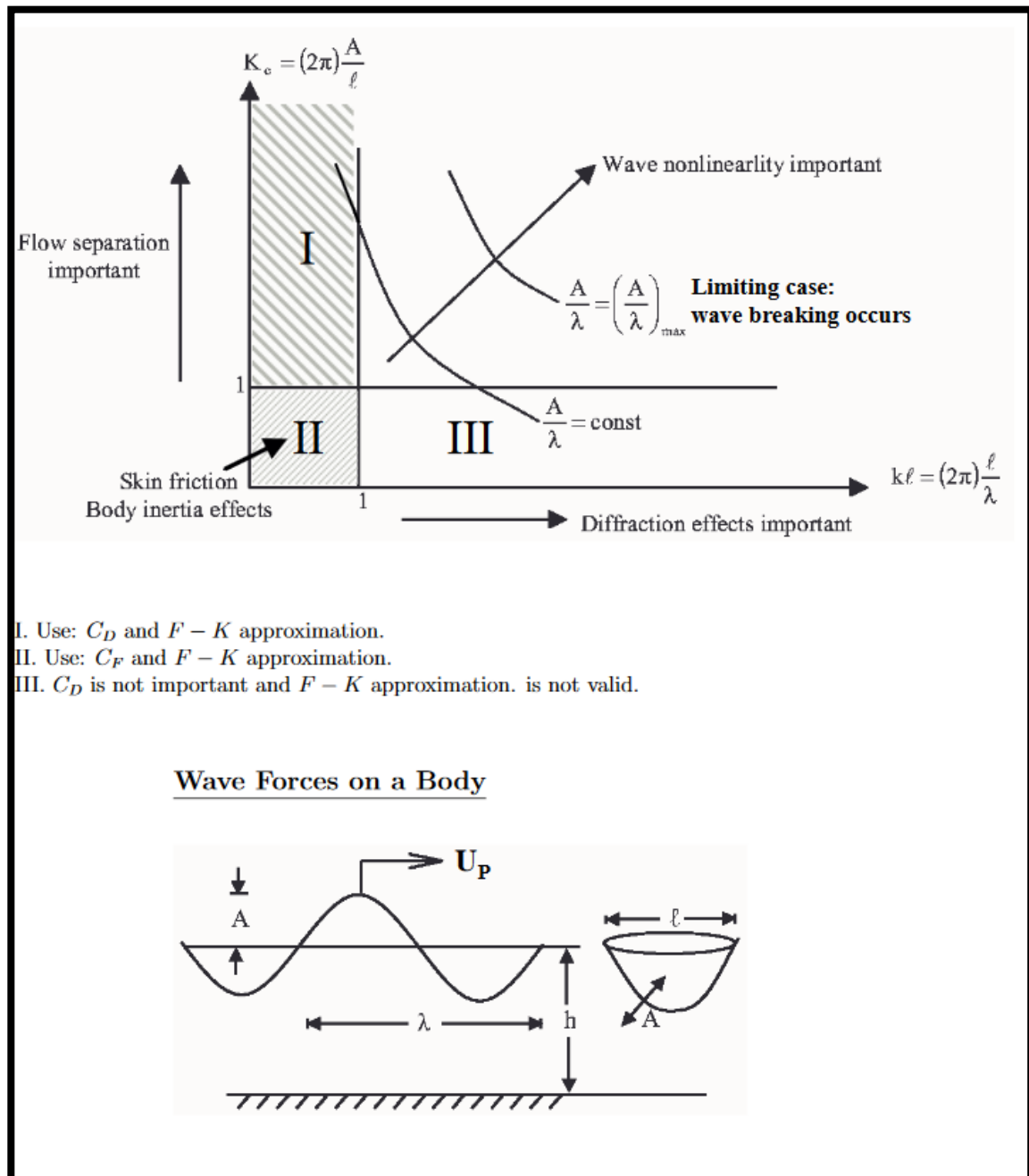


Figura 3.25 Resumen de aplicabilidad de las distintas teorías según el efecto de la difracción y de  $K_c$

Por ello para el cilindro se tendrá que utilizar una teoría de la difracción, al no ser ya validas la aproximación de froude krylov, y al ser el arrastre insignificante en comparación. Las fórmulas que se utilizarán se muestran a continuación:

The following assumptions are made. The fluid is frictionless and moving irrotationally. The ratio of the height of the waves to their length is sufficiently small so that all quantities involving the parameter  $(H/L)$  in the second or higher powers may be neglected without sensible error, thus giving rise to the so-called linear theory. The waves are incident on a vertical circular cylinder which extends to the bottom. The depth of the water is  $d$ , finite.

Letting  $z$  be the distance along the cylinder, in the direction of its axis, with positive direction upward from the still-water level, the  $x$ -component of the force on the cylinder per unit length in the  $z$ -direction and at depth  $z$ , is,

$$F_z = \frac{2 \rho g H}{k} \frac{\cosh k(d+z)}{\cosh kd} A\left(\frac{D}{L}\right) \cos(\sigma t - \alpha) \quad (28)$$

where

$$\tan \alpha = \frac{J_1' \left( \pi \frac{D}{L} \right)}{Y_1' \left( \pi \frac{D}{L} \right)}$$

$$J'(kr) = J_0(kr) - \frac{1}{kr} J_1(kr)$$

$$Y'(kr) = Y_0(kr) - \frac{1}{kr} Y_1(kr)$$

$$A\left(\frac{D}{L}\right) = \frac{1}{\sqrt{J_1'^2 \left( \pi \frac{D}{L} \right) + Y_1'^2 \left( \pi \frac{D}{L} \right)}}$$

Figura 3.26 Formulación de la teoría de la difracción para un cilindro compacto.

La fórmula anterior nos da la fuerza distribuida a lo largo de la altura (eje  $z$ ) del cilindro, por ello tendremos que integrarla. Para ello resolveremos la integral numéricamente teniendo en cuenta que para que el resultado reduzca el error cometido se tomarán un número alto de puntos de integración, para ello se ha decidido elegir  $m=21$  puntos de integración. Esto se ve claramente si se compara la solución, con respecto a la solución que se obtendría al aplicarla a un cilindro que es fijo al fondo, de profundidad  $h$ .

Lo que interesa es conocer la fuerza por unidad de longitud sobre el cilindro. Integrando la fuerza por unidad de longitud a lo largo del puntal del cilindro se obtiene que la fuerza total horizontal sobre el cilindro se puede escribir,

$$F_x = -\rho g \pi R^2 \xi \tanh kh \left[ \frac{4}{\pi(kR)^2} \frac{\cos(\omega t - \delta_1)}{\sqrt{J_1'^2(kR) + Y_1'^2(kR)}} \right] \quad (4.49)$$

Siendo el desfase  $\delta_1$  estimado como,

$$\delta_1 = \arctan \frac{J_1'(kr)}{Y_1'(kr)}$$

Figura 3.27 Solución resultado de la integración

La programación de la fórmula dada en la Figura 3.27 se muestra a continuación:

## Calculamos ahora la fuerza ejercida en el cilindro, hard tank sumergido según la teoría de la difracción

```
%La fuerza en y, y en z son cero, ya que la ola se propaga en x, y en la
%direccion transversal y, se compensa.
m=21; %Numero de puntos de integracion para la primera regla de simpson
zsup=DATOS1(12,1); %Coordenada superior del cilindro
zinf=DATOS1(13,1); %Coordenada inferior del cilindro
espaciado=abs((zsup-zinf)/m); %Hallamos el espaciado de la integraci3n
D_cilindro=DATOS1(14,1); %Extraemos el diametro del cilindro

kr=k*D_cilindro/2; %Hallamos el producto kr
FNx_cilindro=0; %Vector final de la fuerza en la direcci3n x
FN_cilindro=zeros(3,1); %Vector fuerzas en las 3 direcciones x,y,z

%Calculamos los siguientes valores constante de las funciones de Bessel
J0=besselj(0,kr);
J1=besselj(1,kr);
Y0=bessely(0,kr);
Y1=bessely(1,kr);
J_prima=J0-(1/kr)*J1;
Y_prima=Y0-(1/kr)*Y1;
A=1/sqrt(J_prima^2 + Y_prima^2);
alpha=atan(J_prima/Y_prima);
F_cilindrox_rangot=zeros(nwt,1); %En este vector se almacenar3 los valores de las fuerzas
%en los distintos puntos de tiempo t. en donde las
%columnas corresponden a distintos elementos cilindro y las
%filas a los distinto valoroes de tiempo t

t=t_inicial;
while t <= t_final

    wt=w*t; %Hallamos el valor del producto w*t

    z=0; %Reseteamos el valor de z para cada paso de tiempo
    Sumatorio=0; %Reseteamos tambien el sumatorio
    for i=1:m %Recorremos todos los puntos de integraci3n
        FNxi=(2*densidad*g*H/k)*(cosh(k*(d+z))/cosh(k*d))*A*cos(wt-alpha);
        %Multiplicamos la fuerza por el integrando correspondiente
        if i==1 || i==m
            FNxi=FNxi*1;
        elseif rem(i,2)==0 %Numero par
            FNxi=FNxi*4;
        elseif rem(i,2)==1 %Numero impar
            FNxi=FNxi*2;
        end
        Sumatorio=Sumatorio+FNxi;
        z=z-espaciado; %Pasamos al siguiente punto de integraci3n
    end

    FNx_cilindro=(1/3)*espaciado*Sumatorio;

    F_cilindrox_rangot(t-(t_inicial-1),1)=FNx_cilindro(1,1); %Solo tenemos 1 cilindro
    por lo que (1,1)
    t=t+nt; %Pasamos al siguiente paso de tiempo
end
```



## 4º Comparación de los resultados obtenidos entre

Antes de proceder a mostrar los cálculos obtenidos se va a mostrar un índice del significado y que almacenan cada una de las variables finales, de la versión definitiva:

Name ▲	Value	
anormal_total	1452x6 double	matriz que almacena las aceleraciones en los distintos puntos de integración, donde la primera columna corresponde al elemento, la segunda al punto de tiempo t, la tercera al punto de integración en el elemento j, y las 3 ultimas los valores ax, ay y az
d Profundidad, m	250	
DATOS Archivo de datos cargado	14x2 table	
densidad Densidad, kg/m <sup>3</sup>	1025	
F_cilindrox_rangot	11x1 double	En estas matrices se almacenan en las columns los valores de las fuerzas en x o z, de los distintos elementos m que se tengan, y en donde las filas corresponden a los distintos valores de tiempo t, empezando en t_inicial y acabando en t_final y siendo estos valores introducidos por el usuario en el archivo datos.inp
F_plancha_arfada_horizontalz_rangot	11x8 double	
F_plancha_arfada_verticalx_rangot	11x16 double	
F_ponton_horizontalz_rangot	11x2 double	
F_ponton_verticalx_rangot	11x4 double	
F_refuerzosx_rangot	11x44 double	
F_refuerzosz_rangot	11x44 double	
Ftotalx	11x1 double	
Ftotalx_cilindrox_rangot	11x1 double	Fuerza total en x para la Spar completa
Ftotalx_plancha_arfada_verticalx_rangot	11x1 double	Valores de las fuerzas totales, en x, considerando el valor aportado por cada elemento en esa dirección, manteniendo ún los subgrupos.
Ftotalx_ponton_verticalx_rangot	11x1 double	
Ftotalx_refuerzos_rangot	11x1 double	
Ftotalz	11x1 double	Fuerza total en z para la Spar completa
Ftotalz_plancha_arfada_horizontalz_ra...	11x1 double	Valores de las fuerzas totales, en z, considerando el valor aportado por cada elemento en esa dirección, manteniendo aún los subgrupos.
Ftotalz_ponton_horizontalz_rangot	11x1 double	
Ftotalz_refuerzos_rangot	11x1 double	
H Altura de ola, m	8	
k Número de onda	0.0403	
L Longitud entre olas, m	156	
n_axial_refuerzos	44x4 double	Aquí se almacenan los vectores directores axiales, por ejemplo en los refuerzos, o los normales a la superficie para los demás.
n_directores_plancha_arfada_horizontal	8x4 double	
n_directores_plancha_arfada_vertical	16x4 double	
n_directores_ponton	4x4 double	
n_directores_ponton_horizontal	[1 0 0 1.0000; 2 0 0 -1....	
nodos	44x8 double	Valores leídos de los distintos archivos .list
nt Incrementos de tiempo, s	1	
plancha_arfada_horizontal	32x5 double	
plancha_arfada_vertical	64x4 double	
ponton_horizontal	8x5 double	
ponton_vertical	16x4 double	
T Periodo, s	10	
t_final Punto inicial de tiempo, s	150	
t_inicial Punto final de tiempo, s	140	matriz que almacena las velocidades en los distintos puntos de integración, donde la primera columna corresponde al elemento, la segunda al punto de tiempo t, la tercera al punto de integración en el elemento j, y las 3 ultimas los valores ax, ay y az
Vnormal_total	1452x6 double	
w Velocidad angular, rad/s	0.6283	

Figura 4.1 Pequeña guía resumen de que contiene cada variable tras ejecutar el programa

Ahora algo que no se ha añadido en los códigos anteriores, es el sumatorio de las fuerzas que cada subgrupo (refuerzos, pontón, plancha arfada, cilindro) tienen tanto en la dirección x como en la dirección z, debido a la adición de sus componentes. Estos sumatorios de fuerzas se han realizado al final de cada apartado antes de pasar al siguiente, de forma que podamos realizar una representación gráfica individual de las fuerzas. Estas representaciones se muestran a continuación.



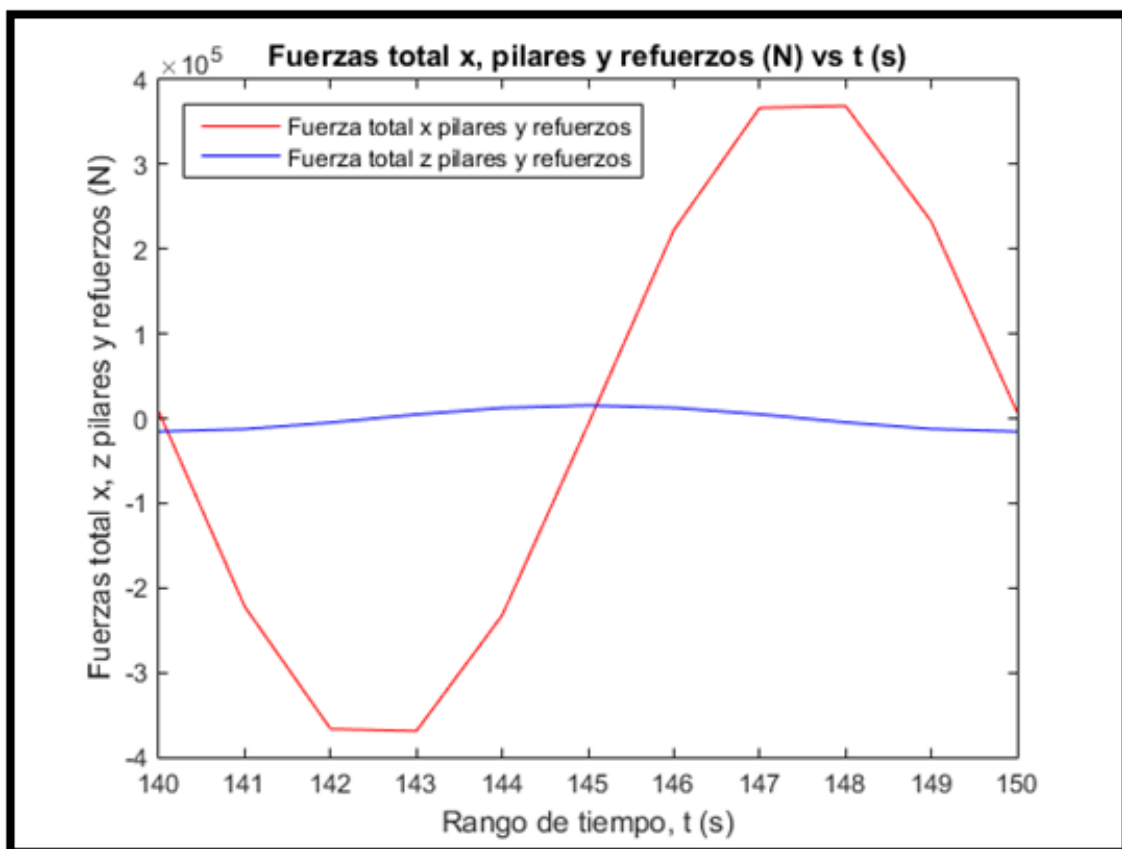


Figura 4.2 Fuerzas x y Fuerzas z para los pilares y refuerzos

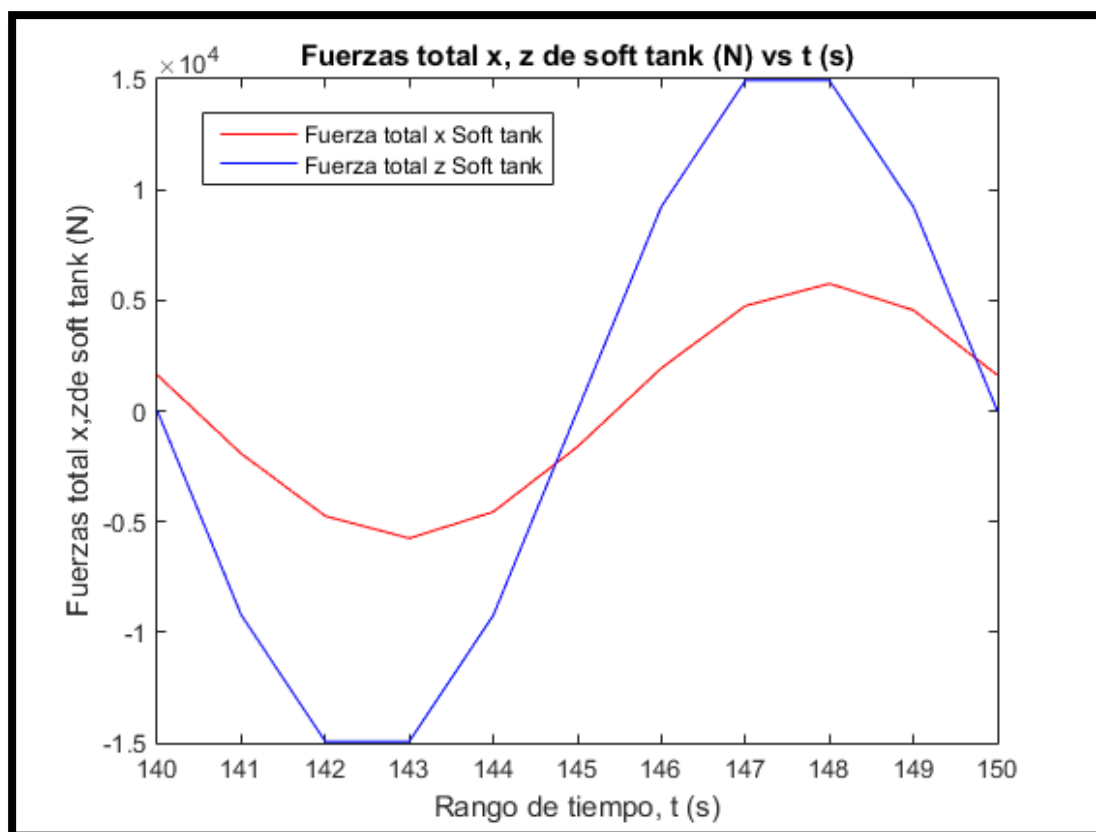


Figura 4.3 Fuerzas x y Fuerzas z para el Soft Tank

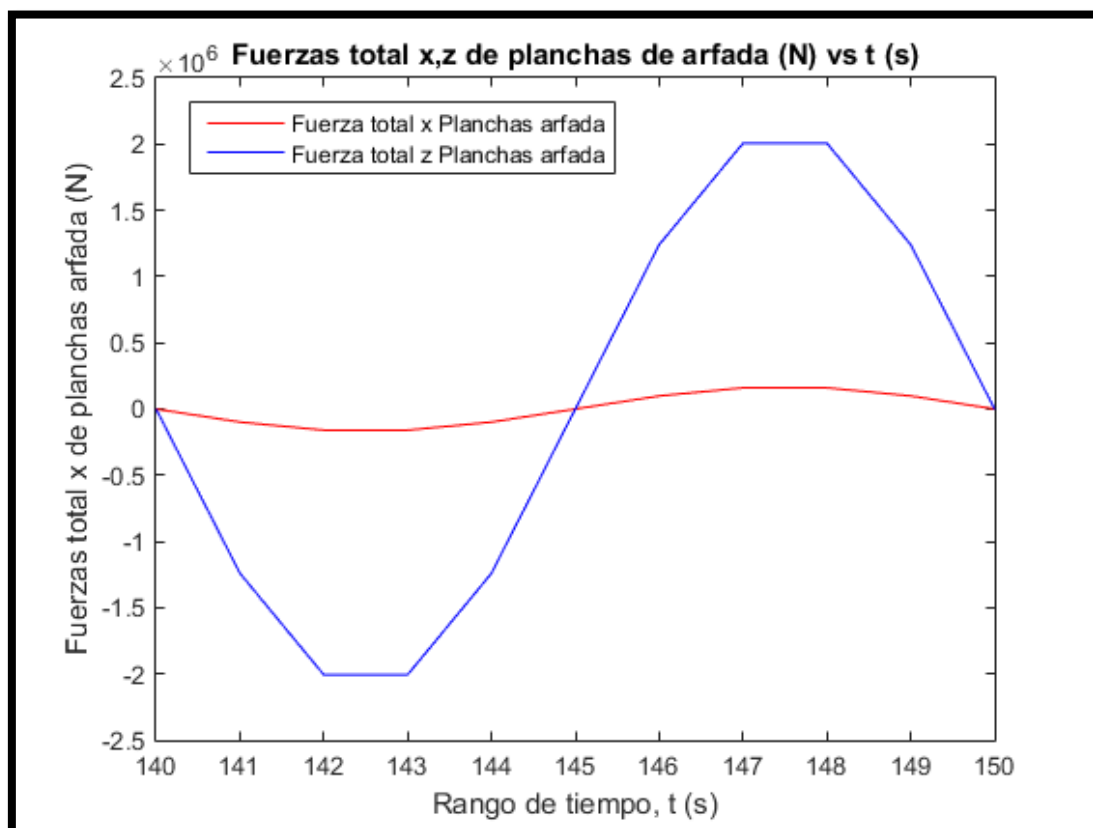


Figura 4.4 Fuerzas x y Fuerzas z para las planchas de arfada

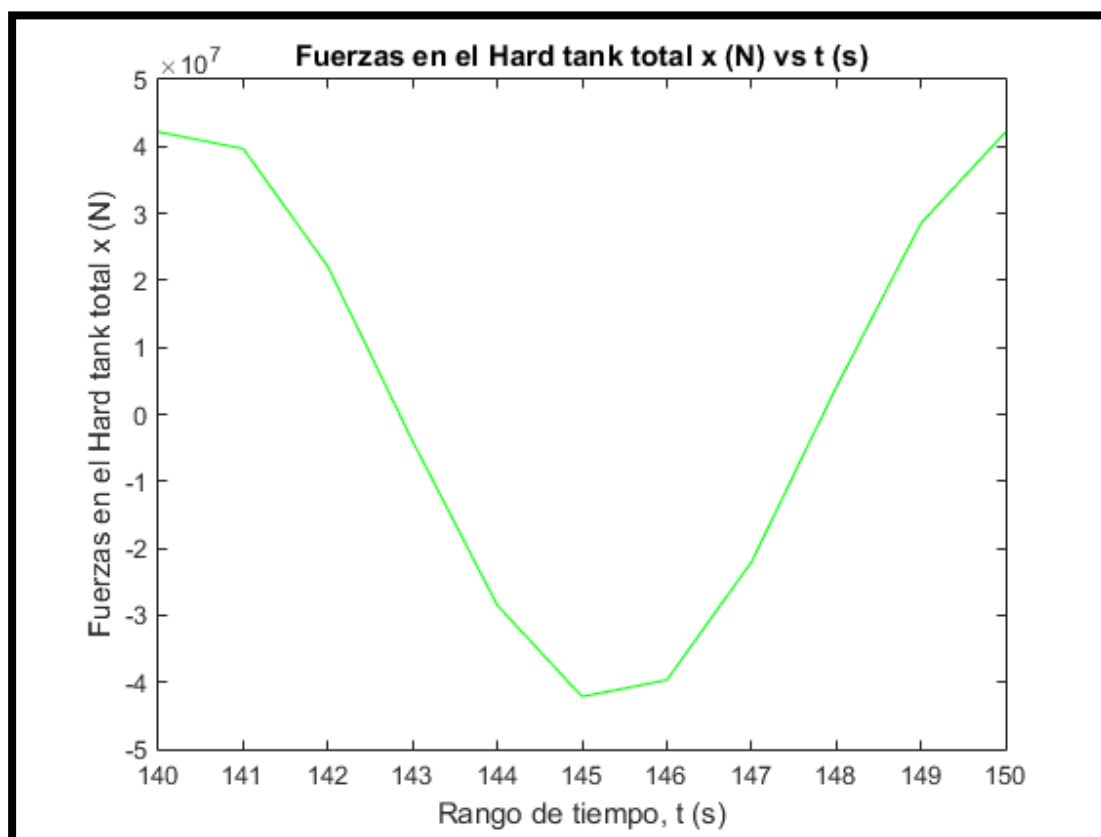


Figura 4.5 Fuerzas x para el Hard Tank

Si analizamos las ordenes de magnitud de todas las fuerzas  $x$  e  $z$ , en su propio subgrupo, para una ola monocromática que se propaga en la dirección  $x$ , se tienen valores de fuerzas  $x$  mayores que las fuerzas  $z$  en los refuerzos y los pilares, con un orden de magnitud de diferencia.

Pero para el pontón/Soft tank y las planchas de arfada se han obtenido valores de fuerzas en  $z$  son mucho mayores que sus fuerzas en  $x$ , esto se debe principalmente a que la doble integral de la superficie mojada hace que aquellas superficies de mayor área aporten más a la fuerza. Por ello dado a que las superficies horizontales del Soft Tank y de las planchas de arfada tienen áreas mucho mayores que las superficies verticales, (sobre todo en el caso de las planchas de arfada) hacen que se obtengan fuerzas en  $z$  mayores.

Ahora bien, como se ve en la Figura 4.5, la fuerza dominante de todas las anteriores es la fuerza ejercida por las olas sobre el Hard Tank. Esto se ve claramente si se comparan los órdenes de magnitud de las fuerzas. Esto es de esperar, por el hecho de que el hard tank es el elemento de la Spar que más próxima a la superficie está, por lo que recibe el impacto directo de las olas, y es donde los esfuerzos de las mismas es máximo.

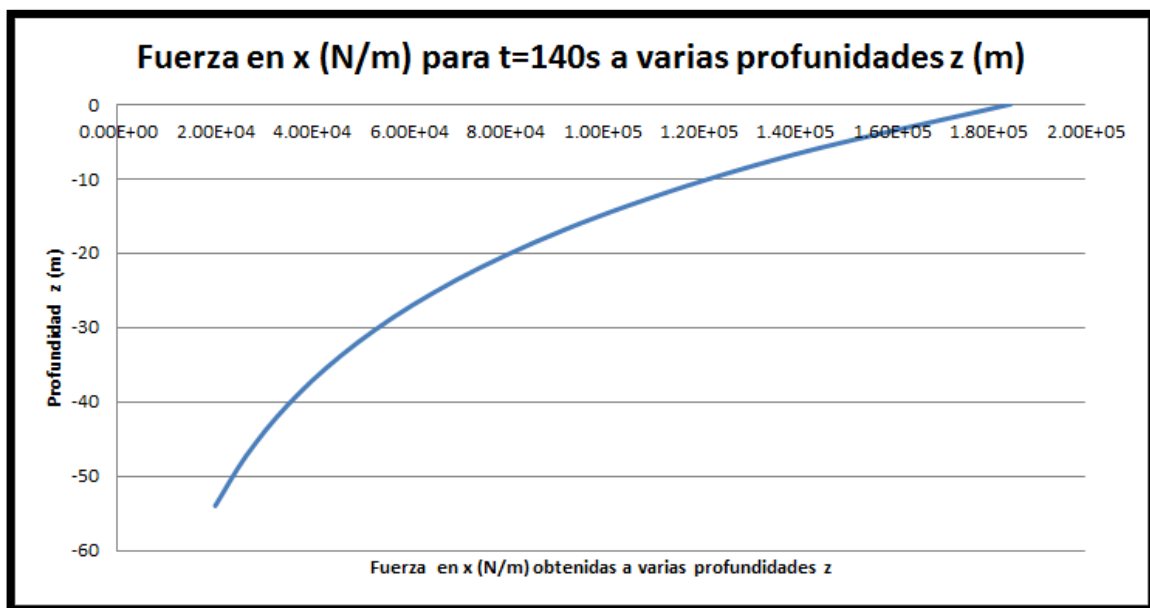


Figura 4.6 Fuerzas en  $x$  (N/m) para  $t=140s$  a varias profundidades  $z$  (m)

Ahora realizando el sumatorio de todas las componentes de fuerzas  $x$  se puede obtener la fuerza total en  $x$  que la Spar está sometida. Y si se realiza lo mismo para las fuerzas  $z$ , se puede obtener la fuerza  $z$  total a la que la Spar está sometida. Ambas gráficas se muestran a continuación.

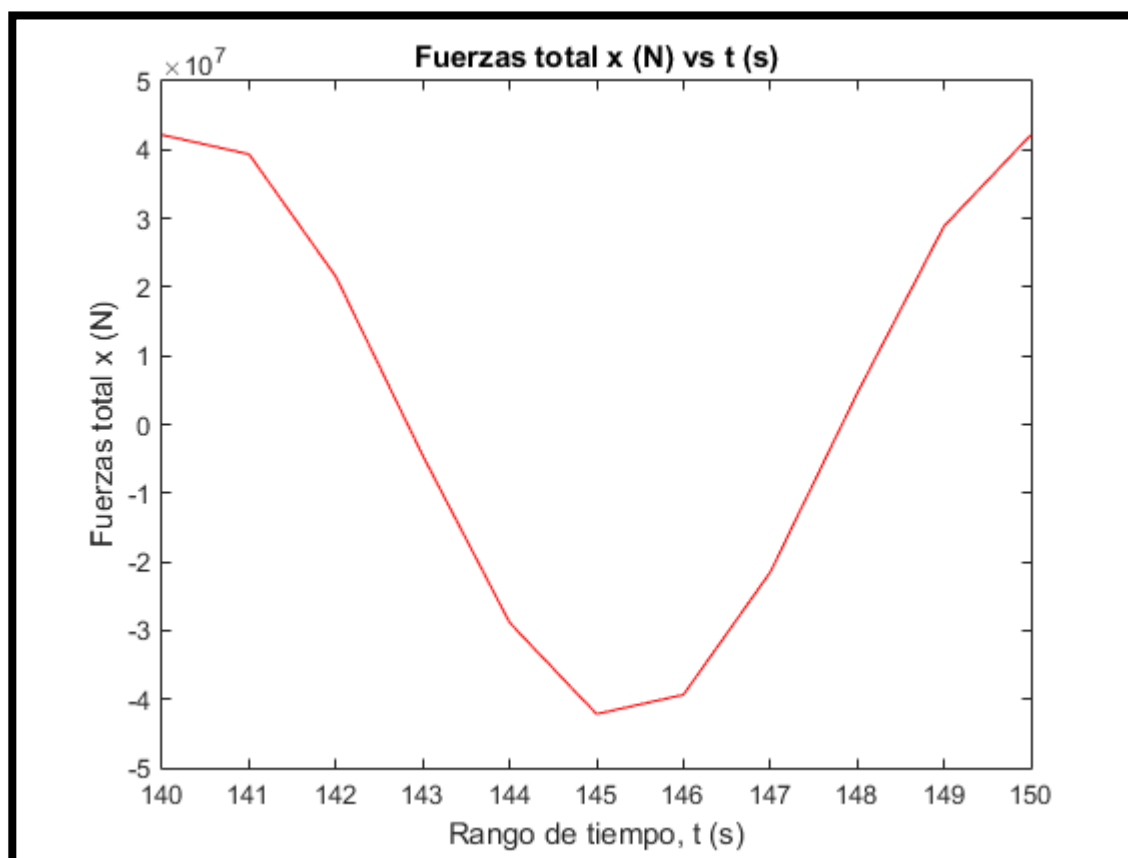


Figura 4.7 Fuerza total en x (N) actuante en la Spar

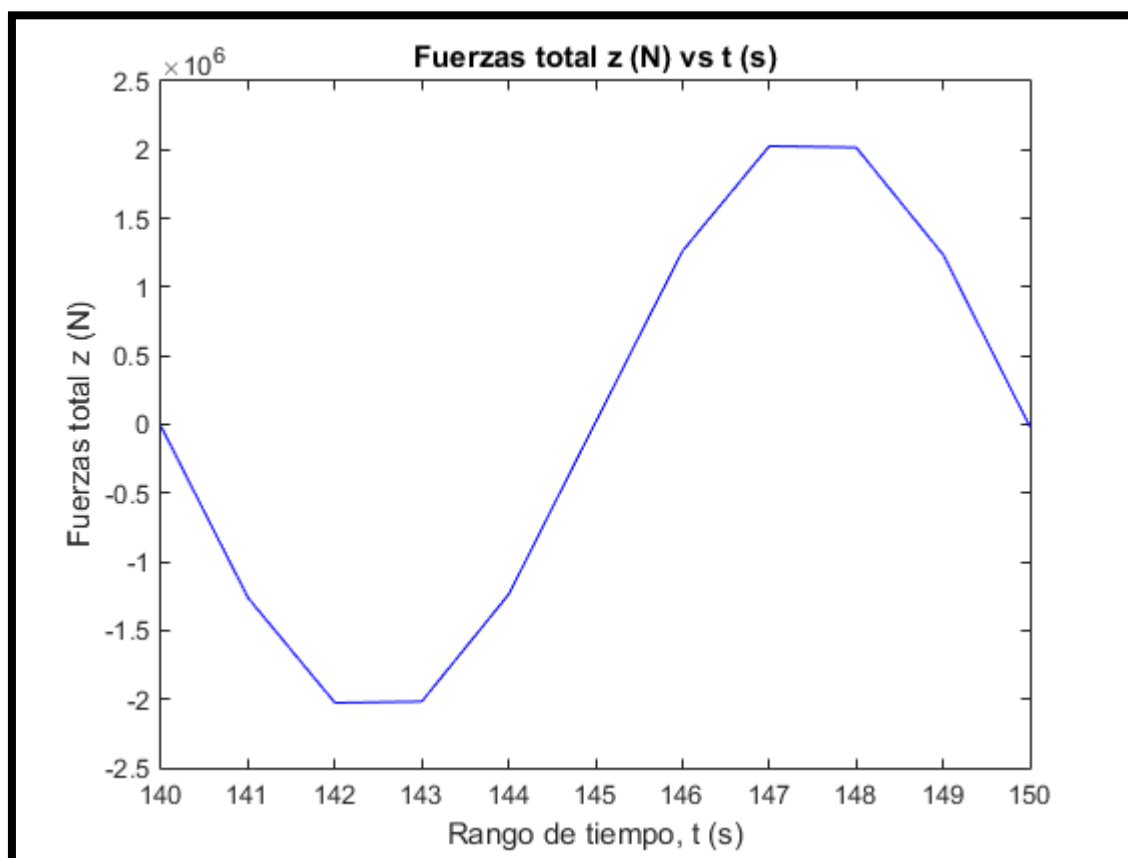


Figura 4.8 Fuerza total en z (N) actuante en la Spar

Las fuerzas totales  $x$ ,  $z$  están desfasadas 2 segundos, y tal y como ya se mencionó con anterioridad las fuerzas en  $x$  son mayores, y la sinusoidal depende casi exclusivamente de la sinusoidal del Hard Tank, ya que es la componente dominante en el sumatorio de fuerzas, al ser varios ordenes de magnitud superiores a las otras componentes en  $x$ .

Ahora cabe mencionar, el hecho de que la fuerza en el eje  $y$  es cero, o prácticamente cero tal y como se verá ahora cuando volvamos a mostrar las curvas obtenidas con seaFEM, ya que la ola se propaga unidireccionalmente en el eje  $x$ , y al ser simétrica la Spar y sobre todo el Hard tank (que es un cilindro) (que es donde se obtiene los mayores componentes de fuerzas), la posibles componentes de fuerzas en  $y$ , se anulan en la realidad una a la otra en el hard tank, al ser iguales pero de sentido contrario, por lo que no se representarán las fuerzas en  $y$ .

Ahora bien la representación conjunta de las fuerzas  $x$  e  $z$  se muestran a continuación, en donde se ve claramente la diferencia de órdenes de magnitud.

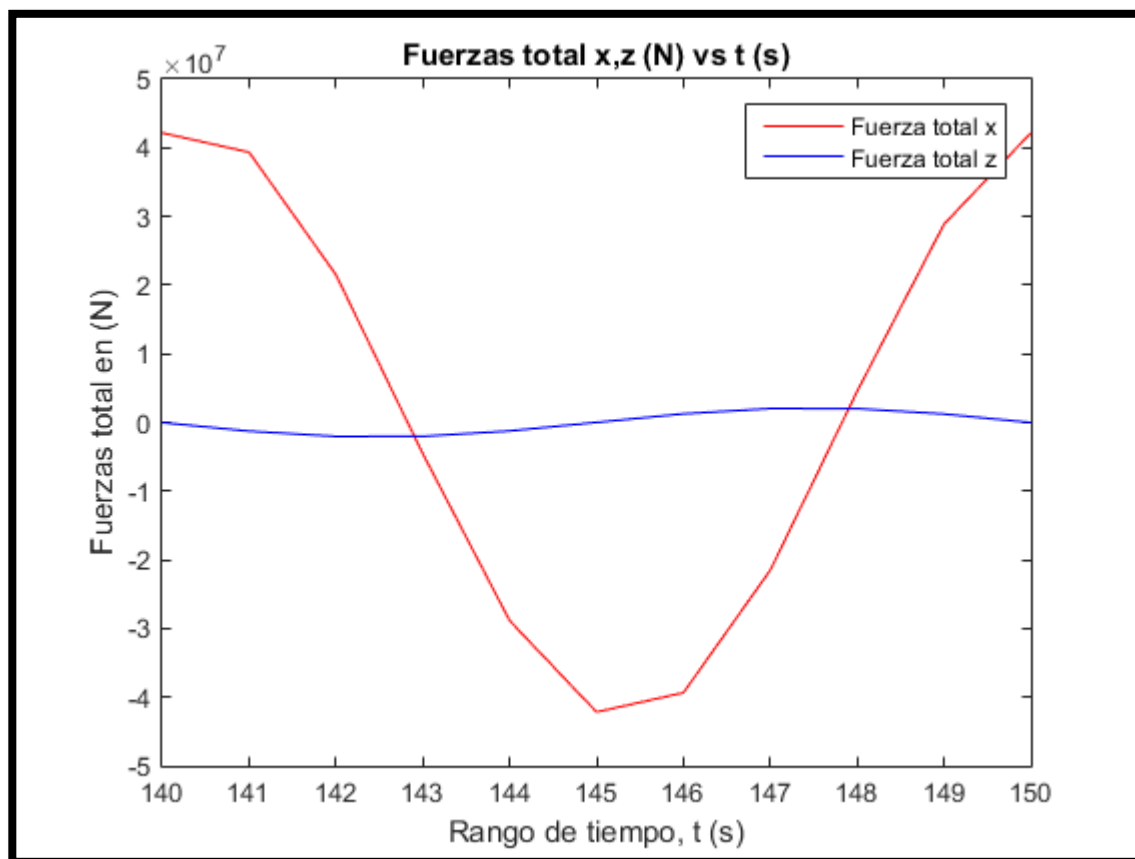


Figura 4.9 Fuerzas  $x$ ,  $z$ , totales, las fuerzas en  $y$  son cero

Ahora se hará una comparación del resultado obtenido analíticamente, con los resultados obtenidos con el programa seaFEM y mostrados en la Figura 1.18, y que se mostrarán de nuevo ahora por comodidad.

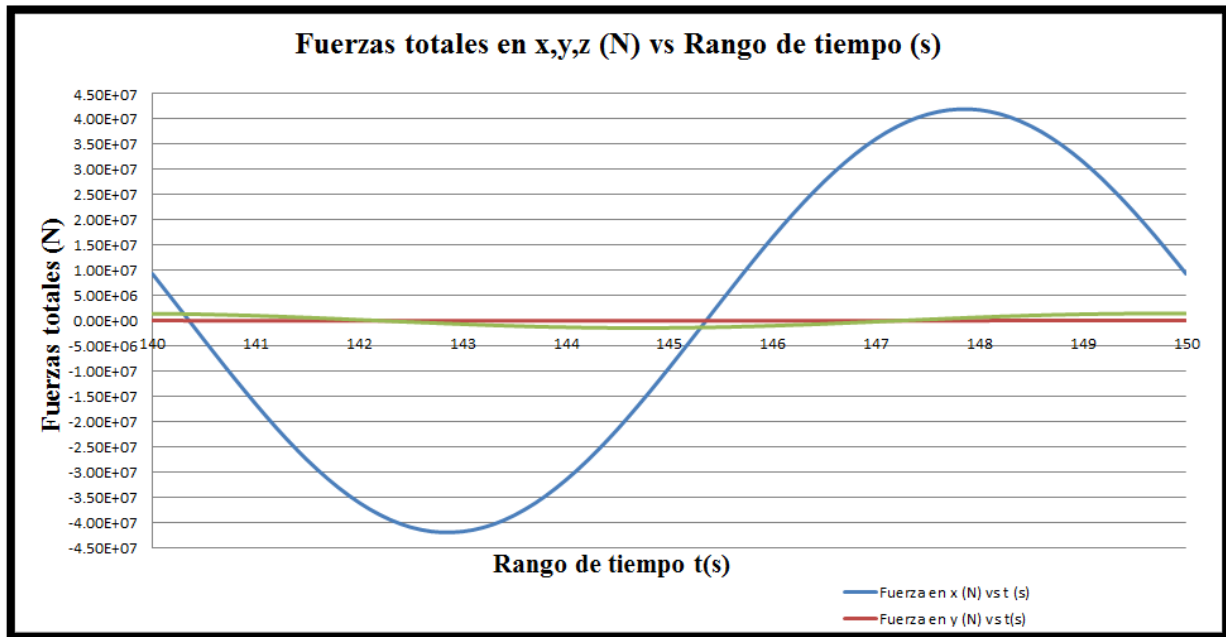


Figura 4.10 Fuerzas x,y,z para una ola monocromática de  $H=8$  m,  $T=10$ s

Tal y como era de esperar la componente x es la dominante de las otras dos, y en donde la componente y es casi despreciable. Y si comparamos los valores máximos vemos que se obtienen valores muy próximos, o prácticamente iguales a los obtenidos analíticamente, de  $4.2E+07$ . Aunque nuestra ola, presenta un desfase de +3 segundos con respecto a la que se ha obtenido con el programa. Si alteramos los límites de tiempo en el archivo datos.inp a un nuevo rango de 143 segundos a 153 segundos obtenemos:

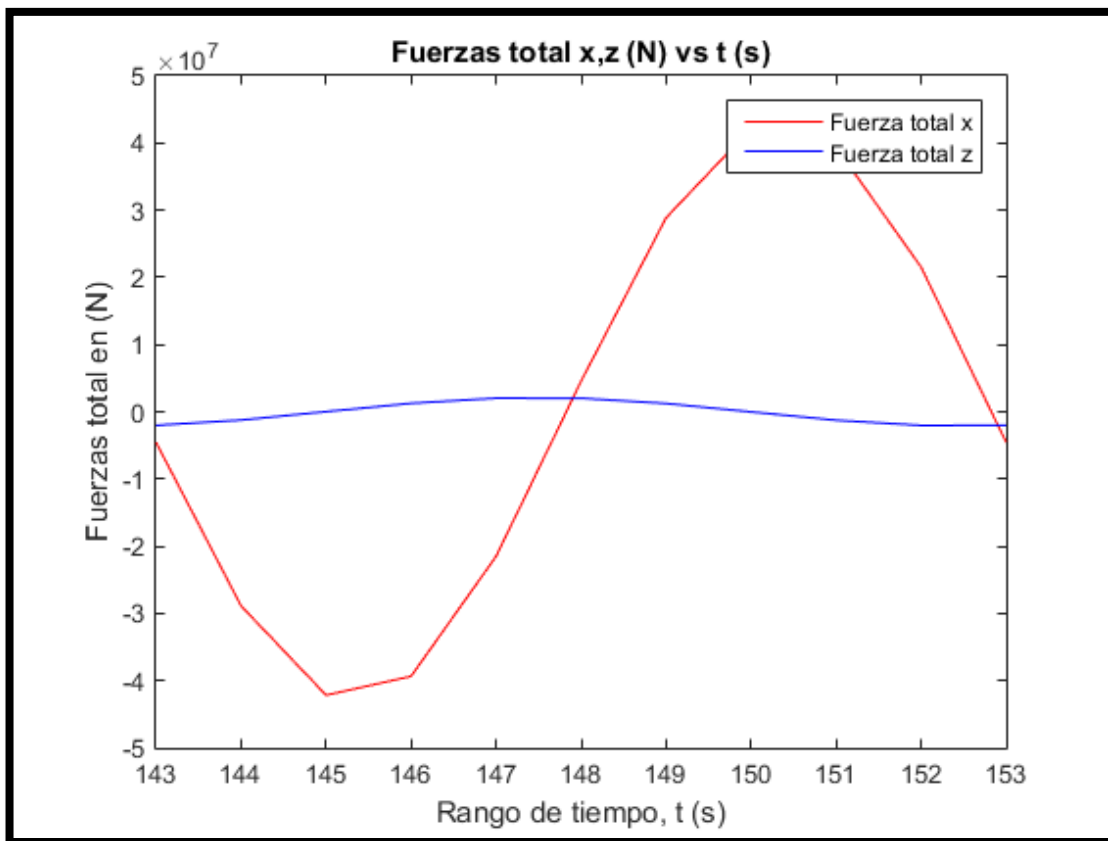


Figura 4.11 Curva de fuerzas totales x,z obtenidas analíticamente para aun rango de tiempo 143-153 s

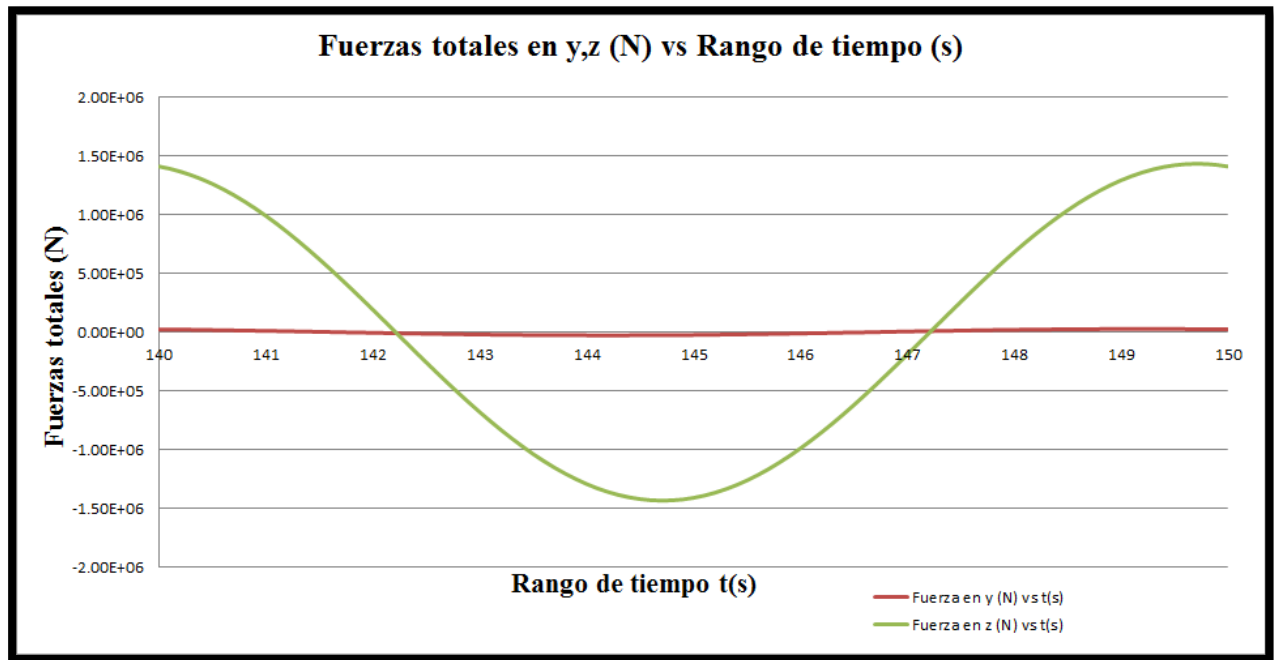


Figura 4.12 Fuerzas x e y obtenidas con el programa seaFEM

Si ahora analizamos la fuerza z obtenida analíticamente y mostrada en la Figura 4.8, con la fuerza z obtenida utilizando el programa vemos que para empezar los resultados están desfasados. Y que los valores máximo obtenidos analíticamente son  $0.5 \cdot 10^6$  mayores que los obtenidos con seaFEM.

Podemos concluir que aunque no se ha obtenido analíticamente curvas con la misma fase que las obtenidas con el programa seaFEM, se han obtenido valores máximos lo suficientemente próximos a los que habría que esperar para las fuerzas x e z. Y en donde la componente y de las fuerzas analíticas se ha considerado como cero directamente, siguiendo el razonamiento mencionado anteriormente. Esta consideración, de despreciar las fuerzas y, queda validada con los resultados obtenidos con el programa seaFEM para dicha componente que se muestra en la Figura 4.12.