TRABAJO FIN DE GRADO

SPLINES INTERPOLANTES EN EL MUNDO NAVAL

Alumno: Octavio García Aparicio

Directores: Juan Carlos Trillo Moya
Juan Ruiz Álvarez

DEPARTAMENTO DE MATEMÁTICA APLICADA Y ESTADÍSTICA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA NAVAL Y OCEÁNICA
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

TRABAJO FIN DE GRADO

SPLINES INTERPOLANTES EN EL MUNDO NAVAL

Alumno: Octavio García Aparicio

Directores: Juan Carlos Trillo Moya
Juan Ruiz Álvarez

DEPARTAMENTO DE MATEMÁTICA APLICADA Y ESTADÍSTICA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA NAVAL Y OCEÁNICA
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

“Gracias a Dios, a su Madre, a mis padres, a mis abuelos, a mis hermanos, a mi novia y a los directores de este trabajo. Porque ellos fueron sustento y motivación para finalizar el grado.”

“Gracias a mis compañeros de promoción por estar siempre allí donde los necesité.”

INDICE

1. INTRODUCCIÓN

¿QUÉ ES UN SPLINE?

El spline es una herramienta matemática que se utiliza para simular diferentes fenómenos físicos o formas reales con la intención de que se aproxime al máximo a la realidad que queremos reflejar.

ESTADO DEL ARTE

En el mundo naval la representación de curvas es parte imprescindible del diseño de un buque. Antiguamente todas las representaciones se hacían sobre el papel y en el caso de querer reflejar esto en un modelo en tres dimensiones solo podíamos hacerlo realizando maquetas a escala del buque o parte de el.

La aparición de los ordenadores y los programas de diseño tridimensional han supuesto una revolución para la representación de curvas a partir de dibujos vectoriales.

Existen multitud de formas de Spline: Splines Interpolantes, splines suavizantes, B-Splines, splines racionales, NURBS (Non Uniform Rational B-Splines), splines en dos dimensiones, etc.

Son tantas las utilidades de los Splines que actualmente se utilizan en infinitud de campos de diseño, no solo en ingeniería y arquitectura naval. Desde la robótica hasta el diseño de videojuegos, desde la industria del automóvil hasta la recreación de imágenes virtuales en dos dimensiones y todo el modelado en tres dimensiones.

Durante los últimos años el estudio de estos ha sido altamente demandado por las diferentes aplicaciones antes mencionadas, y por otras aplicaciones más presentes en nuestra realidad de lo que pensamos. Gran parte del estudio de estos, se lo ha llevado el suavizado de dichas curvas, buscando remplazar la interrelación por el suavizado. Sobre todo, se requiere de suavizado cuando los valores de las ordenadas son experimentales y pueden contener un margen de error notorio. Estos margenes se traducen finalmente en ruido.

2. OBJETIVOS

El objetivo de este TFG consiste en el estudio minucioso de los Splines interpolantes de quinto grado y la posibilidad de implementación de estos en programas de diseño industrial. A su vez se expondrán de forma concisa los diferentes tipos de Spline con el objetivo de poder comparar y valorar las diferentes propiedades de cada uno.

Este proyecto pretende exponer:

- Introducción a la representación de curvas en el mundo naval.
- Desarrollo breve de diferentes tipos splines según estudios y trabajos anteriormente realizados.
- El desarrollo matemático extenso de los splines de quinto grado.
- Posibles ventajas de la aplicación de estos splines frente a los de tercer grado.
- Posibles aplicaciones en el mundo naval y otros sectores de la industria.
- Desarrollo de splines de quinto grado en el programa Matlab. Herramienta muy utilizada en el campo matemático y en el tratamiento de datos.
- Estudio de los resultados obtenidos.
- Desarrollo de una interfaz gráfica intuitiva.
-

3. INTRODUCCIÓN AL DISEÑO NAVAL

Podemos decir que la aplicación de Splines más representativa en el mundo naval, es la representación de las formas del barco . Es por ello que a continuación se va a exponer una introducción al diseño naval. Vamos a exponer desde las zonas de referencia hasta las técnicas más comunes de representación. Muchas de estas formas de representación se remontan a los tiempos anteriores a la llegada de los programas informáticos de diseño.

REFERENCIAS PRINCIPALES DEL BUQUE

A continuación se relatan la nomenclatura más básica de la construcción naval. Estos conceptos aunque triviales son sin embargo pilares básicos necesarios para comprender todo el proceso que requiere el diseño de un buque.

La parte delantera del barco y la parte trasera se llamarán proa y popa respectivamente. Estas referencias son inamovibles y están fijadas según el movimiento normal del barco. Según esta misma referencia quedaría en el lado derecho el costado de estribor, y en el lado izquierdo el costado de babor. La zona del costado más próximo a la proa será la amura, y la zona del costado más próximo a la popa será la aleta.

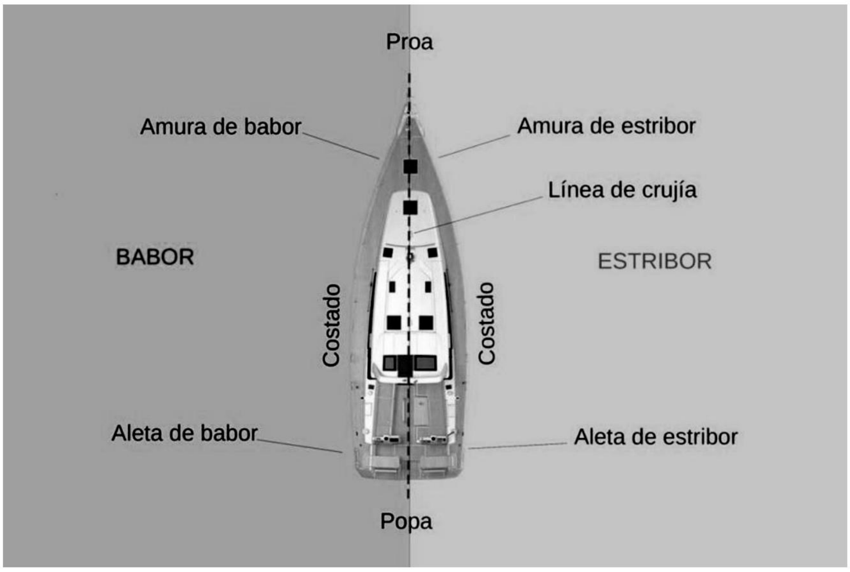


Imagen 3.1: Referencias longitudinales principales del buque.

Transversalmente podemos definir las siguientes referencias: La cubierta superior será la superficie más alta que cierra el casco. El forro es la superficie que forma el cierre exterior del casco. La quilla plana es la zona inferior del casco, la cual se encuentra en el plano de crujía. El fondo será la parte más baja del casco, se encuentra junto a la quilla. La zona curvada que hace de unión entre el fondo y el costado del buque será denominada pantoque. El costado, cómo podemos intuir por su nombre, es la zona lateral del caso del buque.

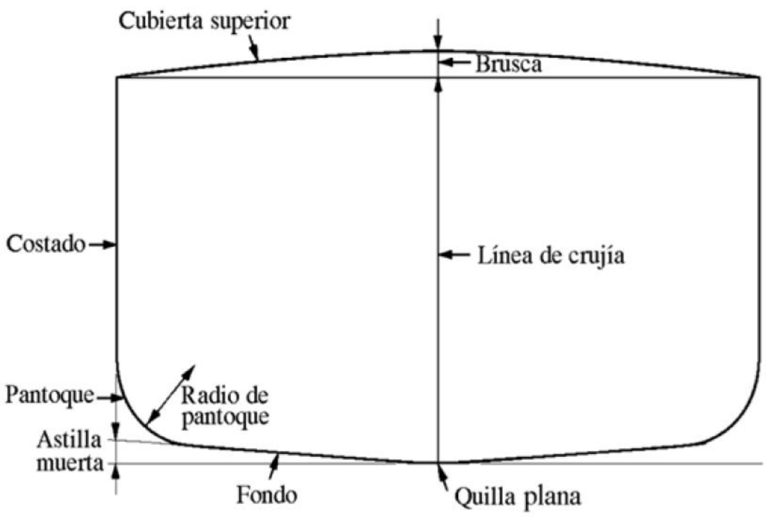


Imagen 3.2: Referencias transversales principales del buque.

Podemos definir el buque como una caja estanca cuya forma dependerá de la función de dicho buque. Esta forma estanca es la que denominamos casco, y sobre este caso se levanta la superestructura. La zona del casco que queda por debajo de la línea flotación se denomina obra viva, el resto se denomina obra muerta.

Podemos definir el buque como una caja estanca cuya forma dependerá de la función a la que se dedique el buque. Esta forma estanca es la que denominamos casco y sobre este caso se levanta la superestructura. La zona del casco que queda por debajo de la línea flotación se denomina obra viva, el resto de se denomina obra muerta.

El plano de simetría longitudinal es lo que denominamos plano de crujía. Este plano es uno de los planos de referencia.

En la siguiente imagen se muestra el nombre que recibe la “altura de una cubierta” y “la curvatura de la cubierta superior en el sentido longitudinal”. Estas se denominan puntal y arrufo respectivamente.

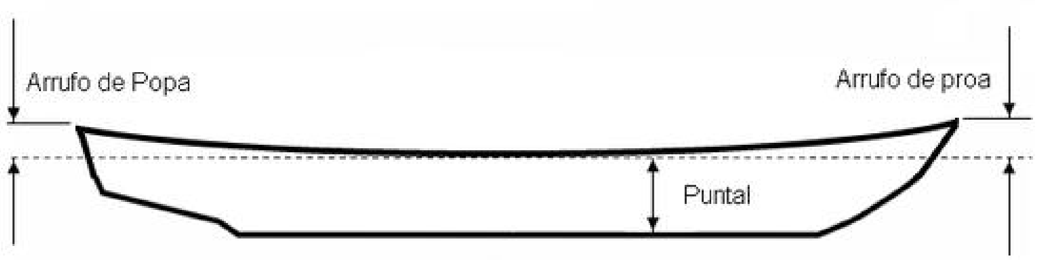


Imagen 3.3: Arrufo y puntal de un buque.

Aunque no se muestran en la imagen, la roda será la zona más a proa del casco, allí es donde se unen los dos costados en proa. Por otro lado está el codaste que es donde se unen los dos costados en popa por debajo de la flotación. Y el arrufo es la curvatura que tiene la cubierta en el sentido longitudinal, esta será mayor en los extremos.

DIMENSIONES DE LOS BUQUES.

La eslora es la dimensión del barco en el sentido longitudinal. Existen diferentes formas de medir la eslora:

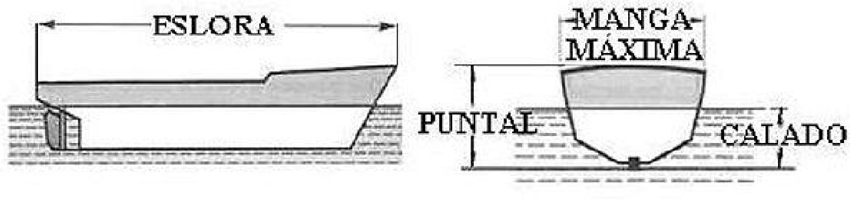


Imagen 3.4: Dimensiones del buque.

- Eslora entre perpendiculares: Es la longitud medida entre las perpendiculares de proa y popa. La perpendicular de proa es una linea perpendicular a la flotación qué pasa por la intersección de la linea de flotación con la roda. Se considera la sección media a la sección transversal que se encuentra en la mitad de la eslora entre perpendiculares.

A continuación se muestran dos ilustraciones que pretenden mostrar la posición de mencionadas perpendiculares.

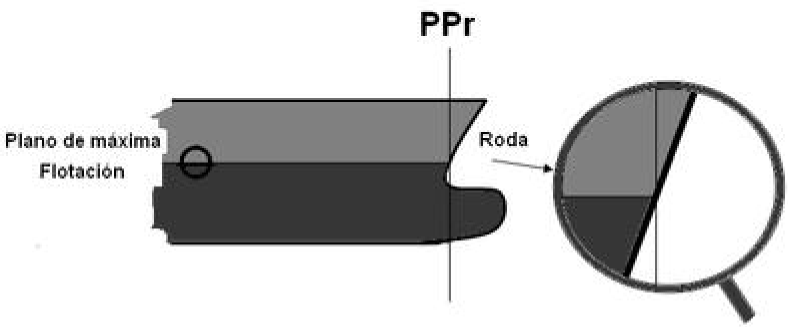


Imagen 3.5: Perpendicular de Proa (Ppr)

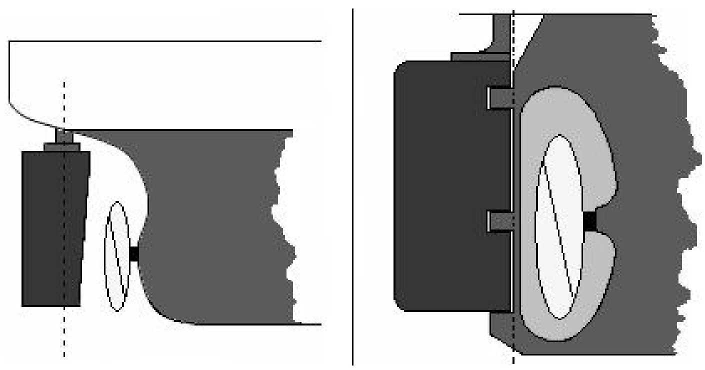


Imagen 3.6: Perpendicular de Popa (Ppp)

- La eslora en la flotación es la longitud total que existe entre la perpendiculares a la linea de flotación. La perpendicular de proa es la que pasa por la intersección de la línea de flotación y la proa, y la perpendicular de popa es la que pasa por la intersección de la línea de flotación y la popa.

- La eslora de francobordo es según el reglamento *"igual al 96% de la eslora de flotación correspondiente al 85% del puntal de trazado, medida en metros desde la perpendicular de la proa, o la eslora comprendida entre la perpendicular de proa y el eje de la mecha de timón, medida en la misma flotación si este fuese mayor."*

La manga es la medida del barco en el sentido transversal, podemos distinguir entre:

- La manga de trazado.. Es la máxima dimensión transversal que existe a lo largo de la eslora del buque. En buques de madera la manga de trazado será fuera de forros, y en buques de metal será fuera de miembros. Por tanto la manga fuera de foros será igual a la manga de trazado más el espesor del casco.

- La manga en una flotación es la manga máxima en una flotación definida.

El puntal es la dimensión en sentido vertical.

- El puntal trazado es la máxima longitud en el sentido vertical del trazado del casco del buque. Es medida en la sección media y en el costado. En el caso de buques con el casco metálico se miden desde el canto alto de la quilla hasta la cara inferior de la cubierta principal. El puntal de trazado se representa con la letra D.

- El puntal de bodega será longitud en el sentido vertical desde la cara alta del fondo, o doble fondo hasta la cara inferior de la cubierta más baja.

- El puntal de entrepuente: Es la longitud vertical entre dos cubiertas seguidas dentro de una bodega. El entre puente es por espacio de la bodega entre dos cubiertas de esa misma bodega.

El calado es la longitud de sentido vertical corresponde la parte sumergida.

- Calado de trazado (T): longitud en sentido vertical de trazado de lo que corresponde la parte sumergida del casco. Queda por debajo de la flotación de trazado y se mide en la sección media del buque.

- Calado en una flotación: Es el calado medido desde la quilla, en su cara interior o exterior, hasta el nivel de la flotación correspondiente.

- Calado a proa (Tpr): Calado real del buque en la perpendicular de proa.

- Calado a popa (Tpp): Calado real del buque en la perpendicular de popa.

- Calado medio: Semisuma de calados de proa y popa.

- Calado en la sección media: el calado en la sección media de nuestro barco

- Asientos o trimado: la resta entre el calado de popa y el calado de proa. Cuando el de popa es mayor que el de proa, el asiento tendrá signo positivo, en estos casos se dice que el buque nuestro tiene asiento positivo. En el caso de que sea al revés, el barco tendrá asiento negativo.

PLANOS Y LINEAS DE REFERENCIA EN EL BUQUE..

Solemos cortar el buque en tres tipos de planos. Los planos más relevantes son: El plano de crujía, el plano de flotación y el plano transversal.

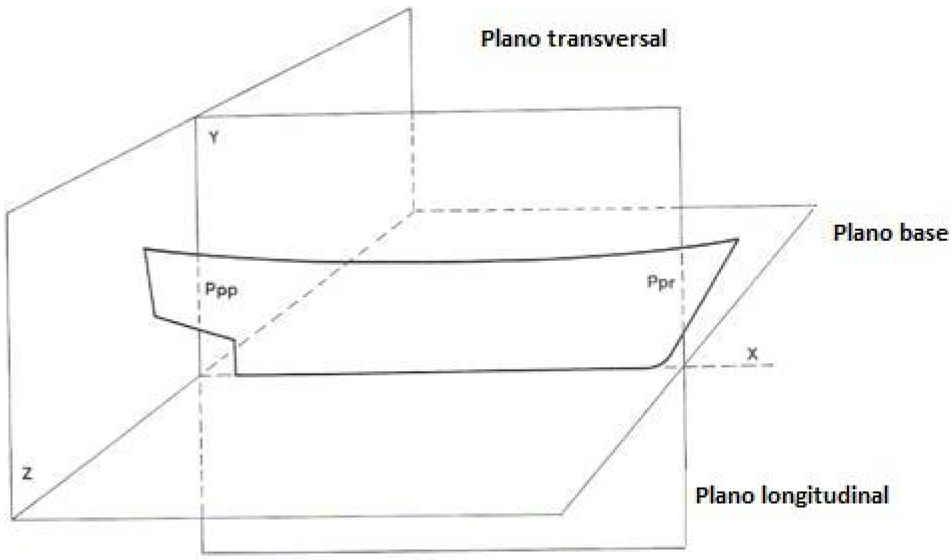


Imagen 3.7: Representación de tres planos perpendiculares en el buque.

- Plano de crujía: Es el plano en el sentido longitudinal que corta en barco en dos mitades, una de estribor y otra a babor. La intersección entre el casco y dicho plano se llama linea de crujía. También existen otros planos paralelos a este que son los planos longitudinales. Las lineas de intersección del casco con estos, son los longitudinales.

- Plano transversal: Es un plano en en el sentido transversal del buque, perpendicular a los planos longitudinales y a los planos de flotación. Las secciones se denominan cuadernas de trazado y se suelen representar en un mismo plano varias secciones del buque, desde proa hasta popa. Debido a su simetría, se representa solo una parte de estas secciones, las secciones que van desde la sección media hasta popa se representan a un lado de crujía y las que van desde la sección media hasta proa a otro lado. Este tipo de planos se denomina caja de cuadernas.

- Plano de flotación: Este plano representaría lo que es la superficie del mar en el caso de que no existiese oleaje alguno. El plano de flotación de trazado es aquel que contiene a la flotación de trazado, esta es la flotación que se supone que tendrá un buque en la realidad, pero no necesariamente corresponde al calado real del buque.

La intersección del plano de flotación con el casco se denomina linea de flotación, si trazamos otros planos paralelos, los cortes con el casco se denominan lineas de agua. El plano paralelo que pase por el canto alto de la quilla en la sección media será el plano base o plano horizontal. La intersección con el plano de crujía se denominará la linea base.

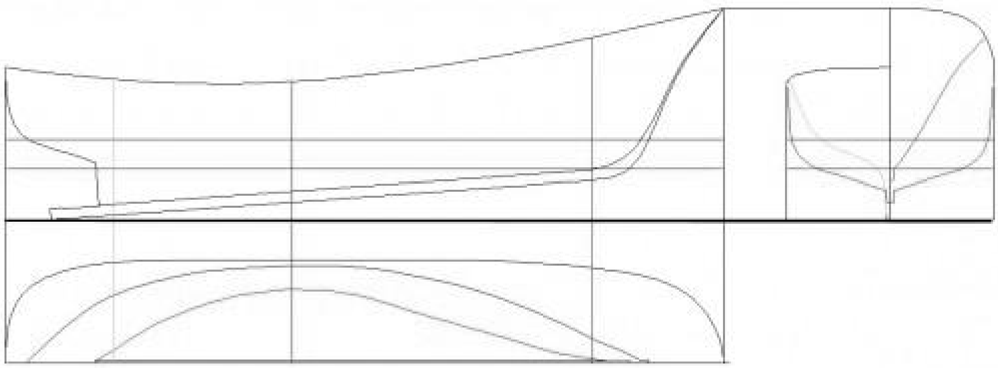


Imagen 3.8. Representación de las intersecciones del casco con los planos.

4. REPRESENTACIÓN DE LAS FORMAS DE UN BUQUE

Para poder representar las características geométricas del casco del buque representamos las intersecciones del casco con diferentes secciones paralelas a los planos que conforman el sistema de referencia que utilizamos. Estas formas representadas a escala en un plano es lo que denominamos plano de formas.

En el plano de formas podemos representar cualquier punto de la superficie del casco. Como los buques son simétricos, se suele representar solo la mitad de babor.

LINEAS DE REFERENCIA.

Se necesita para las representación unos lineas o ejes de referencia que sirvan como eje tridimensional sobre el que construir el plano de formas.

Existen distintas normas respecto a la colocación de ejes, su sentido y dirección, por eso es importante indicar que estándar se está utilizando. Los más conocidos son el americano y el europeo.

Colocaremos el origen de nuestro eje tridimensional en el plano de crujía de forma transversal y de forma vertical en la linea base que se tome. En la dimensión longitudinal puede variar su colocación según su uso.

En el sistema europeo se coloca en la perpendicular de popa, mientras que en el americano se coloca en proa. Para estudios hidrodinámicos se coloca el eje en la sección media, con la intención de facilitar el calculo de la posición del centro de carena respecto a la sección media.

<table>
<tr><th>Criterio</th><th>Eje OX</th><th>Eje OY</th><th>Eje OZ</th></tr>
<tr><td>Europeo</td><td>Positivo en el sentido de popa a proa.</td><td>Positivo Estribor
Negativo Babor</td><td rowspan="2">Sentido positivo de la quilla hacia cubierta</td></tr>
<tr><td>Americano</td><td>Positivo en el sentido de proa a popa</td><td>Positivo Babor
Negativo Estribor</td></tr>
</table>

LONGITUDINALES

Representaremos las llamadas lineas longitudinales en el plano longitudinal. Estas lineas representan los intersecciones del casco con diferentes planos paralelos al eje X e Z, es decir son paralelos al plano de crujía.

El proyectista es el que elije el número de planos, lo normal es utilizar entre 3 y 6 planos equidistantes más la linea de crujía. Los planos se nombran con números romanos.

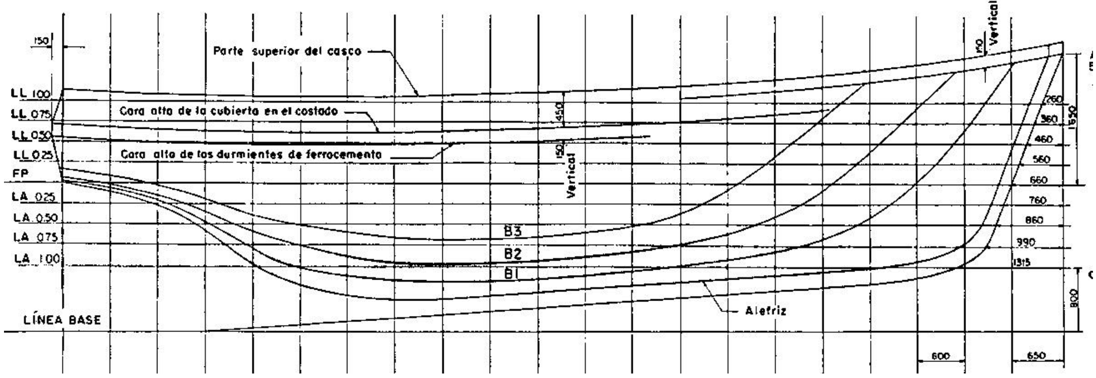


Imagen 4.1. Representación de las lineas longitudinales.

LINEAS DE AGUA

Representaremos lo que llamamos "lineas de agua" en el plano horizontal del nuestro plano de formas. Estas lineas se forman al intersectar el casco con plano paralelos al plano de flotación,es decir, paralelos al eje X e Y del buque.

Se toma como linea 0 la linea del plano base y la linea de flotación se hace coincidir con la 6. Cuando los barcos tienen mucho calado se suele ampliar el número de lineas de flotación, para que el casco quede así mejor definido. También se suele utilizar una linea de agua 1/2 con una distancia de medio intervalo de la linea base.

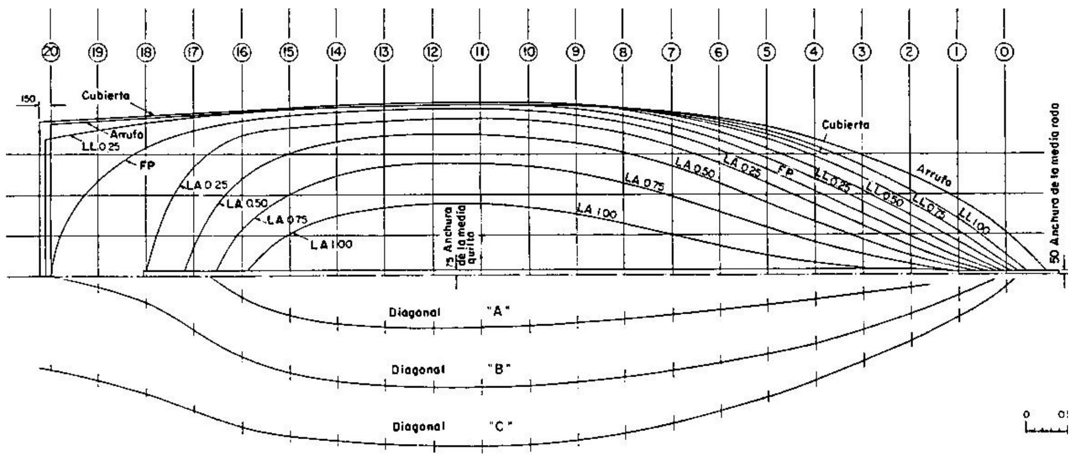


Imagen 4.2. Representación de las lineas de agua.

LINEAS TRANVERSALES

Representaremos las intersecciones de los diferentes planos transversales con el casco del buque en la proyección transversal del plano de formas. Estos planos son paralelos entres si y paralelos al plano contenido en los ejes Z e Y.

Llamamos caja de cuadernas a la representación de estas intersecciones. Se representan las que van desde la sección media hasta proa a la derecha y las que van desde la sección media hasta la popa a la izquierda. Normalmente se divide el buque en unas 10 o 20 secciones según las necesidades de que tenga la representación del buque.

En buques de grandes esloras se usarán las 20 secciones. En los finos de proa y popa se utiliza secciones con una distancia menor a la habitual para definir así mejor las formas, las cuales varían más en estas zonas.

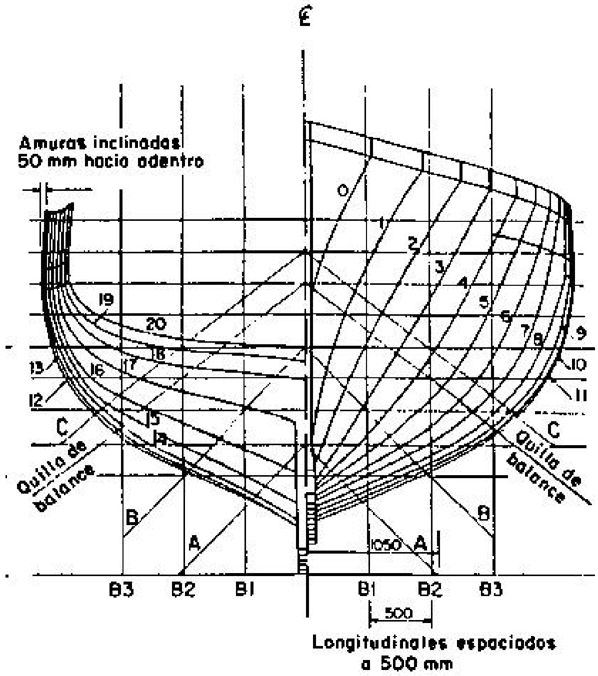


Imagen 4.3. Caja de cuadernas

DIAGONALES

Se trata de secciones "especiales" utilizadas para dar más información sobre las formas del casco. Estos planos son perpendiculares a las secciones transversales y forman un ángulo con el plano de crujía. Su función es dar información adicional en zonas difíciles como podría ser el pantoque, la unión entre el costado y el fondo. Esto evita así la aparición de abolladuras. Se dibujan en la proyección horizontal.

PLANO DE FORMAS.

Tomando los planos anteriormente mencionados y dándoles un orden lógico obtenemos el plano de formas.
La proyección horizontal se coloca debajo de la longitudinal. Se representan en todas las figuras las secciones en todos los sentidos, las secciones transversales se hacen coincidir en los planos longitudinal y horizontal. Las secciones horizontales, llamadas "líneas de agua", se hacen coincidir entre el plano longitudinal y la caja de cuadernas. Además en el plano lo9ngitudinal debemos añadir las intersecciones de las diferentes cubiertas con el plano de crujía.

El criterio europeo nos dice que las secciones se cuentan desde la perpendicular de popa de 0 a 10 o de 0 a 20. El buque se colocará con la proa hacia la derecha.

El criterio americano nos dice que empecemos a contar desde la perpendicular de proa. La proa quedaría mirando hacia la izquierda.

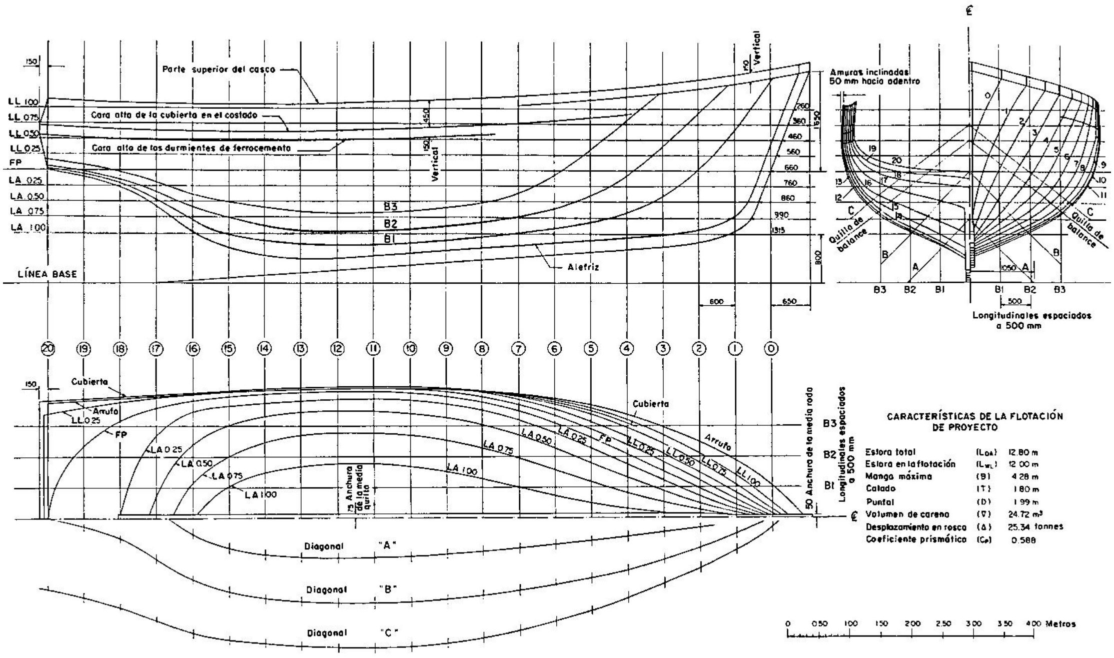


Imagen 4.4. Plano de formas completo. Criterio europeo

TRAZADO

Existen diferentes formas de representar la superficie del casco. Dependiendo del material que esté hecho el buque, tenemos que hacer esta medición de una forma u otra.

Si el casco del buque esta hecho de metal en la medición de trazado no se tendrá en cuenta los espesores de los forros, es decir, el buque tendrá un trazado fuera de miembros. Por el contrario, si el buque fuese de madera o materiales compuestos, donde el espesor es considerable, el buque tendrá un trazado fuera de forro, es decir, el trazado tendrá en cuenta el espesor.

5. CARTILLA DE TRAZADO O TABLA DE GALIBOS.

Parta definir las formas de nuestro buque también disponemos de la cartilla de trazado o tabla de galibos.

STATION Nº	HALF BREADTHS										HEIGHTS ABOVE/BELOW C.W.L.						DIAGONALS			STATION Nº
	W.L. 1	W.L. 2	C.W.L. 3	W.L. 4	W.L. 5	W.L. 6	W.L. 7	W.L. 8	W.L. 9	DECK AT SIDE	PROFILE	BUT. I	BUT. II	BUT. III	DECK AT SIDE	DECK AT ℄	A	B	C	
0	–	–	0	358	511	612	680	727	758	770	0	+ 72	+ 284	–	697	757	662	546	224	0
1	–	235	501	641	738	805	851	881	899	904	– 135	– 82	+ 65	+ 612	700	795	825	709	376	1
2	145	535	707	810	881	931	964	983	991	992	– 210	– 174	– 70	+ 230	706	827	949	820	472	2
3	393	674	812	897	956	996	1021	1034	1039	1040	– 241	– 216	– 137	+ 102	715	854	1018	885	517	3
4	473	715	840	917	971	1006	1029	1040	1044	1046	– 250	– 230	– 159	+ 73	728	875	1037	906	534	4
5	456	685	805	878	928	962	986	1000	1008	1011	– 249	– 229	– 149	+ 137	744	892	1008	893	532	5
6	366	600	714	784	833	870	899	918	930	940	– 241	– 214	– 100	+ 404	762	905	930	848	515	6
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7

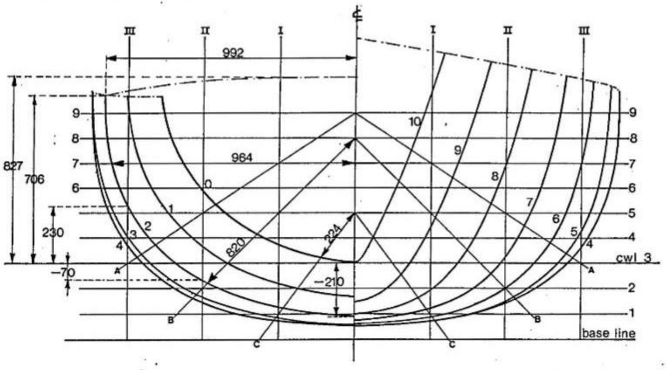


Imagen 5.1. Cartilla de Trazado

La sección uno nos proporciona la distancia que hay entre la intersección de cada línea de agua con las secciones y la línea de crujía, a esto se le denomina semimangas.

La sección dos representa las alturas por encima o por debajo de la línea de flotación central (CWL)

La sección tres de la tabla proporciona las dimensiones para transferir a las diagonales que hayamos tomado.Estas ultimas no se suelen tomar para cascos con pantoque.

6. El ALISADO DE FORMAS Y SUS METODOS.

INTRODUCCIÓN.

En este apartado vamos a hablar del alisado de formas o suavizado al que está sometido el proyecto para obtener unas formas factibles.

Los métodos tradicionales estaban basados en la experiencia del trabajador. La persona que realizaba los planos debía levantar las pesas para relajar el junquillo y colocarlo en una nueva posición según su criterio.

Actualmente se puede hacer a través de los programas un alisado automático, pero debido a que este tipo de alisados pueden cambiar en exceso las formas y apartar estas de su utilidad original, se precisa de un criterio humano para discernir la "intensidad" del suavizado.

A continuación se exponen de forma breve diferentes métodos de alisado tanto históricos como actuales.

ALISADO DE GALIBO

Se trata de un método histórico donde el trazado se reproducía en escala 1:1 en lo que se denominaba la sala de galibos. Los planos de grandes dimensiones se realizan en el piso de la sala, la cual era de grandes dimensiones. Se dibujaba la caja de cuadernas y las mitades proa y popa.

En muchas ocasiones no se disponía del espacio necesario. En estos casos se dibujan solo ciertas secciones del buque o se hacían en escalas menores, 1/2 o 1/4 de la escala longitudinal.

Se preparaba una base firme sobre la que hacer los trazados: el piso debía ser de una madera de gran espesor para poder cepillar el piso cuando se requiriese. Normalmente esta se pintaba de negro para facilitar el marcado. Además la nave debía disponer de una gran iluminación, carecer de columnas que entorpeciesen y estar cerca del taller para poder llevar sin grandes trayectos las plantillas de madera realizadas.

Las líneas rectas se trazaban o con un cordel impregnado en tiza o con un alambre de acero que se colocaba a cierta altura del piso. Se proyectaba este alambre a través de escuadras, niveles o aparatos que emitían un haz de luz. Para las líneas curvas se usan los junquillos, unas listas de madera deformables. Los junquillos debían tener menor espesor en aquellas zonas que se deformaban más.

Imagen 6.1. Sala de galibos del arsenal de Cartagena. En la foto se muestra ya reconvertida en la Base de Submarinos. Año 1920.

Imagen 6.2. Antigua sala de galibos del arsenal de Cartagena. Actualidad.

Se muestran las dos imágenes anteriores como mera curiosidad histórica de la ciudad donde curse la carrera. Cartagena es conocida por su historia íntimamente vinculada con el mar.

ALISADOS A ESCALA 1:10

Se trata de un método manual mucho más sencillo, preciso y rápido que el anterior. Consiste en obtener la cartilla de trazado, a partir de un dibujo a escala 1:10. Es en 1945 cuando en la Europa continental se empiezan a utilizar este método. En los astillero que utilizan este proceso empiezan a notar las ventajas.
Se reproducía en una plantilla transparente e indeformable las curvas que posteriormente son interpretadas por una máquina. La máquina sigue el contorno del dibujo y copia las curvas.

De los datos obtenidos de la cartilla de trazado preliminar obtenida de la sala de proyectos se obtiene el plano de formas tal y como se expuso anteriormente.

Se realizaba el proceso de alisado a través de un junquillo forzándolo para que pase por todos los puntos definidos en la cartilla. En el caso de que sea difícil pasar por estos, se hace de forma promediada. Es importante que cada vez que se rectifica un punto se cambie en todas las vistas. Posteriormente, cuando se armonicen todos los puntos, se harán los cambios convenientes en la cartilla de trazado.

Las zonas más conflictivas son las de los finos de proa y popa, una vez resueltas estas se considerarán resueltas las formas del buque.

INCONVENIENTES

Este tipo de métodos antiguos presentan numerosos inconvenientes, ya que los trazados son muy subjetivos y dependen mucho de la experiencia y criterio del dibujante. Además para realizarlos es necesario una gran superficie. Esto último sumado a lo “aparatoso” que es el proceso, hacen de estos métodos algo totalmente inviable y obsoleto.

ALISADO POR METODOS NUMERICOS.

Es el primer paso que se realiza a día de hoy para el proceso de alisado. En este método se representa a través de una aproximación a la curvatura definida por puntos dados, unidos por integraciones sucesivas o por el método de mínimos cuadrados.

Este método divide el buque en tres partes: El cuerpo cilíndrico, el cuerpo de popa y el cuerpo de proa. Las partes estarán separadas por lo que se conoce como límite del cuerpo cilíndrico y se alisaran normalmente empezando por la popa, siguiendo por la proa y terminando por el cuerpo cilíndrico.

Los datos por los que se obtienen posteriormente las formas de nuestro buque en el ordenador se sacan de:

- A través de una cartilla de trazado, un plano de formas, un modelo o directamente del buque. Medimos directamente y obtenemos los datos necesarios para su simulación.

- Generar las formas a partir de parámetros fundamentales de la carena obteniendo así superficies lisas definidas por estos parámetros y por ecuaciones que nos permiten hallar cualquier punto de estas.

Este procedimiento hace que se pueda obtener datos de forma rápida y precisa. Ademas es aplicable diferentes máquinas de control numérico (CNC).

Para una correcta simulación es indispensable:

- Que la superficie cumpla los criterios de alisado y suavidad.

- Cumplir los requisitos impuestos referentes a la precisión.

- Revisar y corregir los procesos automatizados de suavizado.

MÉTODOS GRÁFICOS ITERATIVOS

Existen dos métodos que no solo es que están en pleno desarrollo sino que además ya son ampliamente utilizadas. Estos métodos son el modelo de alambre y el modelo de superficie.

Modelo de alambre: Consiste en la representación de líneas que estructuran el casco buque. Para este modelo se utilizan diferentes métodos:

- Spline

- La simulación de Junquillo

- Curvas de Beizer

- Curvas Nurbs

- B-Splines

Modelos de superficie: Este modelos pretende ajusta una superficie a unos puntos de control representados en una malla. Los tipos de superficie son:

- Superficies de Conos.

- Superficies de Beizer.

- B-Spline.

- Superficies Nurbs

COMPROBACIÓN DEL ALISADO

Para comprobar el alisado existen diferentes métodos:

- Método Visual, nos permiten observar la curvatura. Existen varios tipos:
    - Superficie iluminada: Proyectar un rayo de luz sobre la superficie.
    - Renderizado “metal bruñido”.
    - Renderizado “ cebra”.
    - Representación en color de la curvatura gaussiana.

- Análisis de la curvatura de las curvas: Se utiliza la segunda derivada de las curvas que nos permite observar si nuestra superficie está alisada. Esto será cuando la curvatura sea uniforme. Para la utilización de dichas curvas se requiere una formación específica sobre el manejo e interpretación de estas.

7. CURVAS SUAVES EN DIBUJO NAVAL

En la actualidad el método gráfico interactivo es el más utilizado debido a que presenta muy buenas ventajas y buenísimos resultados. Estos métodos engloban diferentes tipos de superficies curvas de trazado. Otros nos vamos a centrar en los splines de quinto grado.

Puedan entender bien este tipo de curvas vamos a hacer una pequeña explicación de cómo se representa las diferentes secciones, lineas de agua y longitudinales con un junquillo tradicional.

DIBUJO NAVAL ARTESANO

Para el dibujo de curvas bidimensionales, tradicionalmente se se utiliza un instrumento flexible denominado junquillo, en inglés spline. El junquillo recto se pone encima del dibujo entre dos puntos fijos. Posteriormente se le aplican diferentes cargas, ayudándonos de plomos, a lo largo del junquillo para hacer trazar una curva. La suavidad de la curva dependerá de la flexibilidad del junquillo de la posición de los puntos de control y de la experiencia del dibujante.

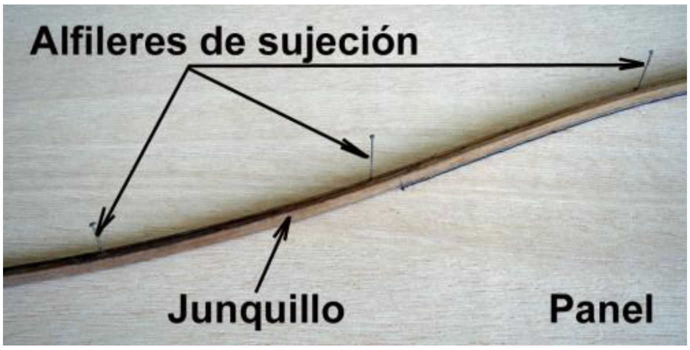


Imagen 7.1. Junquillo puesto sobre el panel.

Podemos definir un Spline, como un junquillo idealmente fino el cual pasa por los puntos de control que serán los plomos.

Para “xi” e “yi” tenemos diferentes valores de "i" que variarán desde el punto cero hasta el punto m.

La ley de Bernoulli-Euler nos dice que la ecuación el junquillo encorvado es la siguiente:

$$EIS''(x) = M(x)$$

Siendo S la flexión m el momento flector, el cual varía linealmente entre apoyos, y por último EI qué es la rigidez. En nuestro caso S será la función de quinto grado, entre punto y punto, y es 4 veces diferenciable.

Los splines interpolantes se hace pasar por los puntos de control a la curva dibujada, por otro lado los splines suavizantes atraen la curva hacia esos puntos de control a través de unos coeficientes denominados pesos. Esto quiere decir que en vez de usar

plomos que se apoyan a lo largo del junquillo, estos usan muelles que atraen la curva Asia y sus puntos de control. Segundo vamos los puntos de control el junquillo y los muelles combinan su flexibilidad para realizar una curva suave. Esto hace que en la curva solo coinciden los extremos en los puntos de control ya que los muelles atraen hacia los puntos de control interiores la curva pero no lo posicionan sobre estos.
Es moviendo los puntos de control como conseguimos el junquillo tomé una forma deseada. Está curva se presenta sin irregularidades debido a qué el junquillo tiene una flexibilidad y los muelles también. Sí aumentamos el número de puntos de control es probable que surjan cambios bruscos de curvatura o puntos dónde se produzcan inflexiones.

Esto podría ser una breve explicación básica de las diferencias que existen entre los splines interpolantes y los splines suavizantes.

SPLINES EN PROGRAMAS TRIDMENSIONALES

En la actualidad existen diferentes programas de diseño qué son capaces de realizar splines tridimensionales. Estos programas también son capaces de generar superficies a partir de esas curvas o puntos, ya que, sí a partir de una serie de puntos en un plano bidimensional, podemos realizar splines bidimensionales, sí disponemos de una nube de puntos tridimensional, podemos crear una superficie tridimensional.

Las mallas que conforman las superficies tridimensionales se forman a partir de filas y columnas de puntos. Estás mayas tienen cuatro laterales y cuatro esquinas.

Dependiendo de la complejidad que se requiera se tomarán más o menos puntos de control, lo que se traduce en más filas y en más columnas.

Tres apuntes característicos de las superficies que recreamos en los programas son:

- Las esquinas coinciden exactamente con el punto de control.
- Los bordes solo son influidos por los puntos de control de los bordes.
- El resto de puntos interiores son influidos por un gran número de puntos de control o incluso todos los puntos de la malla.

8. DEFINICIÓN DE LOS SPLINES

Imaginemos que queremos construir una curva que se asemeje lo máximo posible a un perfil real. Si cogemos muchos puntos de esta curva e intentamos unirlos a través de una interpolación polinómica el resultado no sería ni por el asomo parecido el resultado esperado. Por tanto, para ello, usamos lo que se denominan Splines o dicho de otra manera, funciones polinómicas a trozos.

Disponemos de un intervalo [a,b] que contiene una serie de nodos, siendo el primero el nodo a y el último el nodo b, ambos se denominan nodos fronterizos. El resto de nodos que están entre a y b son denominados nodos interiores.

Partición del segmento [a, b] por distintos nodos.

$$P = \{a = t1 < t2 \cdot \cdot \cdot < tm\text{-}1 < tm = b\}$$

Como ya hemos mencionado anteriormente los nodos fronterizos son t1 y tm. Los nodos interiores son los contenidos entre t2 y tm -1 incluyendo estos dos.

Diremos que S(x) es un spline en el intervalo [a,b], siempre y cuando exista una partición de de ese intervalo en unos nodos, S(x) entre dos nodos contiguos es un polinomio.
La función S(x) que esta definida en un intervalo [a,b], se llamará Spline de orden p+1 y grado p siempre y cuando:

Todos los segmentos [ti, ti+1], teniendo i valores desde 1 hasta m -1, son polinomios de grado menor o igual que p.

$$S(x) = Si(x) = \sum_{K=0}^{p} a_k^{(i)} (x - ti)^k \quad \forall i = 1, ..., m\text{-}1$$

Tenemos m-1 segmentos parciales en todo el segmento [a, b].

En todo [a, b] el Spline será un polinomio de grado p y con p+1 números de coeficientes.

Como consecuencia de esto, para definir un Spline de forma correcta es necesario hallar tantos números como nos da el producto de m-1 y p+1.

El Spline debe ser diferenciable (p - 1) veces y manteniendo la continuidad.

Esto quiere decir que pertenece a la clase C^{p-1} [a,b], esto significa que la función S(x) y sus derivadas S'(x), S''(x), ... ,S^{p-1}(x) son continuas.

Tenemos m-2 nodos internos por tanto para hallar los coeficientes de todos los polinomios se dispone de p(m - 2) condiciones (ecuaciones).

Por tanto para definir el Spline completo necesitamos:

$$(p+1)(m-1) - p(m-2) = (m+p-1)$$

Existen diferentes clases de Splines la mayoría de problemas que nos vamos a encontrar son problemas de Splines Interpolantes y Splines Suavizantes obtenidos a partir de un conjunto de puntos dados.

Mientras que en los Splines interpolantes, la función debe pasar por los puntos dados. Por tanto si tenemos m puntos esto quiere decir que tenemos m condiciones dadas. El resto de p-1 ecuaciones las obtenemos a partir de las derivadas menores del Spline en sus extremos. Esto se denomina condiciones de contorno. Eligiendo las condiciones de contorno podemos hacer gran variedad de Splines con propiedades diferentes.

Por otro lado están los Splines Suavizantes, que se construyen de manera que pasen cerca de los puntos dados pero no necesariamente por estos. Dado que podemos variar la cercanía a estos puntos podemos definir gran variedad de Splines Suavizantes.

En las próximas páginas se explicaran estos tipos de Splines con más profundidad.

9. INTRODUCCIÓN A LOS SPLINES CUBICOS INTERPOLANTES

Llamaremos Spline Interpolante S(x) en un retículo w a aquel que cumpla los siguientes puntos:

Ser una función de tercer grado en cada uno de los intervalos del retículo w.

$$S(x) = S(x) = a_1^{(i)}(x-ti) + a_2^{(i)}(x-ti)^2 + a_3^{(i)}(x-ti)^3$$

Estará definido por cuatro coeficientes, por tanto si, como ya hemos mencionado antes, el retículo contiene (m-1) intervalos, será necesario hallar 4(m-1) número para confeccionar nuestro Spline.

Debe ser dos veces diferenciable con continuidad en S(x), su derivada primera S'(x) y su derivada segunda S''(x) en todos los nodos internos del retículo w. Podemos afirmar entonces que el retículo [a,b] pertenece a la clase $C^2[a,b]$.

Por tanto si el numero de nodos internos es m-2 y las tres funciones deben tener continuidad, esto hace que tengamos 3(m-2) condiciones más.

Si como ya se menciono en el apartado anterior, nuestro Spline cumple que para cada nodo exite un valor, es decir S(ti) = yi, consecuentemente obtendremos m condiciones más.

Por tanto tenemos un total de 3(m-2) + m = 4m - 6 condiciones, pero para definir el Spline en su totalidad nos es necesario un total de 4m - 4 condiciones, por tanto nos faltarían dos. Estas dos que nos faltan serán restricciones sobre los extremos del Spline o sus derivadas.

Las condiciones que podemos utilizar para definir un Spline suelen ser de los cuatro tipos siguientes:

Primer tipo: Valores que debe dar la primera derivada en sus extremos.

$$S'(x) = f'(a) \quad S'(b) = f'(b)$$

Segundo tipo: Valores que debe dar la segunda derivada en sus extremos.

$$S''(x) = f''(a) \quad S''(b) = f''(b)$$

Tercer tipo: Condiciones periódicas, se suelen imponer cuando la función a interpolar es periódica con un periodo de T = b - a

$$S'(a) = S'(b)\ ;\ S''(a) = S''(b)$$

Condiciones de cuarto tipo.

$$S'''(y, t_{2izq.}) = S'''(y, t_{2Drh.})$$

$$S'''(y, t_{m-izq.}) = S'''(y, t_{2Drh.})$$

En la tercera derivada los puntos interiores acostumbran a ser discontinuos, sin embargo aplicando esta última condición podemos reducir el numero de discontinuidades.

10. INTRODUCCION A LOS SPLINES SUAVIZANTES

Suponemos una vez más que nuestro intervalo [a,b] está definido un retículo w.

w: a = t1 < t2 < ... < tm-1 < tm = b

Para cada uno de los nodos tenemos un número.

y1,y2, ... , ym-1, ym.

Para cada punto (xi,yi) puede existir un error en el valor yi. Es decir que para cada uno de los nodos existirá un intervalo.

(C_I, D_I) ó bien (y_{I-d}, y_{I+d})

De forma que cualquier número dentro de este intervalo podría ser un valor de yi. Lo valores de yi podrían ser por ejemplo los resultados de una medición en una función y(x) dados unos valores x. Por tanto para construir dicha función y(x) tomando esos valores experimentales, no es conveniente usar una función interpolante.

En este tipo de casos la mejor opción es usar un spline suavizante, con la intención de reducir la aleatoriedad de los valores medidos. Por lo general en este tipo de problemas nos pedirán una función dados los número correspondientes a cada xi dentro de un intervalo. Ademas nos pedirán que esta función tenga propiedades buenas, como pueden ser: Continuidad en la primera derivada, continuidad en la segunda derivada, un gráfico que no presente curvaturas muy pronunciadas, ausencia de oscilaciones fuertes...)

También podría suceder que nos den unos puntos exactos y nosotros construyamos una función suavizante a partir de estos. Es decir que pase cerca de esos puntos exactos pero sin llegar a interpolar.

Consideramos nuevamente un intervalo [a,b] al que llamaremos retículo w.

w: a = t1 < t2 < ... < tm-1 < tm = b

Para cada uno de los nodos tenemos un número.

y1,y2, ... , ym-1, ym.

Denominamos Spline Suavizante de un retículo w a una función S(x) que cumpla:

En todos sus intervalo [xi, xi+1] sea una función de tercer grado.

$$S(x) = Si(x) = a_0^i + a_1^i(x - xi)^2 + a_2^i(x - xi) + a_3^i(x - xi)$$

11. SPLINES DE QUINTO GRADO

PLANTEAMIENTO DEL PROBLEMA

Para definir un Spline interpolante en un intervalo [a,b] dado que coincida con los nodos.

Suponemos que nuestro intervalo [a,b] está definido un retículo w.

w: a = x1 < x2 < ... < xm-1 < xm = b

Para cada uno de los nodos sacamos un número.

y1,y2, ... , ym-1, ym.

Definimos una función suave S(x) en nuestro intervalo. Dicha función devolverá los valores dados para cada nodo.

S(xi) = yi ∀ i=1,...,m-1,m

Nuestra función tendrá un polinomio por cada intervalo y esos polinomios de quinto grado tendrán una forma que describiremos a continuación:

DEFINICIÓN

Vamos a construir un spline de grado p = 5 por tanto se llamará de orden 6 = (p+1).

En todos sus segmentos se cumple:

$$\Delta_i = [x_i, x_{i+1}] \quad \forall i = 1,...,m-1$$

Es un polinomio de grado ≤ 5

$$S(x) = S_i(x) = \sum_{k=0}^{p} a_k^{(i)} (x - t_i)^k \quad \forall i = 1,...,m-1$$

$$S(x) = S_i(x) = a_0^{(i)} + a_1^{(i)}(x - t_i) + a_2^{(i)}(x - t_i)^2 + a_3^{(i)}(x - t_i)^3 + a_4^{(i)}(x - t_i)^4 + a_5^{(i)}(x - t_i)^5$$

En todo el segmento nuestro Spline será un polinomio de grado 5 con 6 coeficientes. Tenemos en total (m-1) segmentos parciales, es por ello que para definir nuestro Spline de grado 5 necesitaremos (p+1)(m-1) = 6(m-1) números. Es decir, 6 variables por polinomio por el número de polinomios que disponemos.

Variables:

a0(i),a1(i),a2(i),a3(i)a4(i),a5(i) ∀ i = 1 , ... ,m - 1

$$6m-6$$

El número de incógnitas será igual a la suma de los número “zi” de cada ecuación (m) más las cuatro variables por nodo (4*m), pero como el último nodo no tiene polinomio a su derecha quedaría 4*(m-1).

m + 4*(m-1) = m*(4+1)-4 = 5 m - 4

En los puntos interiores debemos pedir que exista continuidad entre polinomios. Por tanto:

$$S_i(x_i) = S_{i-1}(x_i)$$

Nuestro Spline debe ser (p-1) = 4 Veces diferenciable en el segmento [a,b].

Deberá ser por tanto continua no solo en la función S(x) y sus derivadas S’(x) S´’(x) S’’’(x) y S’’’’(x) en todos los nodos interiores (m-2).

$$S'_i(x_i) = S'_{i-1}(x_i)$$
$$S''_i(x_i) = S''_{i-1}(x_i)$$
$$S'''_i(x_i) = S'''_{i-1}(x_i)$$
$$S''''_i(x_i) = S''''_{i-1}(x_i)$$

Por tanto como en cada uno de los nodos interiores de cada derivada dispondremos pues de 4(m-2) condiciones, es decir, 4 derivadas por cada una de las condiciones que se da en los nodos interiores (m-2).

Ademas de las condiciones de continuidad, debemos conocer los valores que toma cada nodo en la dimensión vertical en el Spline. Por tanto para empezar tendremos m condiciones más, una por cada nodo. Es decir, imponer un valor en cada nodo en el Spline. $S_i(x_i) = f_i$. Estas también serán condiciones que deben cumplir nuestro Spline.

Número de condiciones que tengo:

3*(m-2) ----- Condiciones de continuidad en las tres derivadas.

2*(m-1)----- Por cada polinomio dos valores Yi

Por tanto tenemos 5 m - 8 condiciones, y como anteriormente hemos visto que tenemos 5 m - 4 incógnitas, por tanto nos faltan 4 condiciones más para resolverlo.

Como condiciones extra hemos elegido los valores de los extremos de la segunda y los extremos de la cuarta derivada.

3*(m-2)+ 2*(m-1) + 4 = 3m - 6 + 2m - 2 + 4 = 5m - 4

CONSTRUCCION DE SPLINE DE QUINTO GRADO

Vamos a determinar la forma de Si(x) en [xi , xi+1]

Llamaremos Zi a los valores verticales asociados a cada nodo en la cuarta derivada de nuestro Spline. Es decir Zi = S’’’’(xi).

Como S(x) es un polinomio de grado 5, S’(x) lo será de grado 4, S’’(x) de grado 3, S’’’(x) de grado 2 y S’’’’(x) lo será de grado 1.

Por tanto S’’’’(x) serán un conjunto de rectas que cumplirán que:

$$S_i''''(x_i) = Z_i \quad \text{y} \quad S_i''''(x_{i+1}) = Z_{i+1}$$

Como sabemos que S’’’’(x) tendrá entonces una forma usando entonces la interpolación de Lagrage, podemos obtener:

$$S_i''''(x) = z_i \frac{(x_{i+1} - x)}{(x_{i+1} - x_i)} + z_{i+1} \frac{(x - x_i)}{(x_{i+1} - x_i)}$$

Llamamos: $hi = x_{i+1} - x_i$ quedando.

$$S_i''''(x) = z_i \frac{(x_{i+1} - x)}{hi} + z_{i+1} \frac{(x - x_i)}{hi}$$

Integramos sucesivamente 4 veces para obtener S´´´(x), S´´(x), S’(x) y S(x). Una vez obtenidas, aplicamos en ellas las condiciones. A continuación procedemos a explicarlo de forma más detallada.

Integrando una vez obtenemos:

$$\int S_i''''(x)dx = S_i'''(x)$$

$$S_i'''(x) = \frac{-z_i}{2h_i}(x_{i+1} - x)^2 + \frac{z_{i+1}}{2h_i}(x - x_i)^2 + D_i^{(1)}$$

Integrando nuevamente obtendremos:

$$\int S_i'''(x)dx = S_i''(x)$$

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - x_i)^3 + D_i^{(1)}x + D_i^{(2)}$$

Integrando una tercera vez llegamos a:

$$\int S_i''(x)dx = S_i'(x)$$

$$S_i'(x) = \frac{-z_i}{24h_i}(x_{i+1}-x)^4 + \frac{z_{i+1}}{24h_i}(x-x_i)^4 + D_i^{(1)}\frac{x^2}{2} + D_i^{(2)}x + D_i^{(3)}$$

Y para finalizar integramos una cuarta vez, quedando:

$$\int S_i'(x)dx = S_i\ (x)$$

$$S_i\ (x) = \frac{z_i}{120h_i}(x_{i+1}-x)^5 + \frac{z_{i+1}}{120h_i}(x-x_i)^5 + D_i^{(1)}\frac{x^3}{6} + D_i^{(2)}\frac{x^2}{2} + D_i^{(3)}x + D_i^{(4)}$$

NOTA:

Para resolver la integral hemos utilizado:

$$\int (x_{i+1}-x)^2\,dx = \int -t^2 dt$$

t = xi+1-x   dt = -dx

$$\int (x_{i+1}-x)^2\,dx = \int -t^2 dt = \frac{-t^3}{3} + C = -\frac{(x_{i+1}-x)^3}{3} + C$$

PROPOSICIÓN:

El conjunto:

$$B = \left\{(x-x_i),(x_{i+1}-x),\frac{(x-x_i)^3}{3},\frac{(x_{i+1}-x)^3}{3}\right\}$$

Es una base del espacio vectorial de los polinómios con coeficientes reales de grado menor o igual a 3.

PRUEBA:
Dado que Π3 = {polinomios de grado ≤ 3} tiene dimensión 4; es suficiente probar que B es un sistema libre. Inmediatamente por ser dim(Π3)= 4 se seguirá que también es generador y por tanto base.

Veamos que B es libre:

Supongamos que ∃ λ1 , λ2 , λ3, λ4 Є IR tales que:

$$\lambda_1(x-x_i)+\lambda_2(x_{i+1}-x)+\lambda_3\frac{(x-x_i)^3}{3}+\lambda_4\frac{(x_{i+1}-x)^3}{3}=0$$

Igualando según las potencias nos queda:

En x3:

$$\frac{\lambda_3}{3} - \frac{\lambda_4}{4} = 0 \qquad \lambda_3 = \lambda_4$$

En x2:

$$-x_i\lambda_3 + x_{i+1}\lambda_4 = 0 \qquad \lambda_3(x_{i+1} - x_i) = 0$$

En x:

$$\lambda_1 - \lambda_2 = 0 \qquad \lambda_1 = \lambda_2$$

En 1:

$$-\lambda_1 x_i + \lambda_2 x_{i+1} = 0 \qquad \lambda_1(x_{i+1} - x_i) = 0$$

Y como xi ≠ xi+1 serán λ1 = 0 y λ2 = 0

Como hemos comprobado, B es libre y por tanto como tiene 4 elementos, es base de Π3 .

La base B es un sistema generador de Π3 pudiendo entonces expresar cualquier polinomio de Π3.

El polinomio:

$$P(x) = D_1\frac{x^3}{6} + D_2\frac{x^2}{6} + D_3 x + D_4$$

Ǝ C1, C2, C3, C1 Є IR tales que P(x) lo puedo expresar en la base B.

$$P(x) = C_1(x - x_i)^3 + C_2(x_{i+1} - x)^3 + C_3(x - x_i) + C_4(x_{i+1} - x)$$

Por tanto hacemos el cambio de base a la ecuación obtenida para nuestro Spline.

$$S_i\ (x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + D_i^{(1)}\frac{x^3}{6} + D_i^{(2)}\frac{x^2}{2} + D_i^{(3)}x + D_i^{(4)}$$

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + C_i^{(1)}(x - x_i)^3 + C_i^{(2)}(x_{i+1} - x)^3 + C_i^{(3)}(x - x_i) + C_i^{(4)}(x_{i+1} - x)$$

Una vez cambiada la base en nuestro Spline, hacemos el proceso inverso, esta vez derivando, para obtener las derivadas de nuestro Spline.

Derivando una vez nuestro Spline tenemos la primera derivada.

$$S_i'(x) = -\frac{5z_i}{120h_i}(x_{i+1}-x)^4 + \frac{5z_{i+1}}{120h_i}(x-x_i)^4 + 3C_i^{(1)}(x-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x)^2 + C_i^{(3)} - C_i^{(4)}$$

Simplificamos.

$$S_i'(x) = -\frac{z_i}{24h_i}(x_{i+1}-x)^4 + \frac{z_{i+1}}{24h_i}(x-x_i)^4 + 3C_i^{(1)}(x-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x)^2 + C_i^{(3)} - C_i^{(4)}$$

Derivado esta ecuación obtenemos la segunda derivada de nuestro Spline.

$$S_i''(x) = \frac{4z_i}{24h_i}(x_{i+1}-x)^3 + \frac{4z_{i+1}}{24h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

Simplificamos.

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

Derivamos una vez más para obtener la tercera derivada.

$$S_i'''(x) = -\frac{3z_i}{6h_i}(x_{i+1}-x)^2 + \frac{3z_{i+1}}{6h_i}(x-x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

Simplificamos.

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1}-x)^2 + \frac{z_{i+1}}{2h_i}(x-x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

Y por última vez para llegar a la cuarta derivada.

$$S_i''''(x) = \frac{z_i}{h_i}(x_{i+1}-x) + \frac{z_{i+1}}{h_i}(x-x_i)$$

A continuación, una vez sacadas todas las formas generales que tienen las distintas derivadas de nuestro spline y la ecuación de nuestro spline, utilizamos las distintas condiciones para hallar el valor de las incógnitas que se nos han presentado en nuestro Spline de 5º Grado.

Sabemos que tenemos un valor “y” para cada uno de los nodos, por tanto para cada xi tenemos un valor yi, para xi+1 tendremos un valor yi+1.

Por tanto tenemos:

$$S_i(x_i) = yi$$

Y sustituyendo obtenemos la primera condición:

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1}-x)^5 + \frac{z_{i+1}}{120h_i}(x-x_i)^5 + C_i^{(1)}(x-x_i)^3 + C_i^{(2)}(x_{i+1}-x)^3 + C_i^{(3)}(x-x_i) + C_i^{(4)}(x_{i+1}-x)$$

$$S_i(x_i) = \frac{z_i}{120h_i}(x_{i+1}-x_i)^5 + \frac{z_{i+1}}{120h_i}(x_i-x_i)^5 + C_i^{(1)}(x_i-x_i)^3 + C_i^{(2)}(x_{i+1}-x_i)^3 + C_i^{(3)}(x_i-x_i) +$$

$$+ C_i^{(4)}(x_{i+1}-x_i) = y_i$$

Resolviendo la ecuación nos queda.

$$\boxed{\frac{z_i h_i^4}{120} + C_i^2 h_i^3 + C_i^4 h_i = yi}$$

Y por otro lado también tenemos que:

$$S_i(x_{i+1}) = y_{i+1}$$

$$S_i(x_{i+1}) = \frac{z_i}{120h_i}(x_{i+1}-x_{i+1})^5 + \frac{z_{i+1}}{120h_i}(x_{i+1}-x_i)^5 + C_i^{(1)}(x_{i+1}-x_i)^3 + C_i^{(2)}(x_{i+1}-x_{i+1})^3 + C_i^{(3)}(x_{i+1}-x_i) +$$

$$+ C_i^{(4)}(x_{i+1}-x_{i+1}) = y_{i+1}$$

$$\boxed{\frac{z_i h_i^4}{120} + C_i^1 h_i^3 + C_i^3 h_i = y_{i+1}}$$

Continuamos desarrollando todas las condiciones que tenemos para poder hallar así todas las incógnitas que se nos presentan y poder definir nuestro spline.

En la ecuación de la primera derivada del spline, S'(x), podemos imponer la condición de continuidad. Esto quiere decir que en cada uno de los nodos interiores deben coincidir los polinomios contiguos a ese nodo.

En términos matemáticos esto quedaría:

S'i(xi) = S'i-1(xi)

$$S_i'(x) = -\frac{z_i}{24h_i}(x_{i+1}-x)^4 + \frac{z_{i+1}}{24h_i}(x-x_i)^4 + 3C_i^{(1)}(x-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x)^2 + C_i^{(3)} - C_i^{(4)}$$

$$S_i'(x_i) = -\frac{z_i}{24h_i}(x_{i+1}-x_i)^4 + \frac{z_{i+1}}{24h_i}(x_i-x_i)^4 + 3C_i^{(1)}(x_i-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x_i)^2 + C_i^{(3)} - C_i^{(4)}$$

Igualamos el termino (xi+1 - xi) a hi quedando:

$$S_i'(x_i) = -\frac{z_i}{24h_i} h_i^{\,4} - 3C_i^{(2)} h_i^{\,2} + C_i^{(3)} - C_i^{(4)}$$

Después hemos simplificado eliminando los factores que se anulaban entre si.

$$S_i'(x_i) = -\frac{z_i}{24} h_i^{\,3} - 3C_i^{(2)} h_i^{\,2} + C_i^{(3)} - C_i^{(4)}$$

En la siguiente linea se muestran el polinomio contiguo al extremo izquierdo de Si. Volvemos a sustituir xi en los lugares que corresponde.

$$S_{i-1}'(x) = -\frac{z_{i-1}}{24h_{i-1}}(x_i - x)^4 + \frac{z_i}{24h_{i-1}}(x - x_{i-1})^4 + 3C_{i-1}^{(1)}(x - x_{i-1})^2 - 3C_{i-1}^{(2)}(x_i - x)^2 + C_{i-1}^{(3)} - C_{i-1}^{(4)}$$

$$S_{i-1}'(x_i) = -\frac{z_{i-1}}{24h_{i-1}}(x_i - x_i)^4 + \frac{z_i}{24h_{i-1}}(x_i - x_{i-1})^4 + 3C_{i-1}^{(1)}(x_i - x_{i-1})^2 - 3C_{i-1}^{(2)}(x_i - x_i)^2 + C_{i-1}^{(3)} - C_{i-1}^{(4)}$$

$$S_{i-1}'(x_i) = \frac{z_i}{24h_{i-1}} h_{i-1}^{\,4} + 3C_{i-1}^{(1)} h_{i-1}^{\,2} + C_{i-1}^{(3)} - C_{i-1}^{(4)}$$

$$S_{i-1}'(x_i) = \frac{z_i}{24} h_{i-1}^{\,3} + 3C_{i-1}^{(1)} h_{i-1}^{\,2} + C_{i-1}^{(3)} - C_{i-1}^{(4)}$$

Seguimos utilizando las derivadas y les aplicamos las condiciones de continuidad, continuamos con la segunda derivada Sii.

S’’i(xi) = S’’i-1(xi)

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - x_i)^3 + 6C_i^{(1)}(x - x_i) + 6C_i^{(2)}(x_{i+1} - x)$$

$$S_i''(x_i) = \frac{z_i}{6h_i}(x_{i+1} - x_i)^3 + \frac{z_{i+1}}{6h_i}(x_i - x_i)^3 + 6C_i^{(1)}(x_i - x_i) + 6C_i^{(2)}(x_{i+1} - x_i)$$

$$S_i''(x_i) = \frac{z_i}{6h_i} h_i^{\,3} + 6C_i^{(2)} h_i$$

$$S_i''(x_i) = \frac{z_i}{6} h_i^{\,2} + 6C_i^{(2)} h_i$$

$$S_{i-1}''(x) = \frac{z_{i-1}}{6h_{i-1}}(x_i - x)^3 + \frac{z_i}{6h_{i-1}}(x - x_{i-1})^3 + 6C_{i-1}^{(1)}(x - x_{i-1}) + 6C_{i-1}^{(2)}(x_i - x)$$

$$S_{i-1}''(x_i) = \frac{z_{i-1}}{6h_{i-1}}(x_i - x_i)^3 + \frac{z_i}{6h_{i-1}}(x_i - x_{i-1})^3 + 6C_{i-1}^{(1)}(x_i - x_{i-1}) + 6C_{i-1}^{(2)}(x_i - x_i)$$

$$S_{i-1}''(x_i) = \frac{z_i}{6h_{i-1}} h_{i-1}{}^3 + 6C_{i-1}^{(1)} h_{i-1}$$

$$S_{i-1}''(x_i) = \frac{z_i}{6} h_{i-1}{}^2 + 6C_{i-1}^{(1)} h_{i-1}$$

Seguimos con la tercera derivada Siii.

S’’’i(xi) = S’’’i-1(xi)

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1}-x)^2 + \frac{z_{i+1}}{2h_i}(x-x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_i'''(x_i) = -\frac{z_i}{2h_i}(x_{i+1}-x_i)^2 + \frac{z_{i+1}}{2h_i}(x_i-x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_i'''(x_i) = -\frac{z_i}{2h_i} h_i{}^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_{i-1}'''(x) = -\frac{z_{i-1}}{2h_{i-1}}(x_i-x)^2 + \frac{z_i}{2h_{i-1}}(x-x_{i-1})^2 + 6C_{i-1}^{(1)} - 6C_{i-1}^{(2)}$$

$$S_{i-1}'''(x_i) = -\frac{z_{i-1}}{2h_{i-1}}(x_i-x_i)^2 + \frac{z_i}{2h_{i-1}}(x_i-x_{i-1})^2 + 6C_{i-1}^{(1)} - 6C_{i-1}^{(2)}$$

$$S_{i-1}'''(x_i) = \frac{z_i}{2h_{i-1}} h_{i-1}{}^2 + 6C_{i-1}^{(1)} - 6C_{i-1}^{(2)}$$

$$S_{i-1}'''(x_i) = \frac{z_i}{2} h_{i-1} + 6C_{i-1}^{(1)} - 6C_{i-1}^{(2)}$$

Aplicamos la condición de continuidad para la cuarta derivada Siiv :

$$S_i''''(x) = \frac{z_i}{h_i}(x_{i+1}-x) + \frac{z_{i+1}}{h_i}(x-x_i)$$

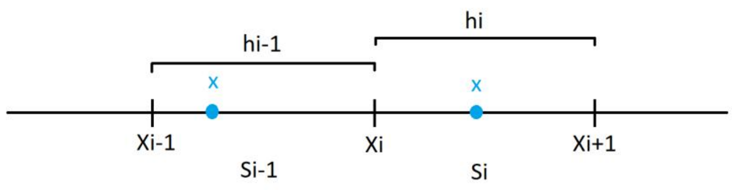
$$S_i''''(x_i) = z_i$$

$$S_{i-1}''''(x) = z_{i-1}\frac{(x_i-x)}{h_{i-1}} + z_i\frac{(x-x_{i-1})}{h_{i-1}}$$

$$S_{i-1}''''(x_i) = z_i$$

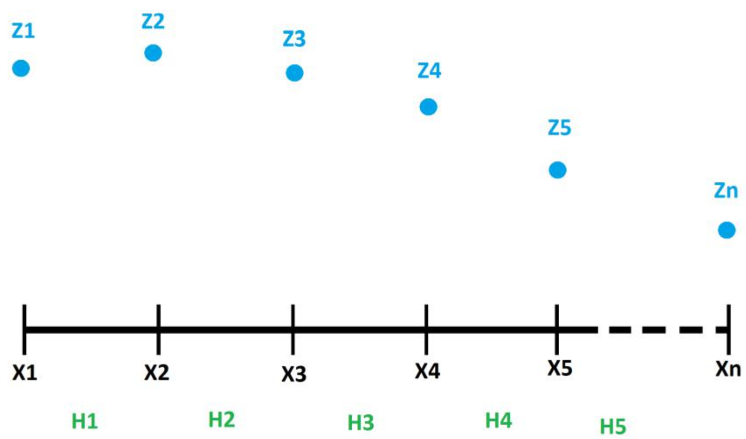
$$z_i = z_i$$

Para entender bien cada una de las variables, se adjunta a continuación un esquema:



CONDICIONES DE CONTORNO PARA VARIOS NODOS:

Antes de exponer detalladamente las condiciones de continuidad para cuatro nodos, se expone un esquema sencillo que pretende ser una ayuda gráfica para identificar cada variable.



Los valores de Zi representan el valor en el eje y para cada nodo en la cuarta derivada.

Para cada polinomio en los nodos existe un valor fijado, estas son las condiciones para los nodos interiores del Spline que debemos sacar.

$$S1(x1) = y1$$
$$S1(x2) = y2$$
$$S2(x2) = y2$$
$$S2(x3) = y3$$
$$S3(x3) = y3$$
$$S3(x4) = y4$$
$$S4(x4) = y4$$
$$S4(x5) = y4$$
$$S5(x5) = y5$$

S1(x1) = y1

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + C_i^{(1)}(x - x_i)^3 + C_i^{(2)}(x_{i+1} - x)^3 + C_i^{(3)}(x - x_i) + C_i^{(4)}(x_{i+1} - x)$$

$$S_1(x) = \frac{z_1}{120h_1}(x_2 - x)^5 + \frac{z_2}{120h_1}(x - x_1)^5 + C_1^{(1)}(x - x_1)^3 + C_1^{(2)}(x_2 - x)^3 + C_1^{(3)}(x - x_1) + C_1^{(4)}(x_2 - x)$$

$$S_1(x_1) = \frac{z_1}{120h_1}(x_2 - x_1)^5 + \frac{z_2}{120h_1}(x_1 - x_1)^5 + C_1^{(1)}(x_1 - x_1)^3 + C_1^{(2)}(x_2 - x_1)^3 + C_1^{(3)}(x_1 - x_1) + C_1^{(4)}(x_2 - x_1)$$

$$S_1(x_1) = \frac{z_1}{120}h_1^{\ 4} + C_1^{(2)}h_1^{\ 3} + C_1^{(4)}h_1$$

$$\boxed{\frac{z_1}{120}h_1^{\ 4} + C_1^{(2)}h_1^{\ 3} + C_1^{(4)}h_1 = y_1}$$

S1(x2) = y2

$$S_1(x) = \frac{z_1}{120h_1}(x_2 - x)^5 + \frac{z_2}{120h_1}(x - x_1)^5 + C_1^{(1)}(x - x_1)^3 + C_1^{(2)}(x_2 - x)^3 + C_1^{(3)}(x - x_1) + C_1^{(4)}(x_2 - x)$$

$$S_1(x_2) = \frac{z_1}{120h_1}(x_2 - x_2)^5 + \frac{z_2}{120h_1}(x_2 - x_1)^5 + C_1^{(1)}(x_2 - x_1)^3 + C_1^{(2)}(x_2 - x_2)^3 + C_1^{(3)}(x_2 - x_1) + C_1^{(4)}(x_2 - x_2)$$

$$S_1(x_2) = \frac{z_2}{120}h_1^{\ 4} + C_1^{(1)}h_1^{\ 3} + C_1^{(3)}h_1$$

$$\boxed{\frac{z_2}{120}h_1^{\ 4} + C_1^{(1)}h_1^{\ 3} + C_1^{(3)}h_1 = y_2}$$

S2(x2) = y2

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + C_i^{(1)}(x - x_i)^3 + C_i^{(2)}(x_{i+1} - x)^3 + C_i^{(3)}(x - x_i) + C_i^{(4)}(x_{i+1} - x)$$

$$S_2(x) = \frac{z_2}{120h_2}(x_3 - x)^5 + \frac{z_3}{120h_{i2}}(x - x_2)^5 + C_2^{(1)}(x - x_2)^3 + C_2^{(2)}(x_3 - x)^3 + C_2^{(3)}(x - x_2) + C_2^{(4)}(x_3 - x)$$

$$S_2(x_2) = \frac{z_2}{120h_2}(x_3 - x_2)^5 + \frac{z_3}{120h_2}(x_2 - x_2)^5 + C_2^{(1)}(x_2 - x_2)^3 + C_2^{(2)}(x_3 - x_2)^3 + C_2^{(3)}(x_2 - x_2) + C_2^{(4)}(x_3 - x_2)$$

$$S_2(x_2) = \frac{z_2}{120}h_2^{\ 4} + C_2^{(2)}h_2^{\ 3} + C_2^{(4)}h_2$$

$$\boxed{\frac{z_2}{120} h_2{}^4 + C_2^{(2)} h_2{}^3 + C_2^{(4)} h_2 = y_2}$$

S2(x3) = y3

$$S_2(x) = \frac{z_2}{120h_2}(x_3 - x)^5 + \frac{z_3}{120h_{i2}}(x - x_2)^5 + C_2^{(1)}(x - x_2)^3 + C_2^{(2)}(x_3 - x)^3 + C_2^{(3)}(x - x_2) + C_2^{(4)}(x_3 - x)$$

$$S_2(x_3) = \frac{z_2}{120h_2}(x_3 - x_3)^5 + \frac{z_3}{120h_{i2}}(x_3 - x_2)^5 + C_2^{(1)}(x_3 - x_2)^3 + C_2^{(2)}(x_3 - x_3)^3 + C_2^{(3)}(x_3 - x_2) + C_2^{(4)}(x_3 - x_3)$$

$$S_2(x_3) = \frac{z_3}{120} h_2{}^4 + C_2^{(1)} h_2{}^3 + C_2^{(3)} h_2$$

$$\boxed{\frac{z_3}{120} h_2{}^4 + C_2^{(1)} h_2{}^3 + C_2^{(3)} h_2 = y_3}$$

S3(x3) = y3

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + C_i^{(1)}(x - x_i)^3 + C_i^{(2)}(x_{i+1} - x)^3 + C_i^{(3)}(x - x_i) + C_i^{(4)}(x_{i+1} - x)$$

$$S_3(x) = \frac{z_3}{120h_3}(x_4 - x)^5 + \frac{z_4}{120h_3}(x - x_3)^5 + C_3^{(1)}(x - x_3)^3 + C_3^{(2)}(x_4 - x)^3 + C_3^{(3)}(x - x_3) + C_3^{(4)}(x_4 - x)$$

$$S_3(x_3) = \frac{z_3}{120h_3}(x_4 - x_3)^5 + \frac{z_4}{120h_3}(x_3 - x_3)^5 + C_3^{(1)}(x_3 - x_3)^3 + C_3^{(2)}(x_4 - x_3)^3 + C_3^{(3)}(x_3 - x_3) + C_3^{(4)}(x_4 - x_3)$$

$$S_3(x_3) = \frac{z_3}{120} h_3{}^4 + C_3^{(2)} h_3{}^3 + C_3^{(4)} h_3$$

$$\boxed{\frac{z_3}{120} h_3{}^4 + C_3^{(2)} h_3{}^3 + C_3^{(4)} h_3 = y_3}$$

S3(x4) = y4

$$S_3(x) = \frac{z_3}{120h_3}(x_4 - x)^5 + \frac{z_4}{120h_3}(x - x_3)^5 + C_3^{(1)}(x - x_3)^3 + C_3^{(2)}(x_4 - x)^3 + C_3^{(3)}(x - x_3) + C_3^{(4)}(x_4 - x)$$

$$S_3(x_3) = \frac{z_3}{120h_3}(x_4 - x_3)^5 + \frac{z_4}{120h_3}(x_3 - x_3)^5 + C_3^{(1)}(x_3 - x_3)^3 + C_3^{(2)}(x_4 - x_3)^3 + C_3^{(3)}(x_3 - x_3) + C_3^{(4)}(x_4 - x_3)$$

$$\boxed{S_3(x_3) = \frac{z_3}{120}h_3{}^4 + C_3^{(2)}h_3{}^3 + C_3^{(4)}h_3}$$

S4(x4) = y4

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + C_i^{(1)}(x - x_i)^3 + C_i^{(2)}(x_{i+1} - x)^3 + C_i^{(3)}(x - x_i) + C_i^{(4)}(x_{i+1} - x)$$

$$S_4(x) = \frac{z_4}{120h_4}(x_5 - x)^5 + \frac{z_5}{120h_4}(x - x_4)^5 + C_4^{(1)}(x - x_4)^3 + C_4^{(2)}(x_5 - x)^3 + C_4^{(3)}(x - x_4) + C_4^{(4)}(x_5 - x)$$

$$S_4(x_4) = \frac{z_4}{120h_4}(x_5 - x_4)^5 + \frac{z_5}{120h_4}(x_4 - x_4)^5 + C_4^{(1)}(x_4 - x_4)^3 + C_4^{(2)}(x_5 - x_4)^3 + C_4^{(3)}(x_4 - x_4) + C_4^{(4)}(x_5 - x_4)$$

$$S_4(x_4) = \frac{z_4}{120}h_4{}^4 + C_4^{(2)}h_4{}^3 + C_4^{(4)}h_4$$

$$\boxed{\frac{z_4}{120}h_4{}^4 + C_4^{(2)}h_4{}^3 + C_4^{(4)}h_4 = y_4}$$

S4(x5) = y5

$$S_4(x) = \frac{z_4}{120h_4}(x_5 - x)^5 + \frac{z_5}{120h_4}(x - x_4)^5 + C_4^{(1)}(x - x_4)^3 + C_4^{(2)}(x_5 - x)^3 + C_4^{(3)}(x - x_4) + C_4^{(4)}(x_5 - x)$$

$$S_4(x_5) = \frac{z_4}{120h_4}(x_5 - x_5)^5 + \frac{z_5}{120h_4}(x_5 - x_4)^5 + C_4^{(1)}(x_5 - x_4)^3 + C_4^{(2)}(x_5 - x_5)^3 + C_4^{(3)}(x_5 - x_4) + C_4^{(4)}(x_5 - x_5)$$

$$S_4(x_5) = \frac{z_5}{120}h_4{}^4 + C_4^{(1)}h_4{}^3 + C_4^{(3)}h_4$$

$$\boxed{\frac{z_5}{120}h_4{}^4 + C_4^{(1)}h_4{}^3 + C_4^{(3)}h_4 = y_4}$$

S5(x5) = y5

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1} - x)^5 + \frac{z_{i+1}}{120h_i}(x - x_i)^5 + C_i^{(1)}(x - x_i)^3 + C_i^{(2)}(x_{i+1} - x)^3 + C_i^{(3)}(x - x_i) + C_i^{(4)}(x_{i+1} - x)$$

$$S_5(x) = \frac{z_5}{120h_5}(x_6 - x)^5 + \frac{z_6}{120h_5}(x - x_5)^5 + C_5^{(1)}(x - x_5)^3 + C_5^{(2)}(x_6 - x)^3 + C_5^{(3)}(x - x_5) + C_5^{(4)}(x_6 - x)$$

$$S_5(x_5) = \frac{z_5}{120h_5}(x_6 - x_5)^5 + \frac{z_6}{120h_5}(x_5 - x_5)^5 + C_5^{(1)}(x_5 - x_5)^3 + C_5^{(2)}(x_6 - x_5)^3 + C_5^{(3)}(x_5 - x_5) + C_5^{(4)}(x_6 - x_5)$$

$$S_5(x_5) = \frac{z_5}{120}h_5^{\,4} + C_5^{(2)}h_5^{\,3} + C_5^{(4)}h_5$$

$$\boxed{\frac{z_5}{120}h_5^{\,4} + C_5^{(2)}h_5^{\,3} + C_5^{(4)}h_5 = y_5}$$

A continuación vamos a sacar las condiciones de continuidad para cuatro nodos interiores en en la primera derivada de nuestro spline:

S’2(x2) = S’1(x2)
S’3(x3) = S’2(x3)
S’4(x4) = S’3(x4)
S’5(x5) = S’4(x5)

S’2(x2) = S’1(x2)

$$S_i'(x) = -\frac{z_i}{24h_i}(x_{i+1} - x)^4 + \frac{z_{i+1}}{24h_i}(x - x_i)^4 + 3C_i^{(1)}(x - x_i)^2 - 3C_i^{(2)}(x_{i+1} - x)^2 + C_i^{(3)} - C_i^{(4)}$$

$$S_2'(x) = -\frac{z_2}{24h_2}(x_3 - x)^4 + \frac{z_3}{24h_2}(x - x_2)^4 + 3C_2^{(1)}(x - x_2)^2 - 3C_2^{(2)}(x_3 - x)^2 + C_2^{(3)} - C_2^{(4)}$$

$$S_2'(x_2) = -\frac{z_2}{24h_2}(x_3 - x_2)^4 + \frac{z_3}{24h_2}(x_2 - x_2)^4 + 3C_2^{(1)}(x_2 - x_2)^2 - 3C_2^{(2)}(x_3 - x_2)^2 + C_2^{(3)} - C_2^{(4)}$$

$$S_2'(x_2) = -\frac{z_2}{24}h_2^{\,3} - 3C_2^{(2)}h_2^{\,2} + C_2^{(3)} - C_2^{(4)}$$

$$S_1'(x) = -\frac{z_1}{24h_1}(x_2 - x)^4 + \frac{z_2}{24h_1}(x - x_1)^4 + 3C_1^{(1)}(x - x_1)^2 - 3C_1^{(2)}(x_2 - x)^2 + C_1^{(3)} - C_1^{(4)}$$

$$S_1'(x_2) = -\frac{z_1}{24h_1}(x_2 - x_2)^4 + \frac{z_2}{24h_1}(x_2 - x_1)^4 + 3C_1^{(1)}(x_2 - x_1)^2 - 3C_1^{(2)}(x_2 - x_2)^2 + C_1^{(3)} - C_1^{(4)}$$

$$S_1'(x_2) = +\frac{z_2}{24}h_1^{\,3} + 3C_1^{(1)}h_1^{\,2} + C_1^{(3)} - C_1^{(4)}$$

Igualando ambas partes queda:

$$-\frac{z_2}{24}h_2^{\ 3}-3C_2^{(2)}h_2^{\ 2}+C_2^{(3)}-C_2^{(4)}=\frac{z_2}{24}h_1^{\ 3}+3C_1^{(1)}h_1^{\ 2}+C_1^{(3)}-C_1^{(4)}$$

$$\frac{z_2}{24}h_2^{\ 3}+\frac{z_2}{24}h_1^{\ 3}+3C_1^{(1)}h_1^{\ 2}+C_1^{(3)}-C_1^{(4)}+3C_2^{(2)}h_2^{\ 2}-C_2^{(3)}+C_2^{(4)}=0$$

$$\boxed{\left(\frac{h_2^{\ 3}}{24}+\frac{h_1^{\ 3}}{24}\right)z_2+3C_1^{(1)}h_1^{\ 2}+C_1^{(3)}-C_1^{(4)}+3C_2^{(2)}h_2^{\ 2}-C_2^{(3)}+C_2^{(4)}=0}$$

S’3(x3) = S’2(x3)

$$S_i'(x)=-\frac{z_i}{24h_i}(x_{i+1}-x)^4+\frac{z_{i+1}}{24h_i}(x-x_i)^4+3C_i^{(1)}(x-x_i)^2-3C_i^{(2)}(x_{i+1}-x)^2+C_i^{(3)}-C_i^{(4)}$$

$$S_3'(x)=-\frac{z_3}{24h_3}(x_4-x)^4+\frac{z_4}{24h_3}(x-x_3)^4+3C_3^{(1)}(x-x_3)^2-3C_3^{(2)}(x_4-x)^2+C_3^{(3)}-C_3^{(4)}$$

$$S_3'(x_3)=-\frac{z_3}{24h_3}(x_4-x_3)^4+\frac{z_4}{24h_3}(x_3-x_3)^4+3C_3^{(1)}(x_3-x_3)^2-3C_3^{(2)}(x_4-x_3)^2+C_3^{(3)}-C_3^{(4)}$$

$$S_3'(x_3)=-\frac{z_3}{24}h_3^{\ 3}-3C_3^{(2)}h_3^{\ 2}+C_3^{(3)}-C_3^{(4)}$$

$$S_2'(x)=-\frac{z_2}{24h_2}(x_3-x)^4+\frac{z_3}{24h_2}(x-x_2)^4+3C_2^{(1)}(x-x_2)^2-3C_2^{(2)}(x_3-x)^2+C_2^{(3)}-C_2^{(4)}$$

$$S_2'(x_3)=-\frac{z_2}{24h_2}(x_3-x_3)^4+\frac{z_3}{24h_2}(x_3-x_2)^4+3C_2^{(1)}(x_3-x_2)^2-3C_2^{(2)}(x_3-x_3)^2+C_2^{(3)}-C_2^{(4)}$$

$$S_2'(x_3)=\frac{z_3}{24}h_2^{\ 3}+3C_2^{(1)}h_2^{\ 2}+C_2^{(3)}-C_2^{(4)}$$

Igualando ambas partes queda:

$$-\frac{z_3}{24}h_3^{\ 3}-3C_3^{(2)}h_3^{\ 2}+C_3^{(3)}-C_3^{(4)}=\frac{z_3}{24}h_2^{\ 3}+3C_2^{(1)}h_2^{\ 2}+C_2^{(3)}-C_2^{(4)}$$

$$\frac{z_3}{24}h_2^{\ 3}+\frac{z_3}{24}h_3^{\ 3}+3C_2^{(1)}h_2^{\ 2}+C_2^{(3)}-C_2^{(4)}+3C_3^{(2)}h_3^{\ 2}-C_3^{(3)}+C_3^{(4)}=0$$

$$\boxed{\left(\frac{h_2^{\ 3}}{24}+\frac{h_3^{\ 3}}{24}\right)z_3+3C_2^{(1)}h_2^{\ 2}+C_2^{(3)}-C_2^{(4)}+3C_3^{(2)}h_3^{\ 2}-C_3^{(3)}+C_3^{(4)}=0}$$

S’4(x4) = S’3(x4)

$$S_i'(x) = -\frac{z_i}{24h_i}(x_{i+1}-x)^4 + \frac{z_{i+1}}{24h_i}(x-x_i)^4 + 3C_i^{(1)}(x-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x)^2 + C_i^{(3)} - C_i^{(4)}$$

$$S_4'(x) = -\frac{z_4}{24h_4}(x_5-x)^4 + \frac{z_5}{24h_4}(x-x_4)^4 + 3C_4^{(1)}(x-x_4)^2 - 3C_4^{(2)}(x_5-x)^2 + C_4^{(3)} - C_4^{(4)}$$

$$S_4'(x_4) = -\frac{z_4}{24h_4}(x_5-x_4)^4 + \frac{z_5}{24h_4}(x_4-x_4)^4 + 3C_4^{(1)}(x_4-x_4)^2 - 3C_4^{(2)}(x_5-x_4)^2 + C_4^{(3)} - C_4^{(4)}$$

$$S_4'(x_4) = -\frac{z_4}{24}h_4^{\;3} - 3C_4^{(2)}h_4^{\;2} + C_4^{(3)} - C_4^{(4)}$$

$$S_3'(x) = -\frac{z_3}{24h_3}(x_4-x)^4 + \frac{z_4}{24h_3}(x-x_3)^4 + 3C_3^{(1)}(x-x_3)^2 - 3C_3^{(2)}(x_4-x)^2 + C_3^{(3)} - C_3^{(4)}$$

$$S_3'(x_4) = -\frac{z_3}{24h_3}(x_4-x_4)^4 + \frac{z_4}{24h_3}(x_4-x_3)^4 + 3C_3^{(1)}(x_4-x_3)^2 - 3C_3^{(2)}(x_4-x_4)^2 + C_3^{(3)} - C_3^{(4)}$$

$$S_3'(x_4) = \frac{z_4}{24}h_3^{\;3} + 3C_3^{(1)}h_3^{\;2} + C_3^{(3)} - C_3^{(4)}$$

Igualando ambas partes queda:

$$-\frac{z_4}{24}h_4^{\;3} - 3C_4^{(2)}h_4^{\;2} + C_4^{(3)} - C_4^{(4)} = \frac{z_4}{24}h_3^{\;3} + 3C_3^{(1)}h_3^{\;2} + C_3^{(3)} - C_3^{(4)}$$

$$\frac{z_4}{24}h_3^{\;3} + \frac{z_4}{24}h_4^{\;3} + 3C_3^{(1)}h_3^{\;2} + 3C_4^{(2)}h_4^{\;2} + C_3^{(3)} - C_3^{(4)} - C_4^{(3)} + C_4^{(4)} = 0$$

$$\boxed{\left(\frac{h_3^{\;3}}{24} + \frac{h_4^{\;3}}{24}\right)z_4 + 3C_3^{(1)}h_3^{\;2} + 3C_4^{(2)}h_4^{\;2} + C_3^{(3)} - C_3^{(4)} - C_4^{(3)} + C_4^{(4)} = 0}$$

S’5(x5) = S’4(x5)

$$S_i'(x) = -\frac{z_i}{24h_i}(x_{i+1}-x)^4 + \frac{z_{i+1}}{24h_i}(x-x_i)^4 + 3C_i^{(1)}(x-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x)^2 + C_i^{(3)} - C_i^{(4)}$$

$$S_5'(x) = -\frac{z_5}{24h_5}(x_6-x)^4 + \frac{z_6}{24h_5}(x-x_5)^4 + 3C_5^{(1)}(x-x_5)^2 - 3C_5^{(2)}(x_6-x)^2 + C_5^{(3)} - C_5^{(4)}$$

$$S_5'(x_5) = -\frac{z_5}{24h_5}(x_6 - x_5)^4 + \frac{z_6}{24h_5}(x_5 - x_5)^4 + 3C_5^{(1)}(x_5 - x_5)^2 - 3C_5^{(2)}(x_6 - x_5)^2 + C_5^{(3)} - C_5^{(4)}$$

$$S_5'(x_5) = -\frac{z_5}{24}h_5^{\ 3} - 3C_5^{(2)}h_5^{\ 2} + C_5^{(3)} - C_5^{(4)}$$

$$S_4'(x) = -\frac{z_4}{24h_4}(x_5 - x)^4 + \frac{z_5}{24h_4}(x - x_4)^4 + 3C_4^{(1)}(x - x_4)^2 - 3C_4^{(2)}(x_5 - x)^2 + C_4^{(3)} - C_4^{(4)}$$

$$S_4'(x_5) = -\frac{z_4}{24h_4}(x_5 - x_5)^4 + \frac{z_5}{24h_4}(x_5 - x_4)^4 + 3C_4^{(1)}(x_5 - x_4)^2 - 3C_4^{(2)}(x_5 - x_5)^2 + C_4^{(3)} - C_4^{(4)}$$

$$S_4'(x_5) = +\frac{z_5}{24}h_4^{\ 3} + 3C_4^{(1)}h_4^{\ 2} + C_4^{(3)} - C_4^{(4)}$$

Igualando ambas partes queda:

$$-\frac{z_5}{24}h_5^{\ 3} - 3C_5^{(2)}h_5^{\ 2} + C_5^{(3)} - C_5^{(4)} = \frac{z_5}{24}h_4^{\ 3} + 3C_4^{(1)}h_4^{\ 2} + C_4^{(3)} - C_4^{(4)}$$

$$\frac{z_5}{24}h_4^{\ 4} + \frac{z_5}{24}h_5^{\ 3} + 3C_4^{(1)}h_4^{\ 2} + 3C_5^{(2)}h_5^{\ 2} + C_4^{(3)} - C_4^{(4)} - C_5^{(3)} + C_5^{(4)} = 0$$

$$\boxed{\left(\frac{h_4^{\ 4}}{24} + \frac{h_5^{\ 3}}{24}\right)z_5 + 3C_4^{(1)}h_4^{\ 2} + 3C_5^{(2)}h_5^{\ 2} + C_4^{(3)} - C_4^{(4)} - C_5^{(3)} + C_5^{(4)} = 0}$$

Condiciones de continuidad de la primera derivada del spline para los cuatro nodos.

$$\boxed{\left(\frac{h_1^{\ 3}}{24} + \frac{h_2^{\ 3}}{24}\right)z_2 + C_1^{(1)}h_1^{\ 2} + C_1^{(3)} - C_1^{(4)} + 3C_2^{(2)}h_2^{\ 2} - C_2^{(3)} + C_2^{(4)} = 0}$$

$$\boxed{\left(\frac{h_2^{\ 3}}{24} + \frac{h_3^{\ 3}}{24}\right)z_3 + 3C_2^{(1)}h_2^{\ 2} + C_2^{(3)} - C_2^{(4)} + 3C_3^{(2)}h_3^{\ 2} - C_3^{(3)} + C_3^{(4)} = 0}$$

$$\boxed{\left(\frac{h_3^{\ 3}}{24} + \frac{h_4^{\ 3}}{24}\right)z_4 + 3C_3^{(1)}h_3^{\ 2} + 3C_4^{(2)}h_4^{\ 2} + C_3^{(3)} - C_3^{(4)} - C_4^{(3)} + C_4^{(4)} = 0}$$

$$\boxed{\left(\frac{h_4^{\ 4}}{24} + \frac{h_5^{\ 3}}{24}\right)z_5 + 3C_4^{(1)}h_4^{\ 2} + 3C_5^{(2)}h_5^{\ 2} + C_4^{(3)} - C_4^{(4)} - C_5^{(3)} + C_5^{(4)} = 0}$$

Seguimos exponiendo las condiciones de continuidad, esta vez en los nodos interiores de la segunda derivada.

S''2(x2) = S''1(x2)
S''3(x3) = S''2(x3)
S''4(x4) = S''3(x4)
S''5(x5) = S''4(x5)

S''2(x2) = S''1(x2)

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

$$S_2''(x) = \frac{z_2}{6h_2}(x_3-x)^3 + \frac{z_3}{6h_2}(x-x_2)^3 + 6C_2^{(1)}(x-x_2) + 6C_2^{(2)}(x_3-x)$$

$$S_2''(x_2) = \frac{z_2}{6h_2}(x_3-x_2)^3 + \frac{z_3}{6h_2}(x_2-x_2)^3 + 6C_2^{(1)}(x_2-x_2) + 6C_2^{(2)}(x_3-x_2)$$

$$S_2''(x_2) = \frac{z_2}{6}h_2^{\,2} + 6C_2^{(2)}h_2$$

$$S_1''(x) = \frac{z_1}{6h_1}(x_2-x)^3 + \frac{z_2}{6h_1}(x-x_1)^3 + 6C_1^{(1)}(x-x_1) + 6C_1^{(2)}(x_2-x)$$

$$S_1''(x_2) = \frac{z_1}{6h_1}(x_2-x_2)^3 + \frac{z_2}{6h_1}(x_2-x_1)^3 + 6C_1^{(1)}(x_2-x_1) + 6C_1^{(2)}(x_2-x_2)$$

$$S_1''(x_2) = \frac{z_2}{6}h_1^{\,2} + 6C_1^{(1)}h_1$$

Igualando ambas partes queda:

$$\frac{z_2}{6}h_1^{\,2} + 6C_1^{(1)}h_1 = \frac{z_2}{6}h_2^{\,2} + 6C_2^{(2)}h_2$$

$$\frac{z_2}{6}h_2^{\,2} - \frac{z_2}{6}h_1^{\,2} + 6C_2^{(2)}h_2 - 6C_1^{(1)}h_1 = 0$$

$$\boxed{\left(\frac{h_2^{\,2}}{6} - \frac{h_1^{\,2}}{6}\right)z_2 + 6C_2^{(2)}h_2 - 6C_1^{(1)}h_1 = 0}$$

S’’3(x3) = S’’2(x3)

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

$$S_3''(x) = \frac{z_3}{6h_3}(x_4-x)^3 + \frac{z_4}{6h_3}(x-x_3)^3 + 6C_3^{(1)}(x-x_3) + 6C_3^{(2)}(x_4-x)$$

$$S_3''(x_3) = \frac{z_3}{6h_3}(x_4-x_3)^3 + \frac{z_4}{6h_3}(x_3-x_3)^3 + 6C_3^{(1)}(x_3-x_3) + 6C_3^{(2)}(x_4-x_3)$$

$$S_3''(x_3) = \frac{z_3}{6}h_3^{\ 2} + 6C_3^{(2)}h_3$$

$$S_2''(x_3) = \frac{z_2}{6h_2}(x_3-x_3)^3 + \frac{z_3}{6h_2}(x_3-x_2)^3 + 6C_2^{(1)}(x_3-x_2) + 6C_2^{(2)}(x_3-x_3)$$

$$S_2''(x_3) = \frac{z_3}{6}h_2^{\ 2} + 6C_2^{(1)}h_2$$

Igualando ambas partes queda:

$$\frac{z_3}{6}h_3^{\ 2} + 6C_3^{(2)}h_3 = \frac{z_3}{6}h_2^{\ 2} + 6C_2^{(1)}h_2$$

$$\frac{z_3}{6}h_3^{\ 2} - \frac{z_3}{6}h_2^{\ 2} + 6C_3^{(2)}h_3 - 6C_2^{(1)}h_2 = 0$$

$$\boxed{\left(\frac{h_3^{\ 2}}{6} - \frac{h_2^{\ 2}}{6}\right)z_3 + 6C_3^{(2)}h_3 - 6C_2^{(1)}h_2 = 0}$$

S’’4(x4) = S’’3(x4)

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

$$S_4''(x) = \frac{z_4}{6h_4}(x_5-x)^3 + \frac{z_5}{6h_4}(x-x_4)^3 + 6C_4^{(1)}(x-x_4) + 6C_4^{(2)}(x_5-x)$$

$$S_4''(x_4) = \frac{z_4}{6h_4}(x_5-x_4)^3 + \frac{z_5}{6h_4}(x_4-x_4)^3 + 6C_4^{(1)}(x_4-x_4) + 6C_4^{(2)}(x_5-x_4)$$

$$S_4''(x_4) = \frac{z_4}{6}h_4^{\ 2} + 6C_4^{(2)}h_4$$

$$S_3''(x)=\frac{z_3}{6h_3}(x_4-x)^3+\frac{z_4}{6h_3}(x-x_3)^3+6C_3^{(1)}(x-x_3)+6C_3^{(2)}(x_4-x)$$

$$S_3''(x)=\frac{z_3}{6h_3}(x_4-x)^3+\frac{z_4}{6h_3}(x-x_3)^3+6C_3^{(1)}(x-x_3)+6C_3^{(2)}(x_4-x)$$

$$S_3''(x_4)=\frac{z_3}{6h_3}(x_4-x_4)^3+\frac{z_4}{6h_3}(x_4-x_3)^3+6C_3^{(1)}(x_4-x_3)+6C_3^{(2)}(x_4-x_4)$$

$$S_3''(x_4)=\frac{z_4}{6}h_3{}^2+6C_3^{(1)}h_3$$

Igualando ambas partes queda:

$$\frac{z_4}{6}h_4{}^2+6C_4^{(2)}h_4=\frac{z_4}{6}h_3{}^2+6C_3^{(1)}h_3$$

$$\frac{z_4}{6}h_4{}^2-\frac{z_4}{6}h_3{}^2+6C_4^{(2)}h_4-6C_3^{(1)}h_3=0$$

$$\boxed{\left(\frac{h_4{}^2}{6}-\frac{h_3{}^2}{6}\right)z_4+6C_4^{(2)}h_4-6C_3^{(1)}h_3=0}$$

S’’5(x5) = S’’4(x5)

$$S_i''(x)=\frac{z_i}{6h_i}(x_{i+1}-x)^3+\frac{z_{i+1}}{6h_i}(x-x_i)^3+6C_i^{(1)}(x-x_i)+6C_i^{(2)}(x_{i+1}-x)$$

$$S_5''(x)=\frac{z_5}{6h_5}(x_6-x)^3+\frac{z_6}{6h_5}(x-x_5)^3+6C_5^{(1)}(x-x_5)+6C_5^{(2)}(x_6-x)$$

$$S_5''(x_5)=\frac{z_5}{6h_5}(x_6-x_5)^3+\frac{z_6}{6h_5}(x_5-x_5)^3+6C_5^{(1)}(x_5-x_5)+6C_5^{(2)}(x_6-x_5)$$

$$S_5''(x_5)=\frac{z_5}{6}h_5{}^2+6C_5^{(2)}h_5$$

$$S_4''(x_5)=\frac{z_4}{6h_4}(x_5-x_5)^3+\frac{z_5}{6h_4}(x_5-x_4)^3+6C_4^{(1)}(x_5-x_4)+6C_4^{(2)}(x_5-x_5)$$

$$S_4''(x_5)=\frac{z_5}{6}h_4{}^2+6C_4^{(1)}h_4$$

Igualando ambas partes queda:

$$\frac{z_5}{6}h_5^{\ 2}+6C_5^{(2)}h_5=\frac{z_5}{6}h_4^{\ 2}+6C_4^{(1)}h_4$$

$$\frac{z_5}{6}h_5^{\ 2}-\frac{z_5}{6}h_4^{\ 2}+6C_5^{(2)}h_5-6C_4^{(1)}h_4=0$$

$$\boxed{\left(\frac{h_5^{\ 2}}{6}-\frac{h_4^{\ 2}}{6}\right)z_5+6C_5^{(2)}h_5-6C_4^{(1)}h_4=0}$$

Condiciones de continuidad de la segunda derivada del spline para los cuatro nodos.

$$\boxed{\left(\frac{h_2^{\ 2}}{6}-\frac{h_1^{\ 2}}{6}\right)z_2+6C_2^{(2)}h_2-6C_1^{(1)}h_1=0}$$

$$\boxed{\left(\frac{h_3^{\ 2}}{6}-\frac{h_2^{\ 2}}{6}\right)z_3+6C_3^{(2)}h_3-6C_2^{(1)}h_2=0}$$

$$\boxed{\left(\frac{h_4^{\ 2}}{6}-\frac{h_3^{\ 2}}{6}\right)z_4+6C_4^{(2)}h_4-6C_3^{(1)}h_3=0}$$

$$\boxed{\left(\frac{h_5^{\ 2}}{6}-\frac{h_4^{\ 2}}{6}\right)z_5+6C_5^{(2)}h_5-6C_4^{(1)}h_4=0}$$

Y por último exponemos las condiciones de continuidad en los nodos interiores para la tercera derivada.

S'''2(x2) = S'''1(x2)
S'''3(x3) = S'''2(x3)
S'''4(x4) = S'''3(x4)
S'''5(x5) = S'''4(x5)

S'''2(x2) = S'''1(x2)

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1}-x)^2 + \frac{z_{i+1}}{2h_i}(x-x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_2'''(x) = -\frac{z_2}{2h_2}(x_3-x)^2 + \frac{z_3}{2h_2}(x-x_2)^2 + 6C_2^{(1)} - 6C_2^{(2)}$$

$$S_2'''(x_2) = -\frac{z_2}{2h_2}(x_3-x_2)^2 + \frac{z_3}{2h_2}(x_2-x_2)^2 + 6C_2^{(1)} - 6C_2^{(2)}$$

$$S_2'''(x_2) = -\frac{z_2}{2}h_2 + 6C_2^{(1)} - 6C_2^{(2)}$$

$$S_1'''(x) = -\frac{z_1}{2h_1}(x_2-x)^2 + \frac{z_2}{2h_1}(x-x_1)^2 + 6C_1^{(1)} - 6C_1^{(2)}$$

$$S_1'''(x_2) = -\frac{z_1}{2h_1}(x_2-x_2)^2 + \frac{z_2}{2h_1}(x_2-x_1)^2 + 6C_1^{(1)} - 6C_1^{(2)}$$

$$S_1'''(x_2) = \frac{z_2}{2}h_1 + 6C_1^{(1)} - 6C_1^{(2)}$$

Igualando ambas partes queda:

$$-\frac{z_2}{2}h_2 + 6C_2^{(1)} - 6C_2^{(2)} = \frac{z_2}{2}h_1 + 6C_1^{(1)} - 6C_1^{(2)}$$

$$\frac{z_2}{2}h_1 + \frac{z_2}{2}h_2 + 6C_1^{(1)} - 6C_1^{(2)} - 6C_2^{(1)} + 6C_2^{(2)} = 0$$

$$\boxed{\left(\frac{h_1}{2} + \frac{h_2}{2}\right)z_2 + 6C_1^{(1)} - 6C_1^{(2)} - 6C_2^{(1)} + 6C_2^{(2)} = 0}$$

S'''3(x3) = S'''2(x3)

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1} - x)^2 + \frac{z_{i+1}}{2h_i}(x - x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_3'''(x) = -\frac{z_3}{2h_3}(x_4 - x)^2 + \frac{z_4}{2h_3}(x - x_3)^2 + 6C_3^{(1)} - 6C_3^{(2)}$$

$$S_3'''(x_3) = -\frac{z_3}{2h_3}(x_4 - x_3)^2 + \frac{z_4}{2h_3}(x_3 - x_3)^2 + 6C_3^{(1)} - 6C_3^{(2)}$$

$$S_3'''(x_3) = -\frac{z_3}{2}h_3 + 6C_3^{(1)} - 6C_3^{(2)}$$

$$S_2'''(x) = -\frac{z_2}{2h_2}(x_3 - x)^2 + \frac{z_3}{2h_2}(x - x_2)^2 + 6C_2^{(1)} - 6C_2^{(2)}$$

$$S_2'''(x_3) = -\frac{z_2}{2h_2}(x_3 - x_3)^2 + \frac{z_3}{2h_2}(x_3 - x_2)^2 + 6C_2^{(1)} - 6C_2^{(2)}$$

$$S_2'''(x_3) = \frac{z_3}{2}h_2 + 6C_2^{(1)} - 6C_2^{(2)}$$

Igualando ambas partes queda:

$$-\frac{z_3}{2}h_3 + 6C_3^{(1)} - 6C_3^{(2)} = \frac{z_3}{2}h_2 + 6C_2^{(1)} - 6C_2^{(2)}$$

$$\frac{z_3}{2}h_2 + \frac{z_3}{2}h_3 + 6C_2^{(1)} - 6C_2^{(2)} - 6C_3^{(1)} + 6C_3^{(2)} = 0$$

$$\boxed{\left(\frac{h_2}{2} + \frac{h_3}{2}\right)z_3 + 6C_2^{(1)} - 6C_2^{(2)} - 6C_3^{(1)} + 6C_3^{(2)} = 0}$$

S’’’4(x4) = S’’’3(x4)

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1} - x)^2 + \frac{z_{i+1}}{2h_i}(x - x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_4'''(x) = -\frac{z_4}{2h_4}(x_5 - x)^2 + \frac{z_5}{2h_4}(x - x_4)^2 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$S_4'''(x_4) = -\frac{z_4}{2h_4}(x_5 - x_4)^2 + \frac{z_5}{2h_4}(x_4 - x_4)^2 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$S_4'''(x_4) = -\frac{z_4}{2} h_4 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$S_3'''(x) = -\frac{z_3}{2h_3}(x_4 - x)^2 + \frac{z_4}{2h_3}(x - x_3)^2 + 6C_3^{(1)} - 6C_3^{(2)}$$

$$S_3'''(x_4) = -\frac{z_3}{2h_3}(x_4 - x_4)^2 + \frac{z_4}{2h_3}(x_4 - x_3)^2 + 6C_3^{(1)} - 6C_3^{(2)}$$

$$S_3'''(x_4) = +\frac{z_4}{2} h_3 + 6C_3^{(1)} - 6C_3^{(2)}$$

Igualando ambas partes queda:

$$-\frac{z_4}{2} h_4 + 6C_4^{(1)} - 6C_4^{(2)} = \frac{z_4}{2} h_3 + 6C_3^{(1)} - 6C_3^{(2)}$$

$$\frac{z_4}{2} h_3 + \frac{z_4}{2} h_4 + 6C_3^{(1)} - 6C_3^{(2)} - 6C_4^{(1)} + 6C_4^{(2)} = 0$$

$$\boxed{\left(\frac{h_3}{2} + \frac{h_4}{2}\right) z_4 + 6C_3^{(1)} - 6C_3^{(2)} - 6C_4^{(1)} + 6C_4^{(2)} = 0}$$

S’’’5(x5) = S’’’4(x5)

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1} - x)^2 + \frac{z_{i+1}}{2h_i}(x - x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_5'''(x) = -\frac{z_5}{2h_5}(x_6 - x)^2 + \frac{z_6}{2h_5}(x - x_5)^2 + 6C_5^{(1)} - 6C_5^{(2)}$$

$$S_5'''(x_5) = -\frac{z_5}{2h_5}(x_6 - x_5)^2 + \frac{z_6}{2h_5}(x_5 - x_5)^2 + 6C_5^{(1)} - 6C_5^{(2)}$$

$$S_5'''(x_5) = -\frac{z_5}{2} h_5 + 6C_5^{(1)} - 6C_5^{(2)}$$

$$S_4'''(x) = -\frac{z_4}{2h_4}(x_5 - x)^2 + \frac{z_5}{2h_4}(x - x_4)^2 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$S_4'''(x_5) = -\frac{z_4}{2h_4}(x_5 - x_5)^2 + \frac{z_5}{2h_4}(x_5 - x_4)^2 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$S_4'''(x_5) = -\frac{z_4}{2h_4}(x_5 - x_5)^2 + \frac{z_5}{2h_4}(x_5 - x_4)^2 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$S_4'''(x_5) = \frac{z_5}{2}h_4 + 6C_4^{(1)} - 6C_4^{(2)}$$

Igualando ambas partes queda:

$$-\frac{z_5}{2}h_5 + 6C_5^{(1)} - 6C_5^{(2)} = \frac{z_5}{2}h_4 + 6C_4^{(1)} - 6C_4^{(2)}$$

$$\frac{z_5}{2}h_4 + \frac{z_5}{2}h_5 + 6C_4^{(1)} - 6C_4^{(2)} - 6C_5^{(1)} + 6C_5^{(2)} = 0$$

$$\boxed{\left(\frac{h_4}{2} + \frac{h_5}{2}\right)z_5 + 6C_4^{(1)} - 6C_4^{(2)} - 6C_5^{(1)} + 6C_5^{(2)} = 0}$$

Condiciones de continuidad de la tercera derivada del spline para los cuatro nodos.

$$\boxed{\left(\frac{h_1}{2} + \frac{h_2}{2}\right)z_2 + 6C_1^{(1)} - 6C_1^{(2)} - 6C_2^{(1)} + 6C_2^{(2)} = 0}$$

$$\boxed{\left(\frac{h_2}{2} + \frac{h_3}{2}\right)z_3 + 6C_2^{(1)} - 6C_2^{(2)} - 6C_3^{(1)} + 6C_3^{(2)} = 0}$$

$$\boxed{\left(\frac{h_3}{2} + \frac{h_4}{2}\right)z_4 + 6C_3^{(1)} - 6C_3^{(2)} - 6C_4^{(1)} + 6C_4^{(2)} = 0}$$

$$\boxed{\left(\frac{h_4}{2} + \frac{h_5}{2}\right)z_5 + 6C_4^{(1)} - 6C_4^{(2)} - 6C_5^{(1)} + 6C_5^{(2)} = 0}$$

MATRIZ

A continuación vamos a exponer nuestro sistema de forma matricial con intención de resolverlo.

Utilizamos las condiciones de continuidad aplicadas en los nodos internos de nuestras derivadas y los valores impuestos para cada nodo en el Spline.

También impondremos 4 condiciones extra para definir nuestro Spline y que el sistema sea compatible determinado y poder obtener así nuestro Spline.

Las condiciones son las siguientes:

S1''(x1) = d2a

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

$$S_1''(x) = \frac{z_1}{6h_1}(x_2-x)^3 + \frac{z_2}{6h_1}(x-x_1)^3 + 6C_1^{(1)}(x-x_1) + 6C_1^{(2)}(x_2-x)$$

$$S_1''(x_1) = \frac{z_1}{6h_1}(x_2-x_1)^3 + \frac{z_2}{6h_1}(x_1-x_1)^3 + 6C_1^{(1)}(x_1-x_1) + 6C_1^{(2)}(x_2-x_1)$$

$$S_1''(x_1) = \frac{z_1}{6}h_1^{\ 2} + 6C_1^{(2)}h_1$$

$$\boxed{\frac{z_1}{6}h_1^{\ 2} + 6C_1^{(2)}h_1 = d2a}$$

S''(n) = d2b

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-x_i)^3 + 6C_i^{(1)}(x-x_i) + 6C_i^{(2)}(x_{i+1}-x)$$

$$S_{n-1}''(x) = \frac{z_{n-1}}{6h_{n-1}}(x_n-x)^3 + \frac{z_n}{6h_{n-1}}(x-x_{n-1})^3 + 6C_{n-1}^{(1)}(x-x_{n-1}) + 6C_{n-1}^{(2)}(x_n-x)$$

$$S_{n-1}''(x_n) = \frac{z_{n-1}}{6h_{n-1}}(x_n-x_n)^3 + \frac{z_n}{6h_{n-1}}(x_n-x_{n-1})^3 + 6C_{n-1}^{(1)}(x_n-x_{n-1}) + 6C_{n-1}^{(2)}(x_n-x_n)$$

$$S_{n-1}''(x_n) = \frac{z_n}{6}h_{n-1}^{\ 2} + 6C_{n-1}^{(1)}h_{n-1}$$

$$\boxed{\frac{z_n}{6} h_{n-1}{}^{2} + 6C_{n-1}^{(1)} h_{n-1} = d2b}$$

S1’’’’(x1) = d4a

$$S_i''''(x) = z_i \frac{(x_{i+1} - x)}{hi} + z_{i+1} \frac{(x - x_i)}{hi}$$

$$S_1''''(x) = z_1 \frac{(x_2 - x)}{h_1} + z_2 \frac{(x - x_1)}{h_1}$$

$$S_1''''(x_1) = z_1 \frac{(x_2 - x_1)}{h_1} + z_2 \frac{(x_1 - x_1)}{h_1}$$

$$S_1''''(x_1) = d4a$$

S’’’’(xn) = d4b

$$S_i''''(x) = z_i \frac{(x_{i+1} - x)}{hi} + z_{i+1} \frac{(x - x_i)}{hi}$$

$$S_{n-1}''''(x) = z_{n-1} \frac{(x_n - x)}{h_{n-1}} + z_n \frac{(x - x_{n-1})}{h_{n-1}}$$

$$S_{n-1}''''(x_n) = z_{n-1} \frac{(x_n - x_n)}{h_{n-1}} + z_n \frac{(x_n - x_{n-1})}{h_{n-1}}$$

$$S_{n-1}''''(x_n) = d4b$$

Para ello vamos a ordenar nuestras condiciones de izquierda a derecha según los nodos a los que condicionan. Además agruparemos los términos de estas para poder mostrarlo todo en un sistema matricial dando así una matriz con los parámetros de nuestro Spline, un vector de incógnitas y un vector de términos independientes.

Recuadradas quedan las condiciones simplificadas, empezando de izquierda a derecha:

S1(x1) = y1

$$\boxed{\frac{z_1}{120} h_1{}^{4} + C_1^{(2)} h_1{}^{3} + C_1^{(4)} h_1 = y_1}$$

S1’’(x1) = d2a

$$\frac{z_1}{6}h_1^{\,2} + 6C_1^{(2)}h_1 = d2a$$

S1''''(x1) = d2b

S_1创创$x_1() = d4a$

S'2(x2) = S'1(x2)

$$\left(\frac{h_1^{\,3}}{24} + \frac{h_2^{\,3}}{24}\right)z_2 + 3C_1^{(1)}h_1^{\,2} + C_1^{(3)} - C_1^{(4)} + 3C_2^{(2)}h_2^{\,2} - C_2^{(3)} + C_2^{(4)} = 0$$

S''2(x2) = S''1(x2)

$$\left(\frac{h_2^{\,2}}{6} - \frac{h_1^{\,2}}{6}\right)z_2 + 6C_2^{(2)}h_2 - 6C_1^{(1)}h_1 = 0$$

S'''2(x2) = S'''1(x2)

$$\left(\frac{h_1}{2} + \frac{h_2}{2}\right)z_2 + 6C_1^{(1)} - 6C_1^{(2)} - 6C_2^{(1)} + 6C_2^{(2)} = 0$$

S1(x2) = y2

$$\frac{z_2}{120}h_1^{\,4} + C_1^{(1)}h_1^{\,3} + C_1^{(3)}h_1 = y_2$$

Exponemos en una matriz las condiciones anteriormente expuestas el siguiente orden:

S1''(x1)
S1''''(x1)
S1(x1)
S'2(x2)=S'1(x2)
S''2(x2)=S''1(x2)
S'''2(x2)=S'''1(x2)
S1(x2)

Las dos condiciones impuestas en los extremos quedan:

h1^4/120	0	h1^3	0	h1	0	0	0	0	0
h1^2/6	0	6h1	0	0	0	0	0	0	0

A este pequeño bloque de condiciones en los extremos lo llamaremos A1

El resto de condiciones anteriormente expuestas entre el nodo 1 y dos queda:

h1^4/120	0	h1^3	0	h1	0	0	0	0	0
0	3h1^2	0	1	-1	(h2^3/24) + (h1^3/24)	0	3*h2^2	-1	1
0	"-6h1	0	0	0	(h2^2/6) - (h1^2/6)	0	6*h2	0	0
0	6	"-6	0	0	(h2/2) + (h1/2)	"-6	6	0	0
0	h1^3	0	h1	0	h1^4/120	0	0	0	0

A este bloque lo llamaremos Bloque A1,2.

Quedando el vector de incógnitas:

(z1,C1(1),C1(2),C1(3),C1(4),z1,C1(1),C1(2),C1(3),C1(4))

Y el vector de valores independientes:

(d2a,d4a,y1,0,0,0,z2)

A continuación vamos a exponer las condiciones que impondríamos a los siguientes 4 nodos para poder así estudiar la estructura que tendría nuestra matriz.

S2(x2) = y2

$$\boxed{\frac{z_2}{120}h_2^{\,4} + C_2^{(2)}h_2^{\,3} + C_2^{(4)}h_2 = y_2}$$

S’3(x3) = S’2(x3)

$$\boxed{\left(\frac{h_2^{\,3}}{24} + \frac{h_3^{\,3}}{24}\right)z_3 + 3C_2^{(1)}h_2^{\,2} + C_2^{(3)} - C_2^{(4)} + 3C_3^{(2)}h_3^{\,2} - C_3^{(3)} + C_3^{(4)} = 0}$$

S’’3(x3) = S’’2(x3)

$$\boxed{\left(\frac{h_3^{\,2}}{6} - \frac{h_2^{\,2}}{6}\right)z_3 + 6C_3^{(2)}h_3 - 6C_2^{(1)}h_2 = 0}$$

S'''3(x3) = S'''2(x3)

$$\boxed{\left(\frac{h_2}{2}+\frac{h_3}{2}\right)z_3+6C_2^{(1)}-6C_2^{(2)}-6C_3^{(1)}+6C_3^{(2)}=0}$$

S2(x3) = y3

$$\boxed{\frac{z_3}{120}h_2^{\ 4}+C_2^{(1)}h_2^{\ 3}+C_2^{(3)}h_2=y_3}$$

S3(x3) = y3

$$\boxed{\frac{z_3}{120}h_3^{\ 4}+C_3^{(2)}h_3^{\ 3}+C_3^{(4)}h_3=y_3}$$

S'4(x4) = S'3(x4)

$$\boxed{\left(\frac{h_3^{\ 3}}{24}+\frac{h_4^{\ 3}}{24}\right)z_4+3C_3^{(1)}h_3^{\ 2}+3C_4^{(2)}h_4^{\ 2}+C_3^{(3)}-C_3^{(4)}-C_4^{(3)}+C_4^{(4)}=0}$$

S''4(x4) = S''3(x4)

$$\boxed{\left(\frac{h_4^{\ 2}}{6}-\frac{h_3^{\ 2}}{6}\right)z_4+6C_4^{(2)}h_4-6C_3^{(1)}h_3=0}$$

S'''4(x4) = S'''3(x4)

$$\boxed{\left(\frac{h_3}{2}+\frac{h_4}{2}\right)z_4+6C_3^{(1)}-6C_3^{(2)}-6C_4^{(1)}+6C_4^{(2)}=0}$$

S3(x4) = y4

$$\boxed{\frac{z_3}{120}h_3^{\ 4}+C_3^{(2)}h_3^{\ 3}+C_3^{(4)}h_3=y_4}$$

S4(x4) = y4

$$\boxed{\frac{z_4}{120}h_4^{\ 4}+C_4^{(2)}h_4^{\ 3}+C_4^{(4)}h_4=y_4}$$

S'5(x5) = S'4(x5)

$$\boxed{\left(\frac{h_4^{\ 4}}{24}+\frac{h_5^{\ 3}}{24}\right)z_5+3C_4^{(1)}h_4^{\ 2}+3C_5^{(2)}h_5^{\ 2}+C_4^{(3)}-C_4^{(4)}-C_5^{(3)}+C_5^{(4)}=0}$$

S’’5(x5) = S’’4(x5)

$$\boxed{\left(\frac{h_5^{\ 2}}{6}-\frac{h_4^{\ 2}}{6}\right)z_5+6C_5^{(2)}h_5-6C_4^{(1)}h_4=0}$$

S’’’5(x5) = S’’’4(x5)

$$\boxed{\frac{z_5}{2}h_4+\frac{z_5}{2}h_5+6C_4^{(1)}-6C_4^{(2)}-6C_5^{(1)}+6C_5^{(2)}=0}$$

S4(x5) = y5

$$\boxed{\frac{z_5}{120}h_4^{\ 4}+C_4^{(1)}h_4^{\ 3}+C_4^{(3)}h_4=y_4}$$

Y según el orden establecido para la matriz antes expuesta y para las condiciones que se han expuesto después, podemos expresar estos nodos en nuestra matriz.

Para hacernos una primera idea de la forma que tendrá nuestra matriz hemos utilizado una hoja de excell, dado que la matriz es lo suficientemente grande como para no caber en un A4 a continuación se define por bloques la matriz.

Para las condiciones expuestas desde el nodo 2 al 3 hemos obtenido la siguiente matriz.

h2^4/120	0	h2^3	0	h2	0	0	0	0	0
0	3h2^2	0	1	-1	(h2^3/24) + (h3^3/24)	0	3*h3^2	-1	1
0	"-6h2	0	0	0	(h3^2/6) - (h2^2/6)	0	6*h3	0	0
0	6	"-6	0	0	(h2/2) + (h3/2)	"-6	6	0	0
0	h2^3	0	h2	0	h2^4/120	0	0	0	0

La matriz que sigue es la de las condiciones existentes entre los nodos 3 y 4

h3^4/120	0	h3^3	0	h3	0	0	0	0	0
0	3h3^2	0	1	-1	(h3^3/24) + (h4^3/24)	0	3*h4^2	-1	1
0	"-6h3	0	0	0	(h4^2/6) - (h3^2/6)	0	6*h4	0	0
0	6	"-6	0	0	(h3/2) + (h4/2)	"-6	6	0	0

0	h3^3	0	h3	0	h3^4/120	0	0	0	0

Y la siguiente la de los nodos 4 y 5

h4^4/120	0	h4^3	0	h4	0	0	0	0	0
0	3h4^2	0	1	-1	(h4^3/24) + (h5^3/24)	0	3*h5^2	-1	1
0	"-6h4	0	0	0	(h5^2/6) - (h4^2/6)	0	6*h5	0	0
0	6	-6	0	0	(h4/2) + (h5/2)	-6	6	0	0
0	h4^3	0	h4	0	h4^4/120	0	0	0	0

Podemos observar como este bloque sigue un patrón claro que se repite para cada polinomio. Podemos crear una matriz genérica que nos sirva de base para crear la matriz de cada pareja de nodos.

La matriz tendría esta forma:

hi4/120	0	hi^3	0	hi	0	0	0	0	0
0	3hi2	0	1	-1	(hi+13/24) + (hi3/24)	0	3*hi+12	-1	1
0	"-6hi	0	0	0	(hi+12/6) - (hi2/6)	0	6*hi+1	0	0
0	6	-6	0	0	(hi+1/2) + (hi/2)	-6	6	0	0
0	hi3	0	hi	0	hi4/120	0	0	0	0

Por tanto el último bloque será el del ultimo nodo interior y tendrá la siguiente forma.

hn-24/120	0	hn-2^3	0	hn-2	0	0	0	0	0
0	3hn-22	0	1	-1	(hn-23/24) + (hn-13/24)	0	3*hn-12	-1	1
0	"-6hn-2	0	0	0	(hn-22/6) - (hn-12/6)	0	6*hn-1	0	0
0	6	-6	0	0	(hn-2/2) + (hn-1/2)	-6	6	0	0
0	Hn-23	0	Hn-2	0	Hn-24/1	0	0	0	0

					20				

Las ecuaciones de este último bloque quedan:

Sn-2(xn-2)

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1}-x)^5 + \frac{z_{i+1}}{120h_i}(x-x_i)^5 + C_i^{(1)}(x-x_i)^3 + C_i^{(2)}(x_{i+1}-x)^3 + C_i^{(3)}(x-x_i) + C_i^{(4)}(x_{i+1}-x)$$

$$S_{n-2}(x) = \frac{z_{n-2}}{120h_{n-2}}(x_{n-1}-x)^5 + \frac{z_{n-1}}{120h_{n-2}}(x-x_{n-2})^5 + C_{n-2}^{(1)}(x-x_{n-2})^3 + C_{n-2}^{(2)}(x_{n-1}-x)^3 +$$
$$+ C_{n-2}^{(3)}(x-x_{n-2}) + C_{n-2}^{(4)}(x_{n-1}-x)$$

$$S_{n-2}(n-2) = \frac{z_{n-2}}{120h_{n-2}}(x_{n-1}-x_{n-2})^5 + \frac{z_{n-1}}{120h_{n-2}}(x_{n-2}-x_{n-2})^5 + C_{n-2}^{(1)}(x_{n-2}-x_{n-2})^3 + C_{n-2}^{(2)}(x_{n-1}-x_{n-2})^3 +$$
$$+ C_{n-2}^{(3)}(x_{n-2}-x_{n-2}) + C_{n-2}^{(4)}(x_{n-1}-x_{n-2})$$

$$S_{n-2}(n-2) = \frac{z_{n-2}}{120h_{n-2}}h_{n-2}{}^5 + C_{n-2}^{(3)}h_{n-2} + C_{n-2}^{(4)}h_{n-2}$$

$$\boxed{\frac{z_{n-2}}{120h_{n-2}}h_{n-2}{}^5 + C_{n-2}^{(3)}h_{n-2} + C_{n-2}^{(4)}h_{n-2} = y_{n-2}}$$

S'n-2(xn-1)=S'n-1(xn-1)

$$S_i'(x) = -\frac{z_i}{24h_i}(x_{i+1}-x)^4 + \frac{z_{i+1}}{24h_i}(x-x_i)^4 + 3C_i^{(1)}(x-x_i)^2 - 3C_i^{(2)}(x_{i+1}-x)^2 + C_i^{(3)} - C_i^{(4)}$$

$$S_{n-2}'(x) = -\frac{z_{n-2}}{24h_{n-2}}(x_{n-1}-x)^4 + \frac{z_{n-1}}{24h_{n-2}}(x-x_{n-2})^4 + 3C_{n-2}^{(1)}(x-x_{n-2})^2 - 3C_{n-2}^{(2)}(x_{n-1}-x)^2 + C_{n-2}^{(3)} - C_{n-2}^{(4)}$$

$$S_{n-2}'(x_{n-1}) = -\frac{z_{n-2}}{24h_{n-2}}(x_{n-1}-x_{n-1})^4 + \frac{z_{n-1}}{24h_{n-2}}(x_{n-1}-x_{n-2})^4 + 3C_{n-2}^{(1)}(x_{n-1}-x_{n-2})^2$$
$$-3C_{n-2}^{(2)}(x_{n-1}-x_{n-2})^2 + C_{n-2}^{(3)} - C_{n-2}^{(4)}$$

$$S_{n-2}'(x_{n-1})\frac{z_{n-1}}{24}h_{n-2}{}^3 + 3C_{n-2}^{(1)}h_{n-2}{}^2 + C_{n-2}^{(3)} - C_{n-2}^{(4)}$$

$$S_{n-1}'(x) = -\frac{z_{n-1}}{24h_{n-1}}(x_n-x)^4 + \frac{z_n}{24h_{n-1}}(x-x_{n-1})^4 + 3C_{n-1}^{(1)}(x-x_{n-1})^2 - 3C_{n-1}^{(2)}(x_n-x)^2 + C_{n-1}^{(3)} - C_{n-1}^{(4)}$$

$$S_{n-1}'(x_{n-1}) = -\frac{z_{n-1}}{24h_{n-1}}(x_n - x_{n-1})^4 + \frac{z_n}{24h_{n-1}}(x_{n-1} - x_{n-1})^4 + 3C_{n-1}^{(1)}(x_{n-1} - x_{n-1})^2$$
$$-3C_{n-1}^{(2)}(x_n - x_{n-1})^2 + C_{n-1}^{(3)} - C_{n-1}^{(4)}$$

$$S_{n-1}'(x_{n-1}) = -\frac{z_{n-1}}{24h_{n-1}}(x_n - x_{n-1})^4 + \frac{z_n}{24h_{n-1}}(x_{n-1} - x_{n-1})^4 + 3C_{n-1}^{(1)}(x_{n-1} - x_{n-1})^2$$
$$-3C_{n-1}^{(2)}(x_n - x_{n-1})^2 + C_{n-1}^{(3)} - C_{n-1}^{(4)}$$

$$-\frac{z_{n-1}}{24}h_{n-1}{}^3 - 3C_{n-1}^{(2)}h_{n-1}{}^2 + C_{n-1}^{(3)} - C_{n-1}^{(4)} = \frac{z_{n-1}}{24}h_{n-2}{}^3 + 3C_{n-2}^{(1)}h_{n-2}{}^2 + C_{n-2}^{(3)} - C_{n-2}^{(4)}$$

$$-\frac{z_{n-1}}{24}h_{n-1}{}^3 - 3C_{n-1}^{(2)}h_{n-1}{}^2 + C_{n-1}^{(3)} - C_{n-1}^{(4)} = \frac{z_{n-1}}{24}h_{n-2}{}^3 + 3C_{n-2}^{(1)}h_{n-2}{}^2 + C_{n-2}^{(3)} - C_{n-2}^{(4)}$$

$$\boxed{\left(\frac{h_{n-2}{}^3}{24} + \frac{h_{n-1}{}^3}{24}\right)z_{n-1} + 3C_{n-2}^{(1)}h_{n-2}{}^2 + 3C_{n-1}^{(2)}h_{n-1}{}^2 + C_{n-2}^{(3)} - C_{n-2}^{(4)} - C_{n-1}^{(3)} + C_{n-1}^{(4)} = 0}$$

S’’n-2(xn-1)=S’’n-1(xn-1)

$$S_i''(x) = \frac{z_i}{6h_i}(x_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - x_i)^3 + 6C_i^{(1)}(x - x_i) + 6C_i^{(2)}(x_{i+1} - x)$$

$$S_{n-2}''(x) = \frac{z_{n-2}}{6h_{n-2}}(x_{n-1} - x)^3 + \frac{z_{n-1}}{6h_{n-2}}(x - x_{n-2})^3 + 6C_{n-2}^{(1)}(x - x_{n-2}) + 6C_{n-2}^{(2)}(x_{n-1} - x)$$

$$S_{n-2}''(x_{n-1}) = \frac{z_{n-2}}{6h_{n-2}}(x_{n-1} - x_{n-1})^3 + \frac{z_{n-1}}{6h_{n-2}}(x_{n-1} - x_{n-2})^3 + 6C_{n-2}^{(1)}(x_{n-1} - x_{n-2}) + 6C_{n-2}^{(2)}(x_{n-1} - x_{n-1})$$

$$S_{n-2}''(x_{n-1}) = \frac{z_{n-1}}{6}h_{n-2}{}^2 + 6C_{n-2}^{(1)}h_{n-2}$$

$$S_{n-1}''(x) = \frac{z_{n-1}}{6h_{n-1}}(x_n - x)^3 + \frac{z_n}{6h_{n-1}}(x - x_{n-1})^3 + 6C_{n-1}^{(1)}(x - x_{n-1}) + 6C_{n-1}^{(2)}(x_n - x)$$

$$S_{n-1}''(x_{n-1}) = \frac{z_{n-1}}{6h_{n-1}}(x_n - x_{n-1})^3 + \frac{z_n}{6h_{n-1}}(x_{n-1} - x_{n-1})^3 + 6C_{n-1}^{(1)}(x_{n-1} - x_{n-1}) + 6C_{n-1}^{(2)}(x_n - x_{n-1})$$

$$S_{n-1}''(x_{n-1}) = \frac{z_{n-1}}{6}h_{n-1}{}^2 + 6C_{n-1}^{(2)}h_{n-1}$$

$$\boxed{\frac{z_{n-1}}{6}h_{n-1}{}^2 - \frac{z_{n-1}}{6}h_{n-2}{}^2 + 6C_{n-1}^{(2)}h_{n-1} - 6C_{n-2}^{(1)}h_{n-2} = 0}$$

S’’n-2(xn-1)=S’’n-1(xn-1)

$$S_i'''(x) = -\frac{z_i}{2h_i}(x_{i+1}-x)^2 + \frac{z_{i+1}}{2h_i}(x-x_i)^2 + 6C_i^{(1)} - 6C_i^{(2)}$$

$$S_{n-2}'''(x) = -\frac{z_{i-2}}{2h_{n-2}}(x_{n-1}-x)^2 + \frac{z_{n-1}}{2h_{n-2}}(x-x_{n-2})^2 + 6C_{n-2}^{(1)} - 6C_{n-2}^{(2)}$$

$$S_{n-2}'''(x_{n-1}) = -\frac{z_{i-2}}{2h_{n-2}}(x_{n-1}-x_{n-1})^2 + \frac{z_{n-1}}{2h_{n-2}}(x_{n-1}-x_{n-2})^2 + 6C_{n-2}^{(1)} - 6C_{n-2}^{(2)}$$

$$S_{n-2}'''(x_{n-1}) = +\frac{z_{n-1}}{2}h_{n-2} + 6C_{n-2}^{(1)} - 6C_{n-2}^{(2)}$$

$$S_{n-1}'''(x) = -\frac{z_{n-1}}{2h_{n-1}}(x_n-x)^2 + \frac{z_n}{2h_{n-1}}(x-x_{n-1})^2 + 6C_{n-1}^{(1)} - 6C_{n-1}^{(2)}$$

$$S_{n-1}'''(x_{n-1}) = -\frac{z_{n-1}}{2h_{n-1}}(x_n-x_{n-1})^2 + \frac{z_n}{2h_{n-1}}(x_{n-1}-x_{n-1})^2 + 6C_{n-1}^{(1)} - 6C_{n-1}^{(2)}$$

$$S_{n-1}'''(x_{n-1}) = -\frac{z_{n-1}}{2}h_{n-1} + 6C_{n-1}^{(1)} - 6C_{n-1}^{(2)}$$

$$\boxed{\frac{z_{n-1}}{2}h_{n-2} + 6C_{n-2}^{(1)} - 6C_{n-2}^{(2)} = -\frac{z_{n-1}}{2}h_{n-1} + 6C_{n-1}^{(1)} - 6C_{n-1}^{(2)}}$$

Además de las condiciones puestas en el extremo izquierdo, también debemos poner las condiciones en el extremo derecho, ya desarrolladas anteriormente.

A estas condiciones también se le suman las de Sn-1 en los extremos del polinomio, es decir Sn-1(xn-1) y Sn-1(xn). A continuación se desarrollan las ambas condiciones.

$$S_i(x) = \frac{z_i}{120h_i}(x_{i+1}-x)^5 + \frac{z_{i+1}}{120h_i}(x-x_i)^5 + C_i^{(1)}(x-x_i)^3 + C_i^{(2)}(x_{i+1}-x)^3 + C_i^{(3)}(x-x_i) + C_i^{(4)}(x_{i+1}-x)$$

$$S_{n-1}(x) = \frac{z_{n-1}}{120h_{n-1}}(x_n-x)^5 + \frac{z_n}{120h_{n-1}}(x-x_{n-1})^5 + C_{n-1}^{(1)}(x-x_{n-1})^3 + C_{n-1}^{(2)}(x_n-x)^3 + $$
$$+ C_{n-1}^{(3)}(x-x_{n-1}) + C_{n-1}^{(4)}(x_n-x)$$

Sn-1(xn-1)=yn-1

$$S_{n-1}(x_{n-1}) = \frac{z_{n-1}}{120h_{n-1}}(x_n-x_{n-1})^5 + \frac{z_n}{120h_{n-1}}(x_{n-1}-x_{n-1})^5 + C_{n-1}^{(1)}(x_{n-1}-x_{n-1})^3 + C_{n-1}^{(2)}(x_n-x_{n-1})^3 + $$
$$+ C_{n-1}^{(3)}(x_{n-1}-x_{n-1}) + C_{n-1}^{(4)}(x_n-x_{n-1})$$

$$\boxed{\frac{z_{n-1}}{120} h_{n-1}{}^{4} + C_{n-1}^{(2)} h_{n-1}{}^{3} + C_{n-1}^{(4)} h_{n-1} = y_{n-1}}$$

Sn-1(xn)=yn

$$S_{n-1}(x) = \frac{z_{n-1}}{120h_{n-1}}(x_n - x_n)^5 + \frac{z_n}{120h_{n-1}}(x_n - x_{n-1})^5 + C_{n-1}^{(1)}(x_n - x_{n-1})^3 + C_{n-1}^{(2)}(x_n - x_n)^3 +$$
$$+ C_{n-1}^{(3)}(x_n - x_{n-1}) + C_{n-1}^{(4)}(x_n - x_n)$$

$$\boxed{\frac{z_n}{120} h_{n-1}{}^{4} + C_{n-1}^{(1)} h_{n-1}{}^{3} + C_{n-1}^{(3)} h_{n-1} = y_n}$$

Llamando a cada bloque anteriormente expuesto según el nodo al que pertenece Ai,i+1 y las condiciones del extremo izquierdo CI y CD.

CI		0	0	0	0	...	0	0	0	0
A1,2		0	0	0	0	...	0	0	0	0
0	A2,3		0	0	0	...	0	0	0	0
0	0	A3,4		0	0	...	0	0	0	0
0	0	0	A4,5		0	...	0	0	0	0
0	0	0	0	A5,6		...	0	0	0	0
...	...	...	...	...	...	...	...	0	0	0
0	0	0	0	0	...	Ai,i+1		0	0	0
0	0	0	0	0	0	...	...	...	0	0
0	0	0	0	0	0	0	0	An-2,n-1		0
0	0	0	0	0	0	0	0	0	Sn-1(xn-1)	
0	0	0	0	0	0	0	0	0	Sn-1(xn)	
0	0	0	0	0	0	0	0	0	CD	

12. ESTUDIO DE LOS RESULTADOS

COMPARACIÓN Y OBTENCIÓN DE ERRORES

En este apartado vamos a proceder a comparar los resultados obtenidos al utilizar un spline de quinto grado y el spline de tercer grado. Para ello vamos a utilizar el software desarrollado para este trabajo y el software desarrollado para un trabajo fin de carrera defendido por Irene Gallego Valedellos cuyo titulo es “Splines cúbicos interpolantes en el diseño naval”.

Vamos a empezar haciendo un mayado pequeño, y por pasos lo vamos a ir refinando, es decir haciendo cada vez las h más pequeñas. Así podremos observar como se va ajustando el Spline a la función.

La función elegida es:

Sen(x)*x2 [0, pi]

Empezaremos tomando 3 puntos de control, después 5, seguidamente tomaremos 9 , 17, 33 y por último 65.

Con esos puntos de control crearemos los diferentes splines y compararemos para cada mayado, los resultados de los splines con los resultados obtenido en la función original. Para evaluar el error tenemos que hacer un espaciado (mev) en el eje de abscisas para nuestro spline para cada intervalo. Este espaciado debe ser muy superior al número de nodos escogido.

mev >> n.

Por ello vamos a evaluar los splines y la función en 200 puntos por cada tramo.

Las condiciones extra se realizan automáticamente al introducir la función f. Las condiciones extra que hemos tomado para resolver nuestro spline son los valores en los extremos de las cuartas y las segunda derivadas.

El error será la diferencia máxima en valor absoluto que existe entre los valores de las ordenadas obtenidos de la evaluación del spline y de la función original.

Xev = Valor en x de los puntos obtenidos al tomar un espaciado de las dimensiones de mev.

Fevaluacíón = Valores en y que toma nuestro spline al evaluar con los putos de Xev.

Foriginal = Valores en y que toma la función original al evaluar con los putos de Xev.

Hemos comparado los valores de F evaluación y F original y obtenido el error máximo.

$$E_i = \text{Max } |\text{Fevaluación}(xi) - \text{Foriginal}(xi)|$$

En código quedaría:

```
Error=max(transpose(max(abs(F evaluación))))
```

Una vez aprendida la teoría pasamos a la evaluación de los dos tipos de Spline.

EVALUACIÓN SPLINE DE QUINTO GRADO

Creamos un mallado nuevo E1 para el Spline de quinto grado y evaluamos:

Nuestro primer mallado tiene 3 nodos.

Calculamos el error cometido.

E_1 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.415599287407751

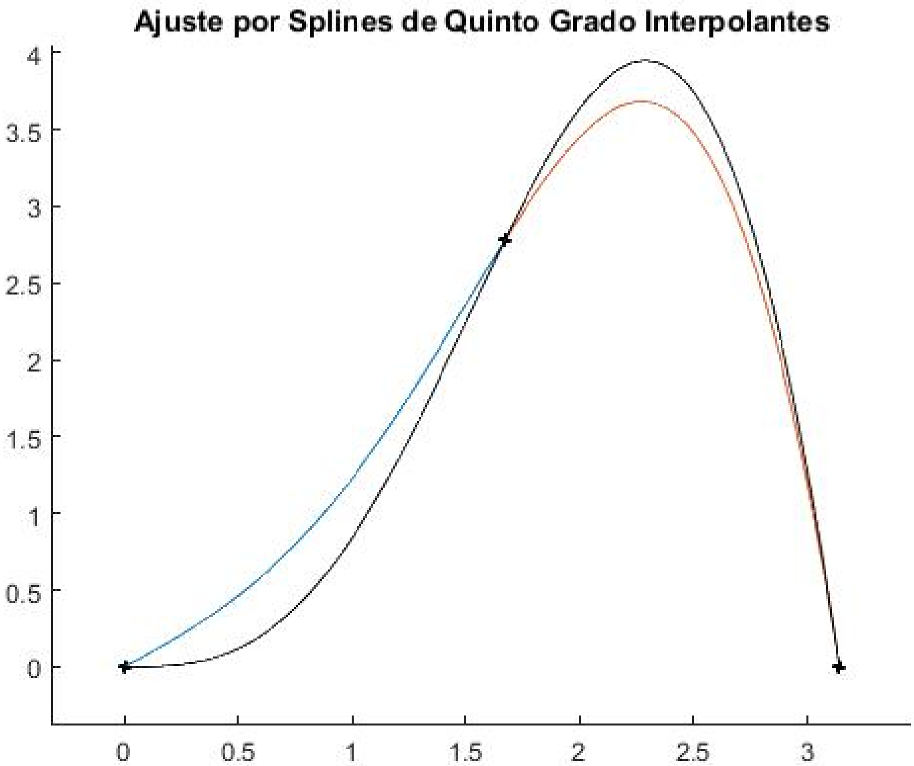


Imagen 12.1 Representación del spline de quinto grado con tres nodos y la función original

En negro se representa la función original, y en colores el Spline con sus diferentes tramos. Observamos como el spline se aproxima en cierto modo a la función original, pero todavía necesita más puntos de control para llegar a parecerse a la función original. Refinando el mallado obtendremos errores menores.

A continuación se muestra la representación de las cuatro derivadas del spline de quinto grado para 3 nodos:

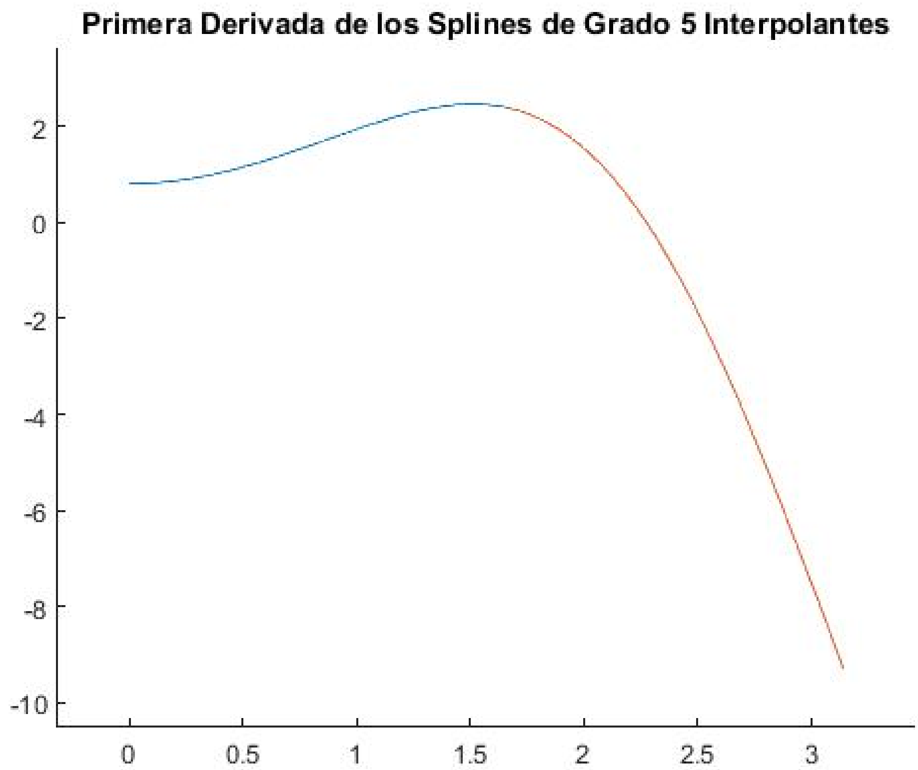


Imagen 12.2 Representación de la primera derivada del spline de quinto grado con tres nodos.

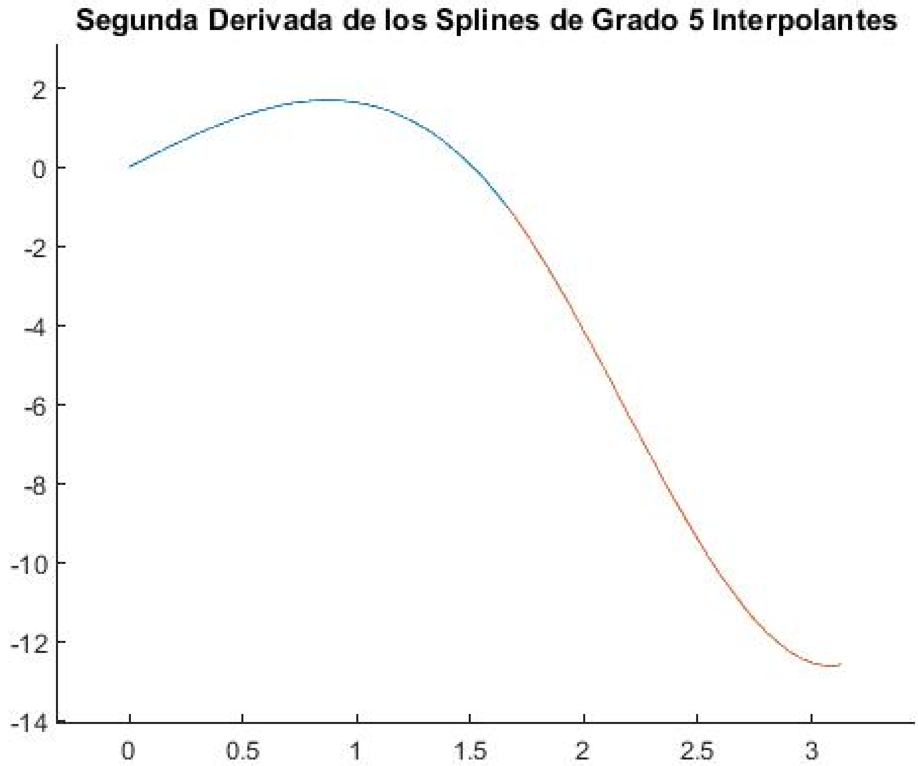


Imagen 12.3 Representación de la segunda derivada del spline de quinto grado con tres nodos.

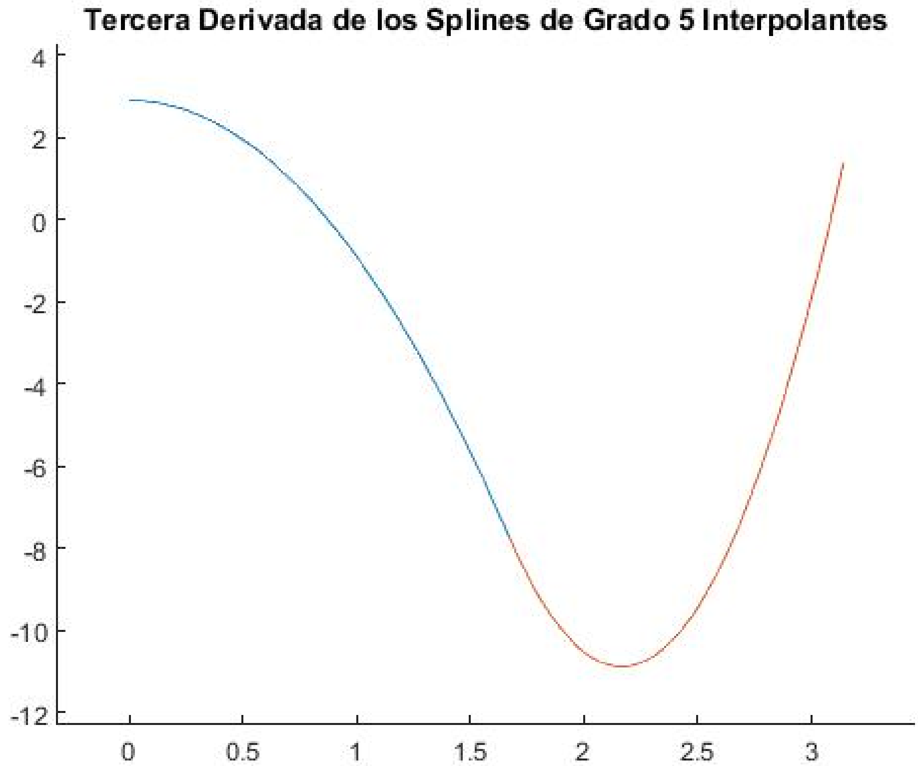


Imagen 12.4 Representación de la tercera derivada del spline de quinto grado con tres nodos.

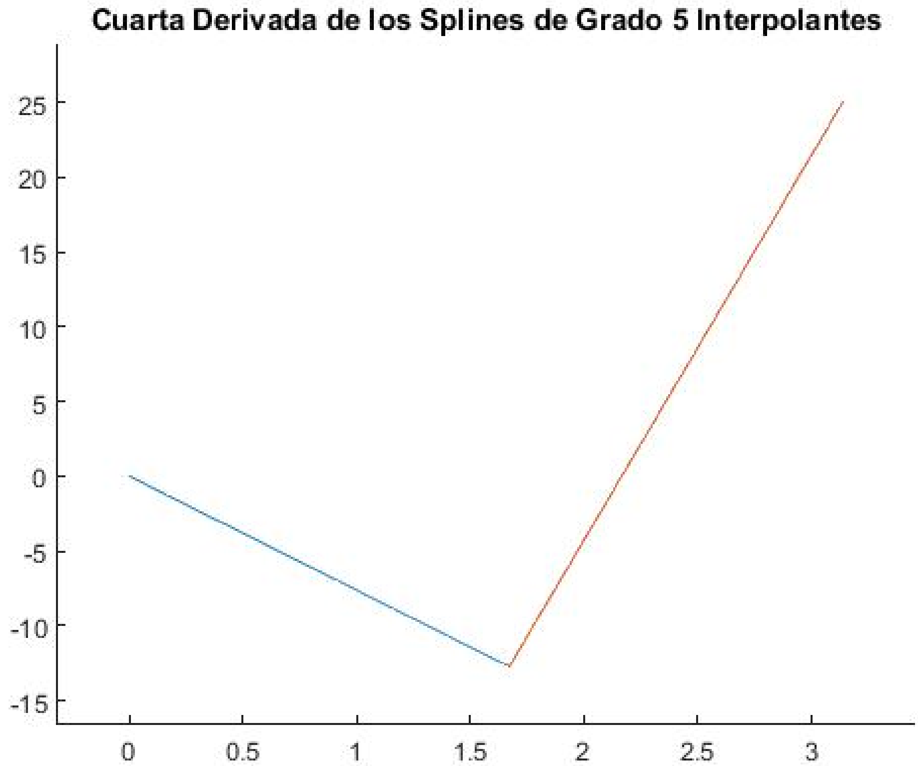


Imagen 12.5 Representación de la cuarta derivada del spline de quinto grado con tres nodos.

En relación a las representación de las cuatro derivadas del spline de quinto grado en 3 nodos podemos observar como las 4 gráficas son continuas y por tanto: La continuidad en la primera derivada nos garantiza la ausencia de picos en el Spline, la segunda la ausencia de cambios bruscos de curvatura en el spline, y la continuidad en la tercera y cuarta son importantes a nivel geométrico.

Es importante comprobar que la cuarta derivada no solo es continua sino que une los puntos a través de funciones lineales.

Refinamos el mallado y evaluamos:

Ponemos 5 puntos por cada tramo

E_2 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.002316045112701

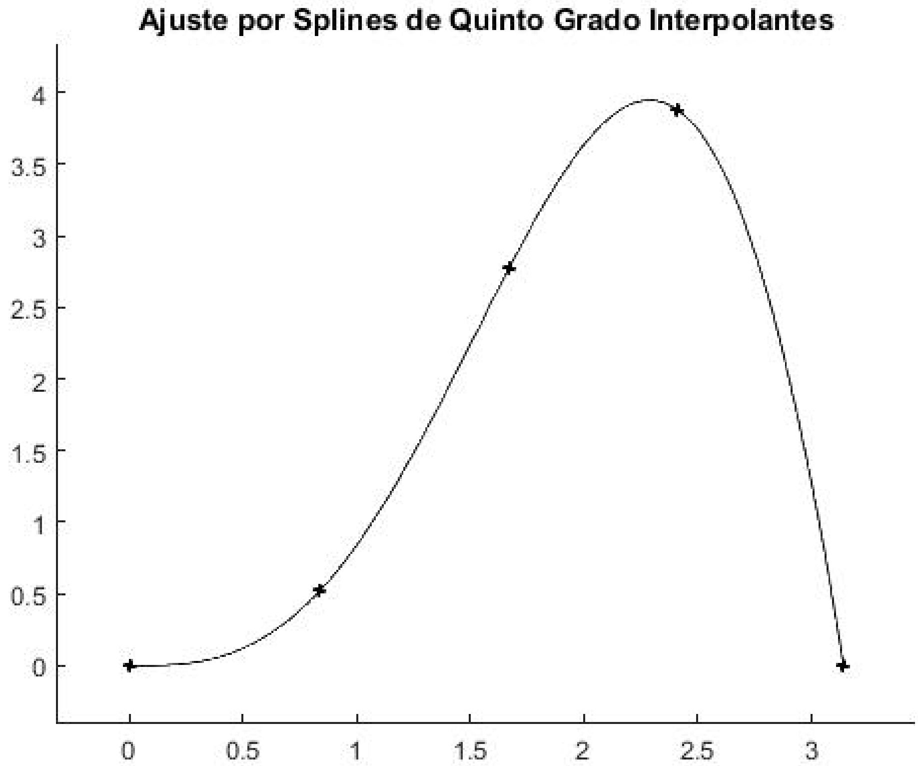


Imagen 12.6 Representación del spline de quinto grado con cinco nodos y la función original

En esta gráfica del spline de quinto grado con 5 puntos de control podemos observar como se ajusta casi a la perfección a la función original

Exponemos las cuatro derivadas a continuación:

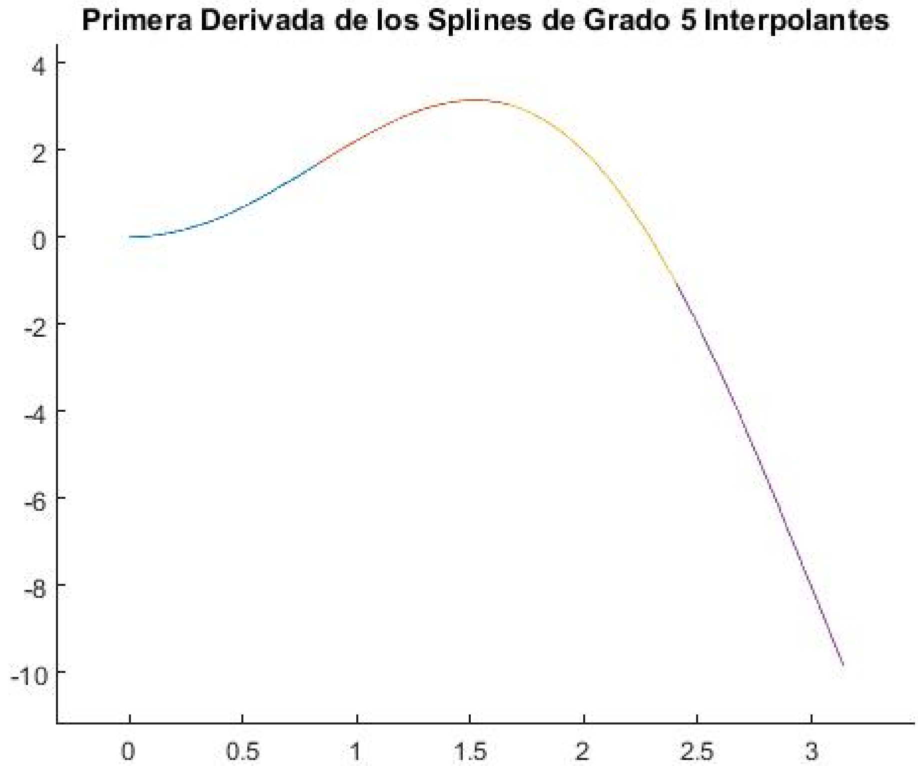


Imagen 12.7 Representación de la primera derivada del spline de quinto grado con cinco nodos.

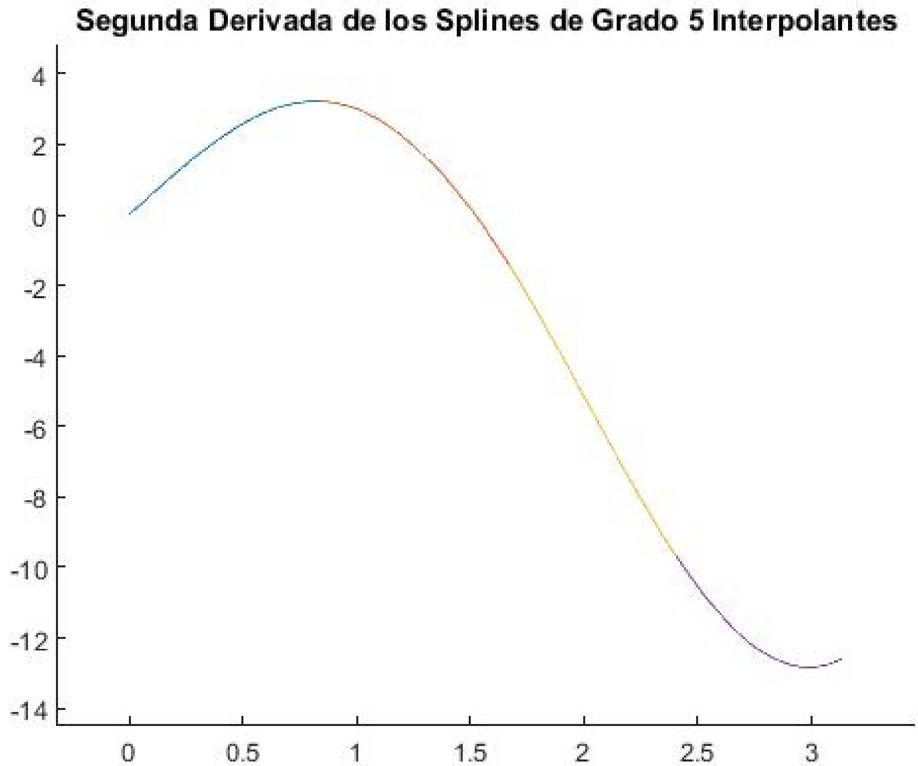


Imagen 12.8 Representación de la segunda derivada del spline de quinto grado con cinco nodos.

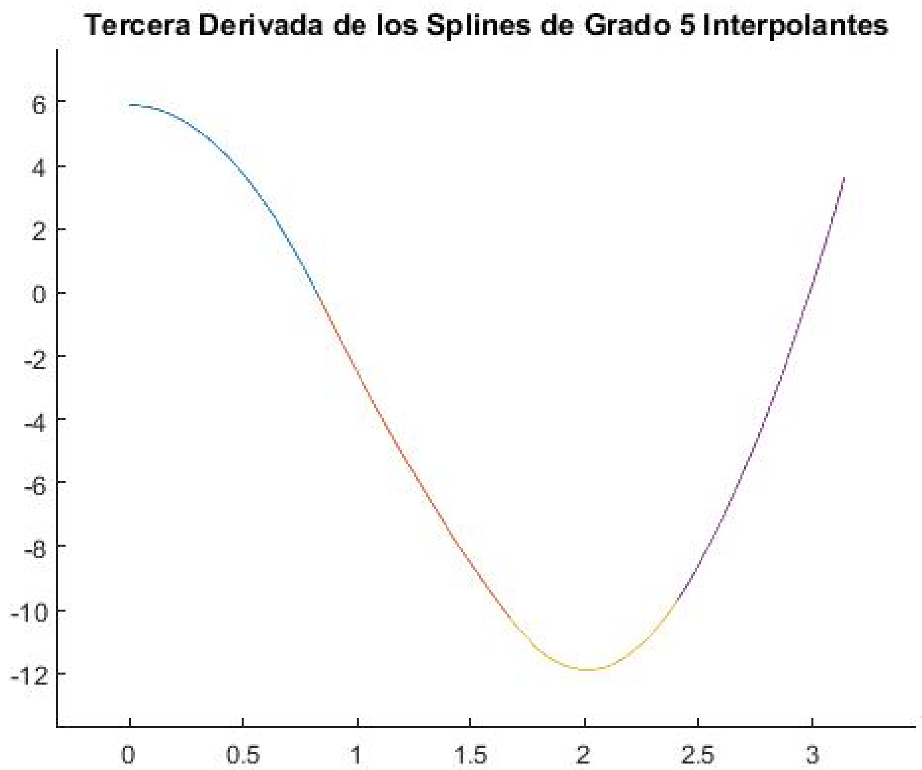


Imagen 12.9 Representación de la tercera derivada del spline de quinto grado con cinco nodos.

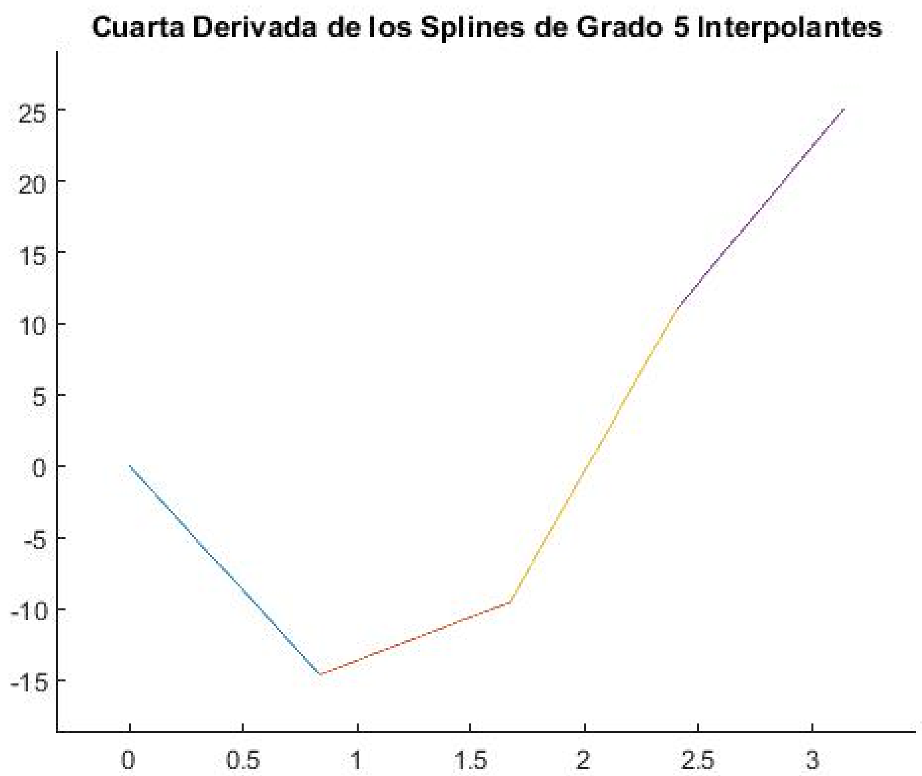


Imagen 12.10 Representación de la cuarta derivada del spline de quinto grado con cinco nodos.

De la representación de estas cuatro derivadas, podemos observar de nuevo que se cumple la continuidad.. Como dijimos anteriormente, la continuidad de la tercera y cuarta derivada presenta mejoras de carácter geométrico, estas mejoras se pueden observar directamente en lo bien que se ajusta el spline de 5 grado con pocos puntos de control, es decir con "h" no muy reducido. Esto presenta mejoras a nivel de precisión pero necesita de más horas para la computación que un spline cúbico.

Seguimos refinando el spline de quinto grado para ver como va disminuyendo el error.

Refinamos el mallado anterior y evaluamos:

Ponemos 9 puntos por cada tramo.

E_3 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000038416664892

Lo refinamos una vez más y evaluamos:

Ponemos 17 puntos por cada tramo.

E_4 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000000608936024

Lo refinamos una vez más y evaluamos:

Ponemos 33 puntos por cada tramo

E_5 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000000009551059

Hacemos un último refinado y evaluamos:

Ponemos 65 puntos por cada tramo

E_6 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000000000149623

Podemos observar como el error es cada vez menor, y el spline se va ajustando cada vez más a la función original.

EVALUACIÓN SPLINE CÚBICO

Hacemos el mismo proceso para un spline de grado tres:

Creamos un mallado nuevo E1 para el Spline cúbico y evaluamos:

Nuestro primer mallado tiene 3 nodos.

E_1 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.950121886418655

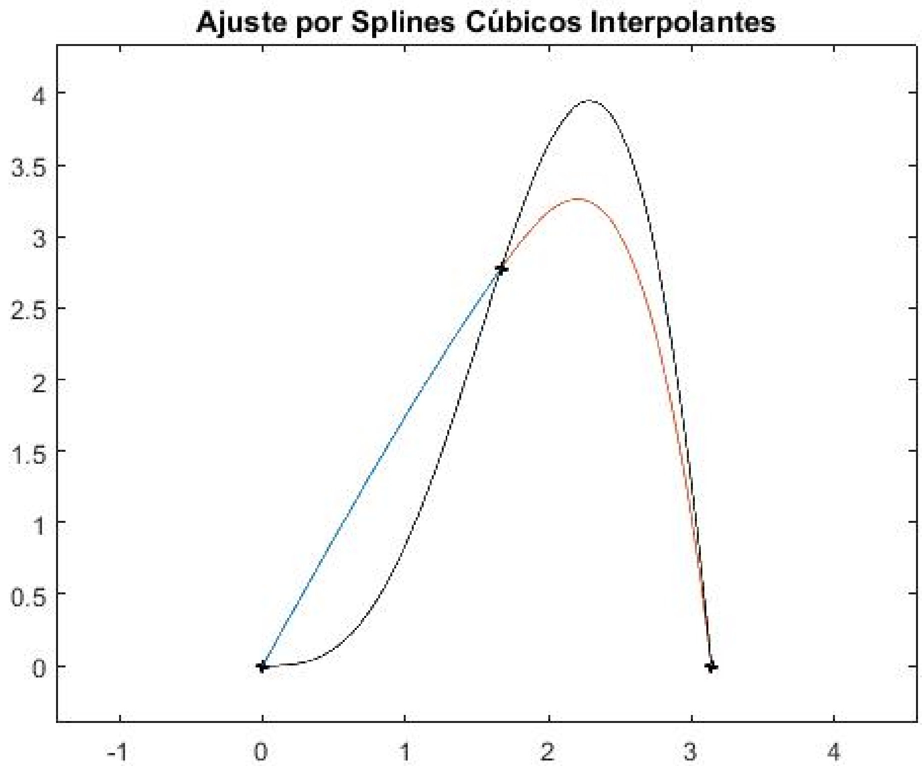


Imagen 12.11 Representación del Spline cúbico con tres nodos y la función original

Podemos observar como con solo tres puntos el Spline no se ajusta bien a la función original. Por tanto será necesario tomar más puntos de control para definir bien la función.

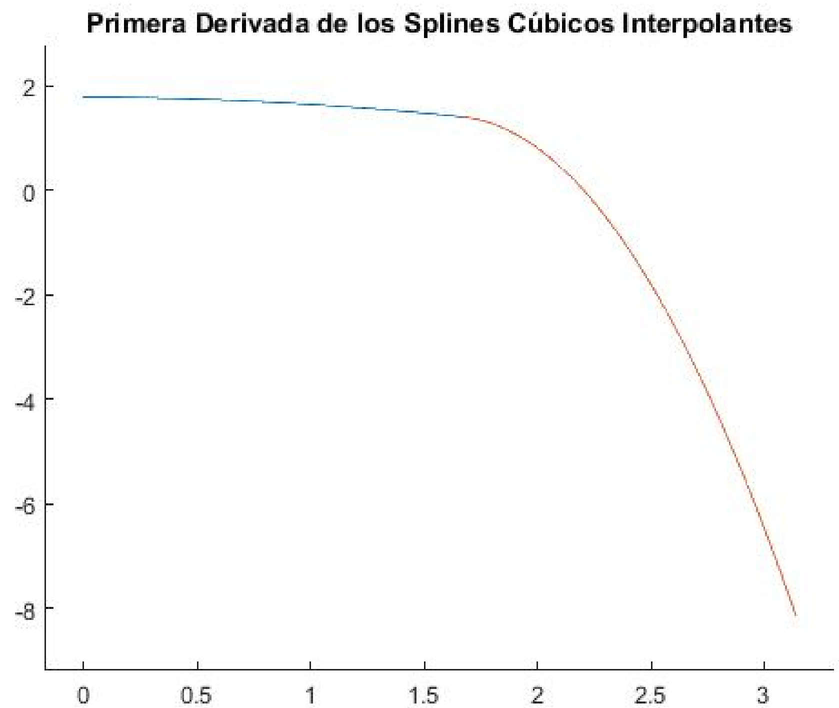


Imagen 12.12 Representación de la primera derivada del spline cúbico en con tres nodos.

La primera derivada de nuestro spline cubico creado con 3 puntos de control nos muestra la continuidad de esta. La continuidad en la segunda derivada es una de las condiciones impuestas.

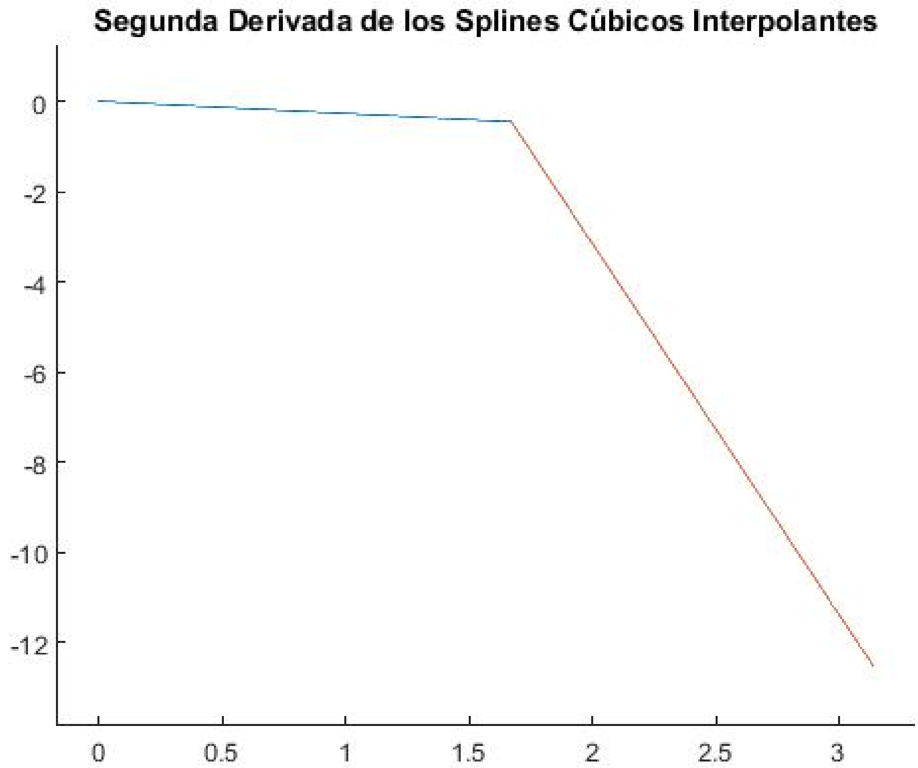


Imagen 12.13 Representación de la segunda derivada del spline cúbico en con tres nodos.

La segunda derivada del spline cúbico cumple la continuidad, importante para no tener cambios bruscos en el spline. Cumple la función de Lagrange por tanto une con rectas los puntos z_i..

Podemos observar como la gráfica lo cumple.

Refinamos el mallado y evaluamos:

Ponemos 5 puntos por cada tramo

E_2 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.045450200446050

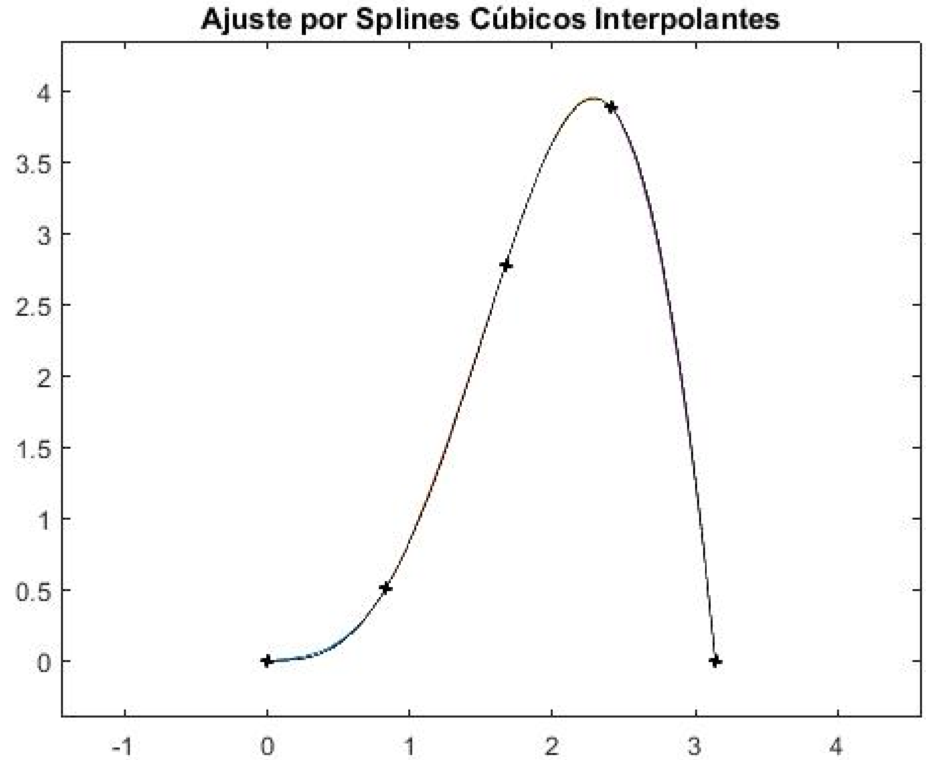


Imagen 12.14 Representación del spline cúbico con 5 nodos y la función original.

Como podemos observar con cinco puntos el spline cúbico representa muy bien la función, es por ello que se solapa y apenas se ven colores. Sin embargo también se puede ver como en el primer tramo se ajusta peor que el spline de quinto grado a la función original, lo cual demuestra visualmente lo que por los valores de los errores hemos observado matemáticamente.

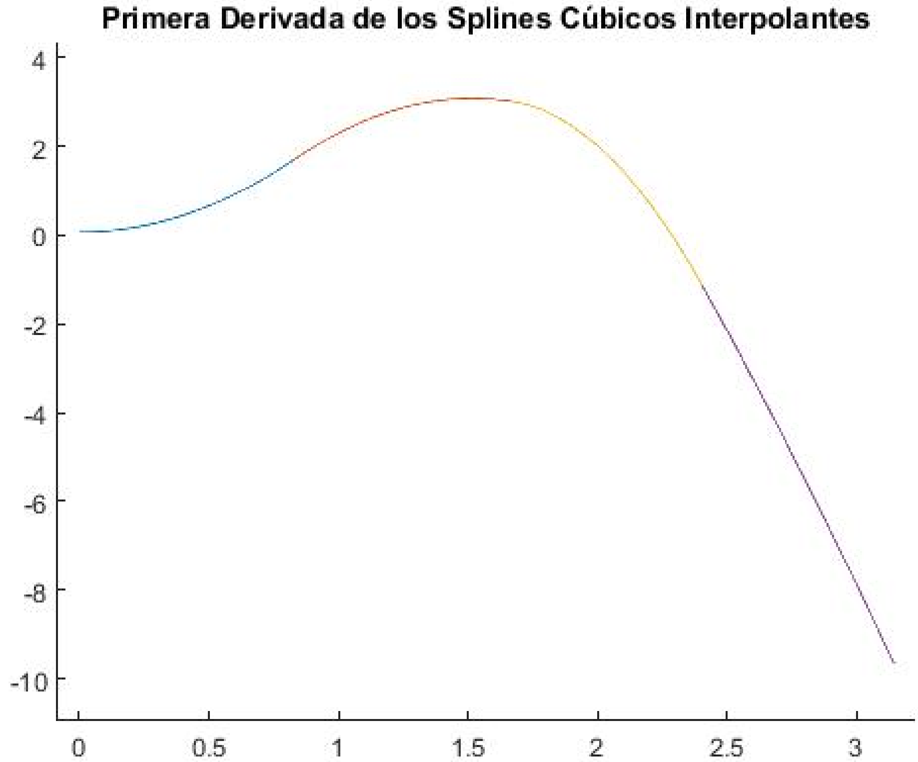


Imagen 12.15 Representación de la primera derivada del spline cúbico con cinco nodos.

Podemos observar una vez más la continuidad exigida en la primera derivada. La cual nos garantiza la ausencia de picos en el spline.

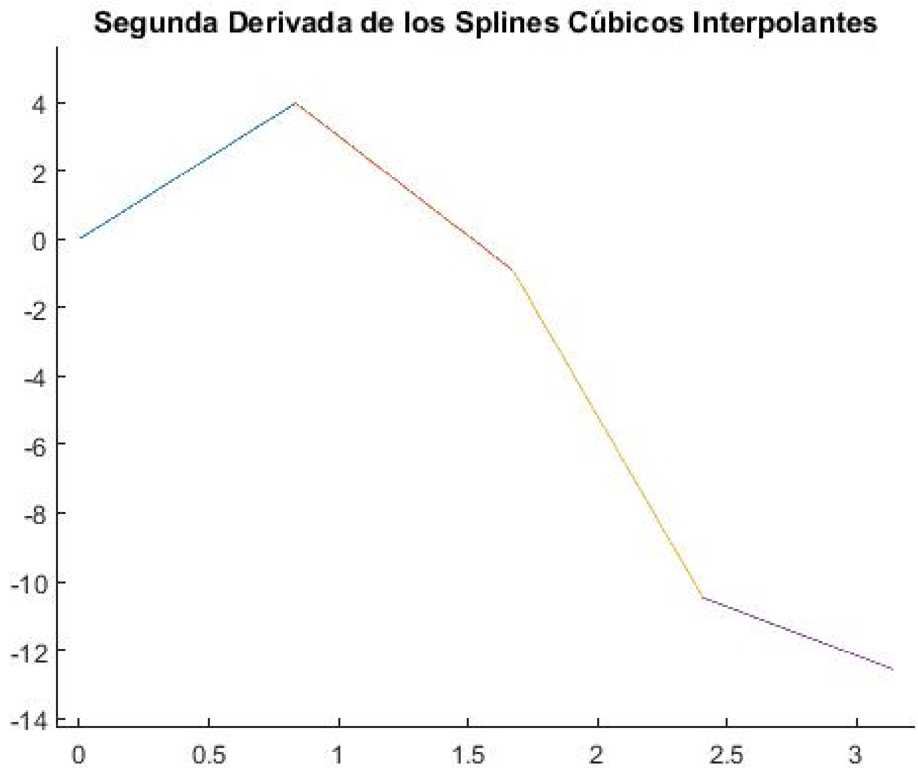


Imagen 12.16 Representación de la segunda derivada del spline cúbico con cinco nodos.

En esta última imagen se puede ver como los cuatro tramos son funciones lineares en la segunda derivada cumpliendo como se dijo en la teoría con la función de Lagaraje a partir de la cual se han desarrollado ambos splines. La continuidad de la segunda derivada nos garantiza que no existen cambios bruscos en el spline.

Lo refinamos una vez más y evaluamos:

Ponemos 9 puntos por cada tramo

E_3 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.002937838031998

Lo refinamos una vez más y evaluamos:

Ponemos 17 puntos por cada tramo

E_4 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000185803885455

Lo refinamos una vez más y evaluamos:

Ponemos 33 puntos por cada tramo

E_5 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000011996085232

Hacemos un último refinado y evaluamos:

Ponemos 65 puntos por cada tramo

E_6 = Max |Fevaluación(xi) - Foriginal(xi)| = 0.000000731011641

Con el spline de grado cinco podemos obtener un error pequeño sin necesidad de hacer un mayado tan fino como el que necesitaríamos para obtener un error pequeño en el spline cúbico.

OBTENCIÓN DEL ORDEN DEL SPLINE

Sabemos que el error que toma nuestro spline cumplirá la siguiente función

$$Eh_i \approx Ch^p$$

Siendo:

- Eh = El error obtenido en cada mallado.
- C = Es una constante.
- h = Será el número de nodos en el que se evalúa.
- p = Orden de aproximación.

Si generamos un mallado uniforme h y vamos realizando un mallado cada vez más fino, como por ejemplo:

$$n1 = 3 \quad n2 = 5 \quad n3 = 9 \quad n4 = 17 \quad n5 = 33 \quad n6 = 65$$

Cada uno de esos mallados tendrá un h diferente que será la mitad que el anterior. Por tanto:

$$\begin{array}{lllll} n_1 & \rightarrow & h_1 & & \\ n_2 & \rightarrow & h_1 / 2 & \rightarrow & h_2 \\ n_3 & \rightarrow & h_2 / 2 & \rightarrow & h_3 \\ n_4 & \rightarrow & h_3 / 2 & \rightarrow & h_4 \\ n_5 & \rightarrow & h_4 / 2 & \rightarrow & h_5 \\ n_6 & \rightarrow & h_5 / 2 & \rightarrow & h_6 \end{array}$$

Sabemos que para un mallado uniforme:

$$E_{hi} \approx C{*}h_i^p$$

Siendo :

- E_{hi} : El error que contiene el mallado i.
- C: Una constante.
- h_i: El valor de cada tramo.
- p: El orden.

Si hacemos una sucesión de errores podemos observar hacia que números tiende el orden (p)

$$\begin{aligned} Eh_1 &= C * h_1^p = 0.415599287407751 \\ Eh_2 &= C * h_2^p = 0.002316045112701 \\ Eh_3 &= C * h_3^p = 0.000038416664892 \\ Eh_4 &= C * h_4^p = 0.000000608936024 \\ Eh_5 &= C * h_5^p = 0.000000009551059 \\ Eh_6 &= C * h_6^p = 0.000000000149623 \end{aligned}$$

Podemos hacer las siguientes operaciones para despejar p y ver hacia que número tiende:

$$\frac{Eh_1}{Eh_2} = \frac{C*h_1{}^p}{C*h_2{}^p} = \frac{h_1{}^p}{\left(\frac{h_1}{2}\right)^p} = 2^p \quad \rightarrow \quad p_1 = \log_2(Eh_1/Eh_2) = 7.48738601456996$$

$$\frac{Eh_2}{Eh_3} = \frac{C*h_2{}^p}{C*h_3{}^p} = \frac{h_2{}^p}{\left(\frac{h_2}{2}\right)^p} = 2^p \quad \rightarrow \quad p_2 = \log_2(Eh_2/Eh_3) = 5.913787266491999$$

$$\frac{Eh_3}{Eh_4} = \frac{C*h_3{}^p}{C*h_4{}^p} = \frac{h_3{}^p}{\left(\frac{h_3}{2}\right)^p} = 2^p \quad \rightarrow \quad p_3 = \log_2(Eh_3/Eh_4) = 5.979297804347880$$

$$\frac{Eh_4}{Eh_5}=\frac{C*h_4{}^p}{C*h_5{}^p}=\frac{h_4{}^p}{\left(\frac{h_4}{2}\right)^p}=2^p \quad \rightarrow \quad p_3=\log_2(Eh_4/Eh_5) = 5.994486148687814$$

$$\frac{Eh_5}{Eh_6}=\frac{C*h_5{}^p}{C*h_6{}^p}=\frac{h_5{}^p}{\left(\frac{h_5}{2}\right)^p}=2^p \quad \rightarrow \quad p_3=\log_2(Eh_5/Eh_6) = 5.996253276307561$$

Como podemos observar nuestro spline de quinto grado tiende un orden p igual a seis.

Si hiciésemos lo mismo pero para un spline cúbico, obtendríamos un orden p igual a cuatro. Lo demostramos a continuación:

Hacemos la sucesión de errores para el spline cúbico con los diferentes mayados hallados anteriormente.

$$Eh_1 = C * h_1{}^p = 0.950121886418655$$
$$Eh_2 = C * h_2{}^p = 0.045450200446050$$
$$Eh_3 = C * h_3{}^p = 0.002937838031998$$
$$Eh_4 = C * h_4{}^p = 0.000185803885455$$
$$Eh_5 = C * h_5{}^p = 0.000011996085232$$
$$Eh_6 = C * h_6{}^p = 0.000000731011641$$

Volvemos a hacer las operaciones para despejar p y ver hacia que número tiende:

$$\frac{Eh_1}{Eh_2}=\frac{C*h_1{}^p}{C*h_2{}^p}=\frac{h_1{}^p}{\left(\frac{h_1}{2}\right)^p}=2^p \quad \rightarrow \quad p_1=\log_2(Eh_1/Eh_2) = 4.385754039328704$$

$$\frac{Eh_2}{Eh_3}=\frac{C*h_2{}^p}{C*h_3{}^p}=\frac{h_2{}^p}{\left(\frac{h_2}{2}\right)^p}=2^p \quad \rightarrow \quad p_2=\log_2(Eh_2/Eh_3) = 3.951459892086313$$

$$\frac{Eh_3}{Eh_4}=\frac{C*h_3{}^p}{C*h_4{}^p}=\frac{h_3{}^p}{\left(\frac{h_3}{2}\right)^p}=2^p \quad \rightarrow \quad p_3=\log_2(Eh_3/Eh_4) = 3.982902283660007$$

$$\frac{Eh_4}{Eh_5}=\frac{C*h_4{}^p}{C*h_5{}^p}=\frac{h_4{}^p}{\left(\frac{h_4}{2}\right)^p}=2^p \quad \rightarrow \quad p_3=\log_2(Eh_4/Eh_5) = 3.953145088310987$$

$$\frac{Eh_5}{Eh_6} = \frac{C*h_5^{\,p}}{C*h_6^{\,p}} = \frac{h_5^{\,p}}{\left(\frac{h_5}{2}\right)^p} = 2^p \quad \rightarrow \quad p_3 = \log_2(Eh_5/Eh_6) = 4.000255065793533$$

Efectivamente, como se puede observar, el orden p del spline cúbico tiende a 4.

13. CASOS PRÁCTICO

Como hemos estudiado, los Splines tienen multitud de aplicaciones en numerosos campos. Para completar este estudio, ya que se trata de un proyecto de final de grado de arquitectura naval e ingeniería de sistemas marinos, hemos buscado unos ejemplos relacionados con ambos campos

CASO 1

Para el caso práctico primero, hemos obtenido las coordenadas de una cuaderna, en relación a la parte de arquitectura naval. A partir de estas coordenadas vamos a reconstruir la forma que tendría la cuaderna. Obtendremos además las derivadas de la función.

Nuestros datos han sido obtenidos de una caja de cuadernas. Se nos ha dado una nube de puntos. La primera columna nos indica la numeración que recibe esa cuaderna, la segunda la posición a lo largo de la eslora del buque y la tercera y cuarta columna serían los valores de Y y Z que debemos representar. En nuestros datos hay hasta 14 cuadernas. A continuación se expone una muestra de algunos de los datos.

1	0,000	4,700	6,561
1	0,000	5,250	8,748
1	0,000	5,378	9,408
2	6,218	0,000	3,505
2	6,218	2,020	3,645
2	6,218	4,210	4,374
2	6,218	5,280	6,561
2	6,218	5,960	8,748
2	6,218	6,102	9,249
3	12,436	0,000	2,832
3	12,436	1,020	2,916
3	12,436	3,990	3,645
3	12,436	4,780	4,374
3	12,436	5,778	6,561
3	12,436	6,530	8,748
3	12,436	6,642	9,115
4	24,872	0,000	1,483
4	24,872	2,670	2,187
4	24,872	4,460	2,916
4	24,872	5,340	3,645
4	24,872	5,770	4,374
4	24,872	6,500	6,561
4	24,872	7,090	8,748
4	24,872	7,134	8,928
		...	

Siendo el eje Y el eje de la manga y Z el eje vertical del buque, vamos a utilizar la cuaderna 11 que tiene las siguientes coordenadas:

Y: [0.000, 0.880, 1.190, 1.410, 1.590, 1.760, 1.950, 2.710, 3.700, 4.902]
Z: [0.000, 0.729, 1.458, 2.187, 2.916, 3.645, 4.374, 6.561, 8.748, 11.207]

Como no tenemos información sobre las cuartas derivadas ni la segundas hemos tomado los valores naturales de estas.

A continuación nos disponemos a insertar los valores de las ordenadas y abscisas en el software que hemos desarrollado, para ello ha sido necesario hacer una pequeña modificación que nos facilite insertar los datos de manera directa.

Una vez insertados los valores, nuestro software construye el spline de quinto grado según el número de puntos de control y sus valores.

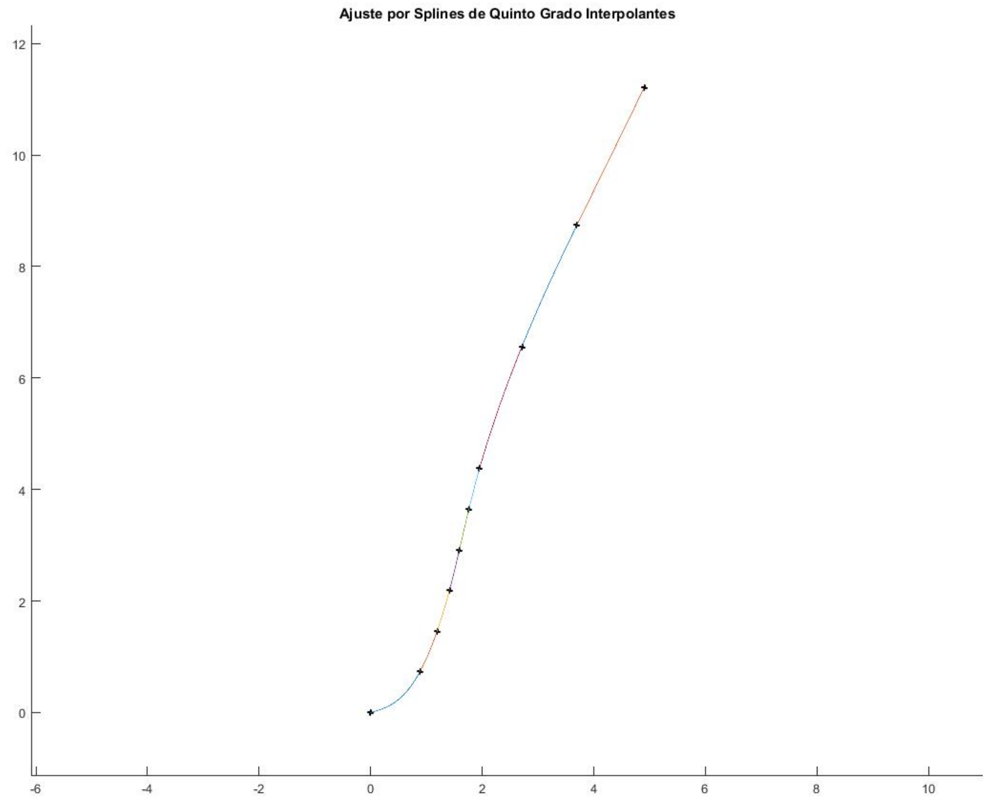


Imagen 13.1. Cuaderna número 11 representado por un spline de quinto grado

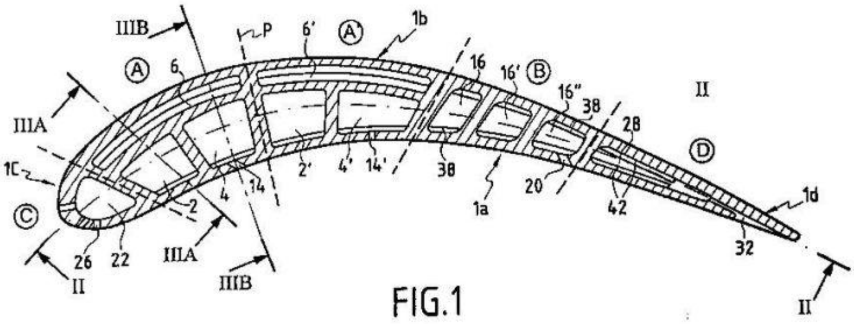
Podemos observar como el spline recrea de forma muy suave la curvatura de la cuaderna. Además podemos concluir que en un poco espacio de tiempo hemos podido reconstruir la forma de una cuaderna del casco, y obtenido las funciones que definen cualquier punto de esta. Hemos comprobado que cumple la continuidad en todos las derivadas.

El resultado es muy bueno, ya que los puntos han sido obtenidos a través de programas de dibujo muy precisos y los puntos de control pertenecen a una superficie ya alisada.

CASO 2

Para el caso segundo, referente a la parte de ingeniería naval, hemos elegido la representación de las curvas de una sección de un alabe de una turbina. La imagen que se muestra a continuación pertenece al circuito de refrigeración interno de un alabe.

A partir de este plano vamos a sacar diferentes punto de control, crear un spline de quinto grado a partir de estos y observar como se cumple con las formas del alabe. Para la obtención de los puntos de control, debemos tener en cuenta que hay que tomar mas puntos de control en las zonas donde más se curva el alabe.



Los puntos de control tomados son los siguientes:

X: [0, 36, 65, 116, 172, 222, 267, 336, 380, 452, 504, 558, 605, 637]
Y: [119, 175, 197, 222, 236, 240, 239, 226, 210, 184, 159, 135, 108, 87]

Ejecutando la gráfica del spline de quinto grado nos queda:

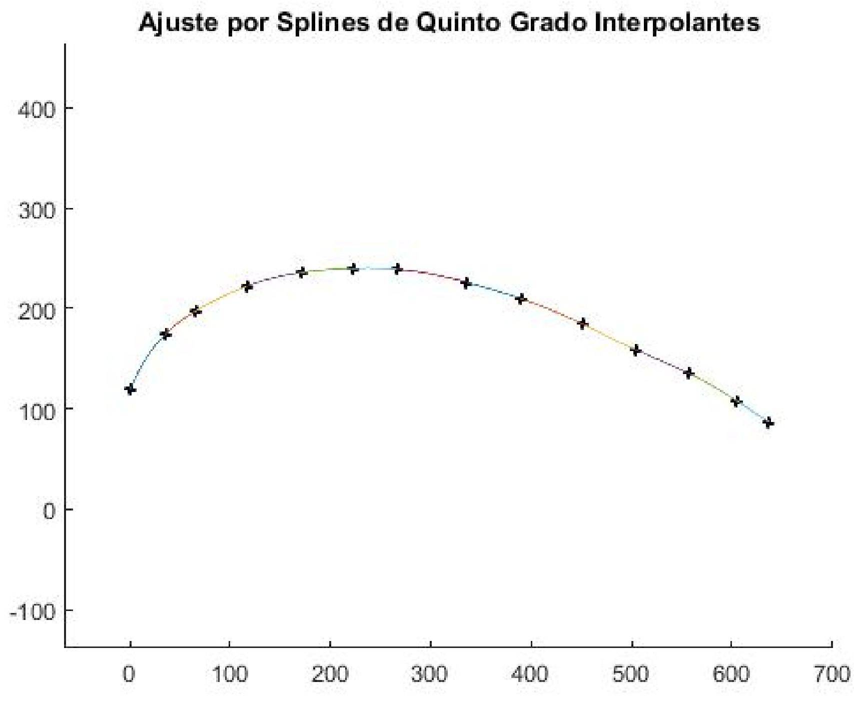


Imagen 11.2. Linea superior de la sección del alabe representado por un spline de quinto grado.

Cuando hay cambios en la convexidad hay que hacer un ajuste muy preciso en las condiciones de contorno para que el spline se ajuste a la función original. La obtención de datos se ha realizado a partir de la imagen y por tanto los puntos de control presentan una distorsión que complica la ejecución del spline de quinto grado. Además para un ajuste optimo se deben estudiar bien las condiciones de contorno de la curva. Para encontrar unas buenas condiciones de contorno, en un caso como este, se requiere mucho tiempo de estudio y es por ello por lo que no lo hemos visto viable.

Para curvas que no están alisadas o donde no se conocen las condiciones de contorno es mejor aproximarla a través de splines suavizantes.

14. INTERFAZ GRÁFICA

Vamos ha describir a continuación un breve tutorial por pasos sobre nuestra interfaz gráfica. Nuestra interfaz gráfica ha sido realizada a partir del entorno gráfico de Matlab Guide. Nuestra intención es acercar, a través de una interfaz intuitiva, los splines de quinto grado a cualquier usuario.

TUTORIAL DE EJECUCIÓN LA INTERFAZ.

Paso 1: Ejecutar el programa Matlab.

Paso 2: Seleccionar como "Current Folder" la carpeta que contenga todos los archivos de nuestro programa.

Paso 3: Ejecutamos el programa a través de la ventana de comandos escribiendo "splines5"

DESCRIPCIÓN DE LA INTERFAZ.

A continuación se muestra una captura de la interfaz que contiene los diferentes paneles:

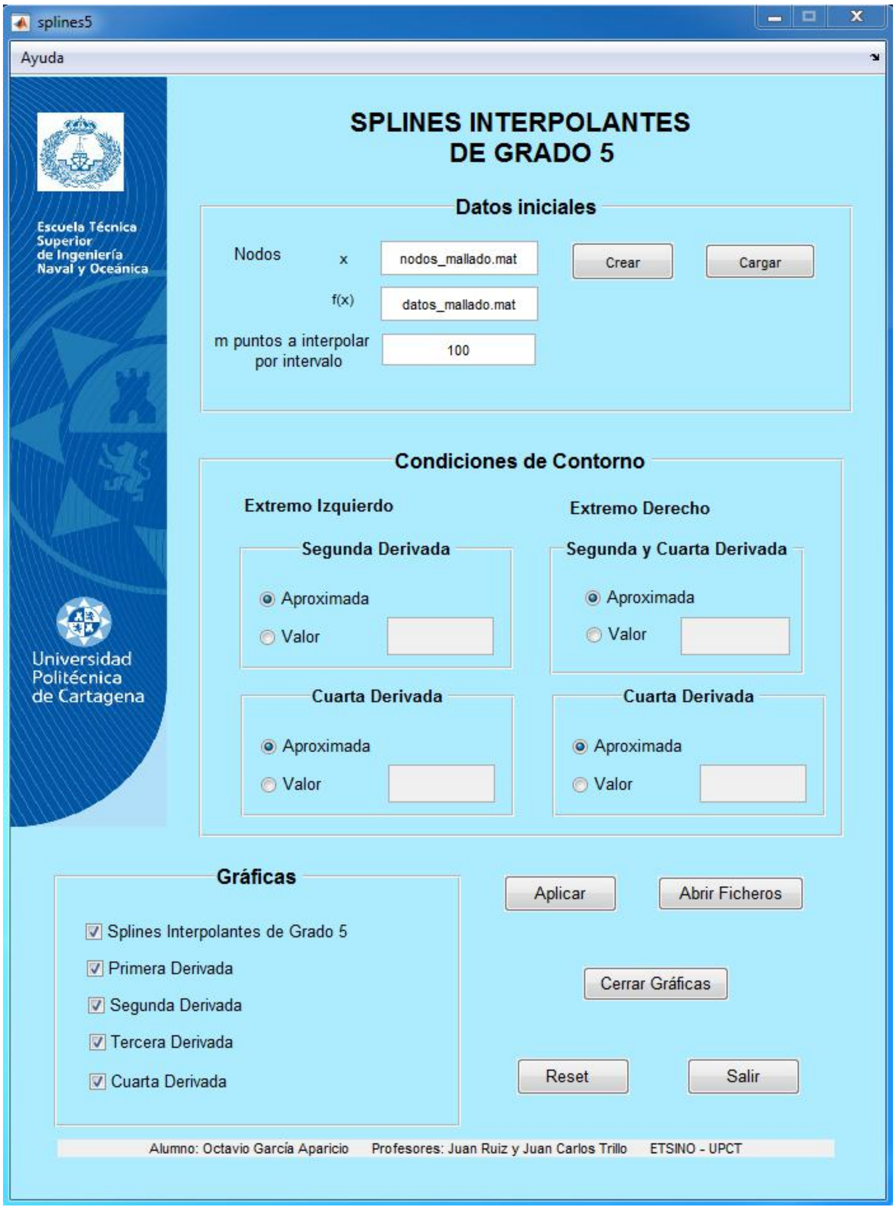


Imagen 14.1 Interfaz gráfica

Como podemos observar, nuestra interfaz esta compuesta por diferentes paneles que se explicarán en los siguientes apartados. Estos paneles son: “Datos Iniciales”, “Condiciones de Contorno” y “Gráficas”.

Ademas en la esquina de arriba disponemos de un botón de ayuda que nos será muy útil en el caso de no aclararnos con el programa.

BOTON DE AYUDA.

Cuando pulsamos en el se nos despliega un menú con tres posibilidades:

- **Acerca de:** Nos da un breve explicación sobre la utilidad del programa y sus creadores.

- Tutorial : Nos abre en un pdf que contiene este tutorial.

- TFG : Nos abre en un pdf el Trabajo fin de grado al que pertenece.

CAPTURA AYUDA Y ACERCA DE

PANEL DATOS INICIALES.

Imagen 14.2. Panel "Datos iniciales" de la interfaz gráfica.

En este panel existen tres espacios en blanco:

- **x** : Este espacio es para insertar las coordenadas en x que definirán los puntos de control de nuestro Spline. Podemos crear nuestras coordenadas pulsando en el botón "crear", en ese caso se nos abrirá un archivo en matlab que nos permitirá editar los datos de nuestras coordenadas. El botón "cargar" nos permite cargar los datos contenidos en un archivo ya existente. Debemos escribir el nombre del archivo en el recuadro blanco. Fichero: nodos_mallado.mat .

- **f (x):** Es para introducir los datos de las abscisas, podemos introducirlos a través de un fichero o crearlos. Fichero: datos_mallado.mat .

- **m puntos a interpolar por intervalo** : Para elegir cuantos puntos por tramos queremos que sea la función evaluada.

PANEL CONDICIONES DE CONTORNO.

Imagen 14.3. Panel "Condiciones de Contorno" de la interfaz gráfica.

Este panel sirve para elegir las condiciones de contorno. En el podemos ver cuatro cuadros:

- Extremo Izquierdo Segunda Derivada
- Extremo Derecho Segunda Derivada
- Extremo Izquierdo Cuarta Derivada
- Extremo Derecho Cuarta Derivada

Como se aprecia en estos cuadros podemos elegir si queremos que el programa aproxime las condiciones de contorno de manera automática o por el contrario preferimos nosotros tomar unos valores.

Para cada condición solo podemos pulsar uno de los botones que aparecen. Podemos pulsar o "Aproximada" o "Valor". Cuando en un mismo recuadro pulsamos el botón "Aproximación", automáticamente se desactiva el botón "Valor" Ocurriendo también al contrario. Esto evita problemas en el programa.

Cuando no tenemos posibilidad de meter unas condiciones de contorno fiables, podemos utilizar los valores naturales, es decir 0. Esto puede darnos buenos resultados a no ser que el valor de las derivadas no sea próximo a 0. En ese caso la precisión de los Splines de quinto grado se verá afectada.

Siempre que se conozcan los valores exactos se aconseja introducirlos, sobre todo cuando no se dispone de la función original que queremos aproximar.

PANEL GRÁFICAS.

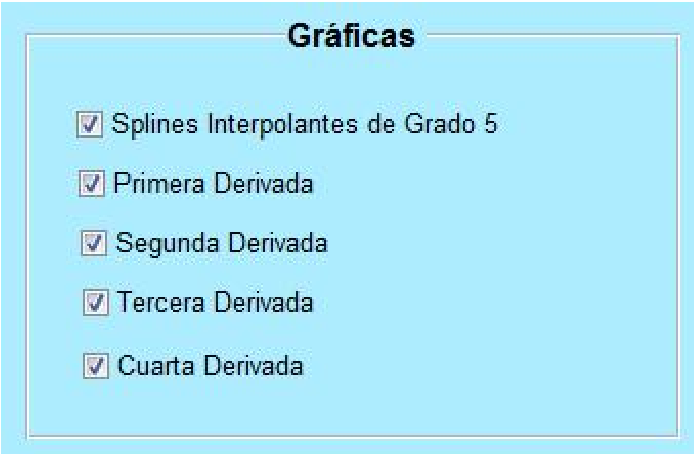


Imagen 14.4. Panel "Gráficas" de la interfaz gráfica.

En este panel aparecen diferentes opciones que podemos elegir:

- Splines Interpolantes de Grado 5
- Primera Derivada.
- Segunda Derivada.
- Tercera Derivada.
- Cuarta Derivada.

Podemos seleccionar las casillas que queramos, el programa nos mostrara las gráficas que marcadas, es decir la representación de nuestro spline y/o sus derivadas.

BOTONES DE ACCIÓN

Estos botones están junto al "Panel Gráficas". Estos son los que realizan diferentes acciones en el programa.

Imagen 14.5. Botones de acción de la interfaz gráfica.

- Aplicar: Este programa toma los datos introducidos en la interfaz gráfica y llama al programa “splinecincodatos.m”. Este programa se encarga de la resolución del problema y de mostrar las gráfica que se han pedido a través de la interfaz.

Cuando le damos a aplicar el programa guarda los datos en ficheros. Los ficheros guardados incluyen las expresiones de polinomios y la evaluación de los polinomios en las abscisas ordenadas en columnas.

- **Abrir Ficheros:** Este botón hace que se abran en Matlab los ficheros guardados en la anterior ejecución.

- Reset: Vuelve al estado inicial de la interfaz, borrando todas las opciones que haya podido escribir o seleccionar el usuario. Lo que no borra son las variables escritas en Matlab.

- Cerrar Gráficas: Cierra las gráficas que haya abierto el programa. Para volver a abrirlas se debe volver a pulsar “Aplicar”.

- Cerrar: Cierra la interfaz, aunque por seguridad el programa preguntara si realmente desea cerrarlo.

ANEXO

BLOQUE.M

```
function [c]=bloque(a)

i=sym(sprintf('h%d', [a]));
j=sym(sprintf('h%d', [a+1]));

c=[i^4/120,0,i^3,0,i,0,0,0,0,0                                      ;
0,3*i^2,0,1,-1,(i^3+j^3)/24,0,3*j^2,-1,1;...
    0,-6*i,0,0,0,(j^2-i^2)/6,0,6*j,0,0;
0,6,-6,0,0,(i+j)/2,-6,6,0,0;...
    0,i^3,0,i,0,(i^4/120),0,0,0,0];


End
```

MALLADO_NU.M

```
function xa=mallado_nu(a,b,n,lambda,nescalas)

% Esta función genera un mallado no uniforme por variación de lambda %
% del mallado uniforme en [a,b] con n puntos dejando fijos a y b
%
% xa=mallado_nu(a,b,n,lambda,nescalas);
%
% Variables de entrada:
% a,b extremos del intervalo
% n número de puntos en la discretización
% lambda tanto por uno de variación respecto del mallado uniforme
% nescalas número de escalas totales por subdivisión de los intervalos
%   por la mitad
%
% Variables de salida:
% xa mallado no uniforme generado en [a,b]
% Se graban tantos ficheros como número de escalas con el nombre
% mallado_nu_n50_Escalas4.mat si el número de puntos del mallado es 50
y
% se trata de la escala 4 de subdivisión.
%
% Ejemplo:
% xa=mallado_nu(0,2*pi,50,0.1,4);

% mallado uniforme inicial

xa=linspace(a,b,n);

% espaciado uniforme

hxa=(b-a)/n;

% variación introducida al mallado uniforme en los nodos centrales
% exceptuando a y b
```

```
rxa=lambda*hxa*rand(1,n-2);

% mallado no uniforme generado

xa(2:n-1)=xa(2:n-1)+rxa;


% generamos los mallados por subdivisión si la variable escalas es mayor
% que 1


str_file=['mallado_nu_n',num2str(n),'_Escalas1.mat'];
save(str_file,'xa');

if nescalas>2
    for i=2:nescalas

        % generamos el nuevo mallado

        n_nuevo=2*n-1;
        xa_nuevo=zeros(1,n_nuevo);
        xa_nuevo(1:2:n_nuevo)=xa(1:n);
        xa_nuevo(2:2:n_nuevo-1)=(xa(1:n-1)+xa(2:n))/2;

        % actualizamos el mallado

        xa=xa_nuevo;
        n=n_nuevo;

        % grabamos el nuevo mallado generado


str_file=['mallado_nu_n',num2str(n),'_Escalas',num2str(i),'.mat'];
        save(str_file,'xa');
    end
end
```

MATRIZ.M

```
function [M,X,Y]=Matriz(n)


clc


% Declarar las variables simbólicas

i=0;
while i<n
    i=i+1;
    h(i)=sym(sprintf('h%d', [i]));
end

M=sym((zeros((n-2)*5)+6));
```

```
% Generar primer bloque
a=1;                  %Este contador servirá para generar tantos bloques
como polinomios.
M(3:7,1:10)=bloque(a);


% Generar segundo bloque
b=0;                %Este contador es para la pósicion de los próximos
bloques en la matriz M

while a<n-2
    a=a+1;
    b=b+5;

    if a==2           %Esta condicion es para que el segundo bloque lo
coloque en una posicion correcta.
        b=0;
    end

    M(8+b:12+b,6+b:15+b)=bloque(a);
end




% Insertar condiciones izquierda.
M(1,1:5)=[h(1)^2/2,0,6*h(1),0,0];     %S1'' en x1
M(2,1:5)=[1,0,0,0,0];     %S1'''' en x1

% rInsertar condiciones derecha.

M(b+13,b+6:b+16)=[h(n-1)^4/120,0,h(n-1)^3,0,h(n-1),0];
M(b+14,b+6:b+16)=[0,h(n-1)^3,0,h(n-1),0,h(n-1)^4/120];
M(b+15,b+6:b+16)=[0,6*h(n-1),0,0,0,h(n-1)^2/6];
M(b+16,b+6:b+16)=[0,0,0,0,0,1];


%Generar vector de de incognitas

i=0;
j=-4;

while i<n-1
    i=i+1;
    j=j+5;
    X(1,j)=sym(sprintf('Z%d', [i]));
    X(1,j+1)=sym(sprintf('C%d%d', [i 1]));
    X(1,j+2)=sym(sprintf('C%d%d', [i 2]));
    X(1,j+3)=sym(sprintf('C%d%d', [i 3]));
    X(1,j+4)=sym(sprintf('C%d%d', [i 4]));
end
X(1,j+5)=sym(sprintf('Z%d', [i+1]));

i=0;
j=-2;
while i<n-2
```

```
    i=i+1;
    j=j+5;
    Y(1,j)=sym(sprintf('y%d', [i]));
    Y(1,j+1)=0;
    Y(1,j+2)=0;
    Y(1,j+3)=0;
    Y(1,j+4)=sym(sprintf('y%d', [i+1]));
end

Y(1,1)=sym(sprintf('d2a'));
Y(1,2)=sym(sprintf('d4a'));
Y(1,j+5)=sym(sprintf('y%d', [i+1]));
Y(1,j+6)=sym(sprintf('y%d', [i+2]));
Y(1,j+7)=sym(sprintf('d2b'));
Y(1,j+8)=sym(sprintf('d4b'));

X=transpose(X); Y=transpose(Y);
```

SPLINECINCO.M

```
function [S,Error]=SplineCinco(mallado,f,m)

% Este programa calcula los splines de grado 5 interpolatorios a partir
% de los datos iniciales del mallado y de los valores de una cierta función
% conocida en los puntos del mallado
%
% [S,Error]=SplineCinco(mallado,f,m)
%
% Variables de entrada:
% mallado   nombre del fichero que contiene las abscisas xa de los datos
%          usados para construir el spline
% f    función de la que obtenemos las ordenadas y de los datos
%          usados para construir el spline y aproximar dicha función
% m  número de puntos en que se divide cada subintervalo de manera uniforme
%
% Variables de salida:
% S vector simbólico que contiene las expresiones de los diferentes
% Error variable que mide la norma infinito entre la función evaluada
%      en un mallado fino y el spline de grado 5 obtenido evaluado en
%      el mismo mallado
%
% Ejemplo:
% syms x
%
[S,Error]=SplineCinco('mallado_nu_n10_Escalas1.mat',x^2*sin(x),200);


% Declaramos la variable x como simbólica

syms x
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           DATOS DE ENTRADA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



% Leemos las abscisas x

load(mallado);

% longitud del mallado

n=length(xa);

% Extremos del intervalo

a=xa(1);
b=xa(n);

% Valores de la función en las abscisas

y=double(subs(f,x,xa));


% Generamos los valores exactos de la función en un mallado más fino
% determinado por el valor de m

i=1;
while i<=n-1

    vx(i,1:m)=linspace(xa(i),xa(i+1),m);
    vyo(i,1:m)=double(subs(f,x,vx(i,1:m)));

    i=i+1;
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
%
% CÁLCULO DE LA DERIVADA SEGUNDA Y LA DERIVADA CUARTA EN LOS EXTREMOS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%



% Derivada segunda simbólica

df2=diff(f,x,2);

% Evaluación en a y b
```

```
d2a=double(subs(df2,x,a));
d2b=double(subs(df2,x,b));

% Derivada cuarta simbólica

df4=diff(f,x,4);

% Evaluación en a y b

d4a=double(subs(df4,x,a));
d4b=double(subs(df4,x,b));


% Crear los valores de hi en un vector

hi=diff(xa);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%        CREACIÓN DEL SISTEMA DE ECUACIONES LINEALES
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Inicializamos la matriz de coeficientes M

M=zeros((5*(n-2))+6,5*n-4);


% Crear el bloque numérico
i=0;
t=-2;
g=-4;

% Insertar bloques
while i<n-2
    i=i+1;
    c=[hi(i)^4/120,0,hi(i)^3,0,hi(i),0,0,0,0,0 ; ...

0,3*hi(i)^2,0,1,-1,(hi(i)^3+hi(i+1)^3)/24,0,3*hi(i+1)^2,-1,1;...
       0,-6*hi(i),0,0,0,(hi(i+1)^2-hi(i)^2)/6,0,6*hi(i+1),0,0;...
       0,6,-6,0,0,(hi(i)+hi(i+1))/2,-6,6,0,0;...
       0,hi(i)^3,0,hi(i),0,(hi(i)^4/120),0,0,0,0];
    t=t+5;   %fila
    g=g+5;   %columna
    M(t:t+4,g:g+9)=c;
end


% Insertar condiciones izquierda.
M(1,1:5)=[hi(1)^2/2,0,6*hi(1),0,0];      %S1'' en x1
M(2,1:5)=[1,0,0,0,0];      %S1'''' en x1

% Insertar condiciones derecha.
t=t+5;   %fila
g=g+5;   %columna
```

```
M(t,g:g+5)=[hi(n-1)^4/120,0,hi(n-1)^3,0,hi(n-1),0];
M(t+1,g:g+5)=[0,hi(n-1)^3,0,hi(n-1),0,hi(n-1)^4/120];
M(t+2,g:g+5)=[0,6*hi(n-1),0,0,0,hi(n-1)^2/6];
M(t+3,g:g+5)=[0,0,0,0,0,1];

% Vector de terminos independientes
V=transpose(zeros(size(1:(5*(n-2))+6)));

i=3;
j=1;
while i<5*(n-2)+6
    V(i,1)=y(1,j);
    V(i-1,1)=y(1,j);
    i=i+5;
    j=j+1;
end

V(1,1)=d2a;
V(2,1)=d4a;

V(5*(n-2)+4,1)=y(n);
V(5*(n-2)+5,1)=d2b;
V(5*(n-2)+6,1)=d4b;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   RESOLVEMOS EL SISTEMA DE ECUACIONES LINEALES
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hi
M
V

X=linsolve(M,V);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  CREAMOS CADA UNO DE LOS POLINOMIOS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Crear Polinomios y coordenadas para representar

syms S

i=1;
j=1;
while i<=n-1
    S(i,1)=((X(j)/(120*hi(i)))*(xa(i+1)-x)^5)...
        +((X(j+5)/(120*hi(i)))*(x-xa(i))^5)...
        +(X(j+1)*(x-xa(i))^3)...
        +(X(j+2)*(xa(i+1)-x)^3)...
        +(X(j+3)*(x-xa(i)))...
        +(X(j+4)*(xa(i+1)-x));
```

```
    vx(i,1:m)=linspace(xa(i),xa(i+1),m);
    vy(i,1:m)=subs(S(i),x,vx(i,1:m));

    i=i+1;
    j=j+5;
end

vy=double(vy);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%    REPRESENTAMOS GRÁFICAMENTE EL RESULTADO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



figure('NumberTitle','off','Name','Splines de Quinto Grado')
hold on

% Dibujamos los puntos de control

plot(xa,y,'ko','MarkerSize',2,'LineWidth',3);


% Dibujamos también la función original y los splines

Msd=max(max(vy));
msd=min(min(vy));

i=1;
while i<n
    plot(vx(i,1:m),vy(i,1:m))   % dibujo de los splines
    plot(vx(i,1:m),vyo(i,1:m),'k')  % dibujo de la función original
    i=i+1;
end
axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                    msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
title('Ajuste por Splines de Quinto Grado Interpolantes');
xlabel('');
hold off


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%    REPRESENTAMOS GRÁFICAMENTE TAMBIÉN LAS DERIVADAS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Calculamos cada una de las derivadas S1,S2,S3,S4 por trozos


for j=1:4
```

```
    i=1;
    while i<=n-1
        Sd(j,i)=diff(S(i),x,j);
        vyd(j,i,1:m)=double(subs(Sd(j,i),x,vx(i,1:m)));
        i=i+1;
    end




end


% Dibujamos la primera derivada

figure('NumberTitle','off','Name','Primera derivada de los splines de
grado 5');
hold on


aux=reshape(vyd(1,:,:),n-1,m);

Msd=max(max(aux));
msd=min(min(aux));

for i=1:n-1
    plot(vx(i,1:m),aux(i,1:m));
end
axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                     msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
title('Primera Derivada de los Splines de Grado 5 Interpolantes');
xlabel('');
clear aux;


% Dibujamos la segunda derivada

figure('NumberTitle','off','Name','Segunda derivada de los splines de
grado 5');
hold on

aux=reshape(vyd(2,:,:),n-1,m);

Msd=max(max(aux));
msd=min(min(aux));

for i=1:n-1
    plot(vx(i,1:m),aux(i,1:m));
end
axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                     msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
title('Segunda Derivada de los Splines de Grado 5 Interpolantes');
xlabel('');
clear aux;
```

```
% Dibujamos la tercera derivada

figure('NumberTitle','off','Name','Tercera derivada de los splines de
grado 5');
hold on

aux=reshape(vyd(3,:,:),n-1,m);

Msd=max(max(aux));
msd=min(min(aux));

for i=1:n-1
   plot(vx(i,1:m),aux(i,1:m));
end
axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                    msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
title('Tercera Derivada de los Splines de Grado 5 Interpolantes');
xlabel('');
clear aux;

% Dibujamos la cuarta derivada

figure('NumberTitle','off','Name','Cuarta derivada de los splines de
grado 5');
hold on

aux=reshape(vyd(4,:,:),n-1,m);

Msd=max(max(aux));
msd=min(min(aux));

for i=1:n-1
   plot(vx(i,1:m),aux(i,1:m));
end
axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                    msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
title('Cuarta Derivada de los Splines de Grado 5 Interpolantes');
xlabel('');
clear aux;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   CALCULAMOS LOS ERRORES COMETIDOS EN NORMA INFINITO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


i=1;
while i<=n-1

   Error_trozo(i)=norm(vy(i,:)-vyo(i,:),inf);
```

```
    i=i+1;

end

Error=max(Error_trozo);
```

LAGRANGE

```
function [C,L,Ye]=Lagrange(X,Y,varargin)

% Construcción del polinomio interpolador de Lagrange que pasa por los
N+1
% puntos(xk,yk) para k=0,1,...,N.
% [C,L,Ye]=Lagrange(X,Y,Xe)
% Variables de entrada
% X es un vector que contiene la lista de las abscisas.
% Y es un vector que contiene la lista de las ordenadas.
% Xe vector de abscisas donde se quiere evaluar el polinomio
% Variables de salida
% C es el vector que contiene los coeficientes del polinomio interpolador
% de Lagrange.
% L es la matriz que contiene los coeficientes de los polinomios
% coeficientes de Lagrange.
% Ye evaluación del polinomio de Lagrange en las abscisas Xe
% Ejemplo.
% Interpolamos la función y=f(x)=cos(x) utilizando los nodos x0=0.0 x1=0.6
% x2=1.2, donde será y0=1 y1=0.825336 e y2=0.362358
%[C,L,Ye]=Lagrange([0 0.6 1.2],[1 0.825336 0.362358])

if nargin==3
    Xe=varargin{1};
else
    Xe='';
    Ye='';
end


w=length(X);
L=zeros(w,w);

%Formación de los polinomios coeficientes de Lagrange.

for k=1:w

    V=1;
    for j=1:w

        if k~=j
            V=conv(V,poly(X(j)))/(X(k)-X(j));
        end

    end

    L(k,:)=V;
```

```
end

% Cálculo de los coeicientes del polinomio interpolador de Lagrange.

C=Y*L;


% Evaluación en Xe

if ~isempty(Xe)
    Ye=polyval(C,Xe);
end
```

CREAR_NODOS.M

```
function [mallado,ordenadasy]=crear_nodos()


% completar vector de abscisas
xa=[0,0.1919,0.5497,0.7847,0.9479,1.0584,1.1433,1.2743,1.4216,1.9949]
;

% le ponemos nombre al fichero donde se guardan los nodos

mallado='nodos_mallado.mat';

% guardamos el fichero de nodos en el directorio

save(mallado,'xa');


% completar vector de ordenadas asociadas

y=[0.1451,0.2415,0.483,0.7245,0.966,1.2075,1.449,1.932,2.415,3.381];


% le ponemos nombre al fichero donde se guardan las ordenadas

ordenadasy='datos_mallado.mat';

% guardemos el fichero con los datos de la función en los nodos

save(ordenadasy,'y');
```

SPLINECINCODATOS.M

```
function
[S,vx,vy]=SplineCincoDatos(mallado,ordenadasy,m,d2i,d4i,d2d,d4d,op)

% Este programa calcula los splines de grado 5 interpolatorios a partir
% de los datos iniciales tanto del mallado como de las ordenadas y que
% toma una cierta función desconocida en los puntos del mallado
%
```

```
% [S]=SplineCincoDatos(mallado,ordenadasy,m,d2i,d4i,d2d,d4d)
%
% Variables de entrada:
% mallado   nombre del fichero que contiene las abscisas xa de los datos
%           usados para construir el spline
% y    ordenadas y de una cierta función desconocida en las abscisas del
%      mallado
% m   número de puntos de evaluación entre la primera y la última abscisa
% d2i valor de la derivada segunda en el extremo izquierdo
%     Si es un caracter en vez de un número se aproximará
% d4i valor de la derivada cuarta en el extremo izquierdo
%     Si es un caracter en vez de un número se aproximará
% d2d valor de la derivada segunda en el extremo derecho
%     Si es un caracter en vez de un número se aproximará
% d4d valor de la derivada cuarta en el extremo derecho
%     Si es un caracter en vez de un número se aproximará
% op  opciones de dibujo, para sacar las 5 gráficas disponibles de los
%     splines cúbicos, '1' para dibujarla, '0' para omitirla.
%
%
% Variables de salida:
% S vector simbólico que contiene las expresiones de los diferentes
% polinomios
% vx abscisas de evaluación
% vy ordenadas que toma el spline en las abscisas vx
%
% Ejemplo aproximando las condiciones de contorno:
% syms x
%
[S,vx,vy]=SplineCincoDatos('coordenadasMangay.mat','coordenadasCalado
z.mat',2000,'A','A','A','A',{'1','0','0','0','0'});
%
% Ejemplo introduciendo las condiciones de contorno naturales:
% syms x
%
[S,vx,vy]=SplineCincoDatos('coordenadasMangay.mat','coordenadasCalado
z.mat',2000,0,0,0,0,{'1','1','1','1','1'});




% Declaramos la variable x como simbólica

syms x


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           DATOS DE ENTRADA
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



% Leemos las abscisas x, creándose una variable xa

load(mallado);
```

```
% longitud del mallado

n=length(xa);

if n<6
    uiwait(msgbox('Debe introducir más de 5 pares de datos', 'Mensaje de
error',...
        'error','modal'))
    return;
end




% Extremos del intervalo

a=xa(1);
b=xa(n);

% Valores de la función en las abscisas, se carga una variable y

load(ordenadasy);


% Generamos los valores de abscisas donde evaluar


i=1;
while i<=n-1

    vx(i,1:m)=linspace(xa(i),xa(i+1),m);
    i=i+1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
%
% APROXIMACIÓN DE LA DERIVADA SEGUNDA Y LA DERIVADA CUARTA EN LOS EXTREMOS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%

% PARA EL EXTREMO IZQUIERDO a


if ischar(d2i) || ischar(d4i)

    % Construímos un polinomio de Lagrange con los seis primeros datos
de la
    % izquierda

    [pizq]=Lagrange(xa(1:6),y(1:6),a);

    % Derivada segunda del polinomio de Lagrange construído en la frontera
    % izquierda
```

```
    dpizq2=polyder(polyder(pizq));

    if ischar(d2i)

        % Evaluación en a de la segunda derivada

        d2a=polyval(dpizq2,a);

    else
        d2a=d2i;

    end

    if ischar(d4i)

        % Derivada cuarta del polinomio de Lagrange construído en la
frontera
        % izquierda

        dpizq4=polyder(polyder(dpizq2));

        % Evaluación en a de la cuarta derivada

        d4a=polyval(dpizq4,a);

    else

        d4a=d4i;

    end

else

    d2a=d2i;
    d4a=d4i;

end



% PARA EL EXTREMO DERECHO b


if ischar(d2d) || ischar(d4d)

    % Construímos un polinomio de Lagrange con los seis últimos datos de
la
    % derecha

    [pder]=Lagrange(xa(n-5:n),y(n-5:n),b);

    % Derivada segunda del polinomio de Lagrange construído en la frontera
    % derecha

    dpder2=polyder(polyder(pder));
```

```
    if ischar(d2d)

        % Evaluación en b de la segunda derivada

        d2b=polyval(dpder2,b);

    else

        d2b=d2d;

    end

    if ischar(d4d)

        % Derivada cuarta del polinomio de Lagrange construído en la
frontera
        % derecha

        dpder4=polyder(polyder(dpder2));

        % Evaluación en b de la cuarta derivada

        d4b=polyval(dpder4,b);

    else

        d4b=d4d;

    end

else

    d2b=d2d;
    d4b=d4d;

end




% Crear los valores de hi en un vector

hi=diff(xa);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%        CREACIÓN DEL SISTEMA DE ECUACIONES LINEALES
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Inicializamos la matriz de coeficientes M
```

```
M=zeros((5*(n-2))+6,5*n-4);


% Crear el bloque numérico
i=0;
t=-2;
g=-4;

% Insertar bloques
while i<n-2
    i=i+1;
    c=[hi(i)^4/120,0,hi(i)^3,0,hi(i),0,0,0,0,0 ; ...

0,3*hi(i)^2,0,1,-1,(hi(i)^3+hi(i+1)^3)/24,0,3*hi(i+1)^2,-1,1;...
        0,-6*hi(i),0,0,0,(hi(i+1)^2-hi(i)^2)/6,0,6*hi(i+1),0,0;...
        0,6,-6,0,0,(hi(i)+hi(i+1))/2,-6,6,0,0;...
        0,hi(i)^3,0,hi(i),0,(hi(i)^4/120),0,0,0,0];
    t=t+5;   %fila
    g=g+5;   %columna
    M(t:t+4,g:g+9)=c;
end



% Insertar condiciones izquierda.
M(1,1:5)=[hi(1)^2/6,0,6*hi(1),0,0];      %S1'' en x1
M(2,1:5)=[1,0,0,0,0];      %S1'''' en x1

% Insertar condiciones derecha.
t=t+5;   %fila
g=g+5;   %columna

M(t,g:g+5)=[hi(n-1)^4/120,0,hi(n-1)^3,0,hi(n-1),0];
M(t+1,g:g+5)=[0,hi(n-1)^3,0,hi(n-1),0,hi(n-1)^4/120];
M(t+2,g:g+5)=[0,6*hi(n-1),0,0,0,hi(n-1)^2/6];
M(t+3,g:g+5)=[0,0,0,0,0,1];

% Vector de terminos independientes
V=transpose(zeros(size(1:(5*(n-2))+6)));

i=3;
j=1;
while i<5*(n-2)+6
    V(i,1)=y(1,j);
    V(i-1,1)=y(1,j);
    i=i+5;
    j=j+1;
end

V(1,1)=d2a;
V(2,1)=d4a;

V(5*(n-2)+4,1)=y(n);
V(5*(n-2)+5,1)=d2b;
V(5*(n-2)+6,1)=d4b;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   RESOLVEMOS EL SISTEMA DE ECUACIONES LINEALES
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


X=linsolve(M,V);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  CREAMOS CADA UNO DE LOS POLINOMIOS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Crear Polinomios y coordenadas para representar

syms S

i=1;
j=1;
while i<=n-1
    S(i,1)=((X(j)/(120*hi(i)))*(xa(i+1)-x)^5)...
        +((X(j+5)/(120*hi(i)))*(x-xa(i))^5)...
        +(X(j+1)*(x-xa(i))^3)...
        +(X(j+2)*(xa(i+1)-x)^3)...
        +(X(j+3)*(x-xa(i)))...
        +(X(j+4)*(xa(i+1)-x));


    vx(i,1:m)=linspace(xa(i),xa(i+1),m);
    vy(i,1:m)=subs(S(i),x,vx(i,1:m));

    i=i+1;
    j=j+5;
end

vy=double(vy);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   REPRESENTAMOS GRÁFICAMENTE EL RESULTADO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if strcmp(op{1},'1')


    figure('NumberTitle','off','Name','Splines de Quinto Grado')
    hold on

    % Dibujamos los puntos de control

    plot(xa,y,'ko','MarkerSize',2,'LineWidth',3);


    % Dibujamos los splines
```

```
    Msd=max(max(vy));
    msd=min(min(vy));

    i=1;
    while i<n
        plot(vx(i,1:m),vy(i,1:m))   % dibujo de los splines
        i=i+1;
    end
    axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                       msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
    title('Ajuste por Splines de Quinto Grado Interpolantes');
    xlabel('');
    hold off

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%    REPRESENTAMOS GRÁFICAMENTE TAMBIÉN LAS DERIVADAS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Calculamos cada una de las derivadas S1,S2,S3,S4 por trozos


for j=1:4

    i=1;
    while i<=n-1
        Sd(j,i)=diff(S(i),x,j);
        vyd(j,i,1:m)=double(subs(Sd(j,i),x,vx(i,1:m)));
        i=i+1;
    end




end


if strcmp(op{2},'1')

    % Dibujamos la primera derivada


    figure('NumberTitle','off','Name','Primera derivada de los splines
de grado 5');
    hold on


    aux=reshape(vyd(1,:,:),n-1,m);

    Msd=max(max(aux));
```

```
    msd=min(min(aux));

    for i=1:n-1
        plot(vx(i,1:m),aux(i,1:m));
    end
    axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                    msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
    title('Primera Derivada de los Splines de Grado 5 Interpolantes');
    xlabel('');
    clear aux;

end


if strcmp(op{3},'1')

    % Dibujamos la segunda derivada

    figure('NumberTitle','off','Name','Segunda derivada de los splines
de grado 5');
    hold on

    aux=reshape(vyd(2,:,:),n-1,m);

    Msd=max(max(aux));
    msd=min(min(aux));

    for i=1:n-1
        plot(vx(i,1:m),aux(i,1:m));
    end
    axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                    msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
    title('Segunda Derivada de los Splines de Grado 5 Interpolantes');
    xlabel('');
    clear aux;


end


if strcmp(op{4},'1')


    % Dibujamos la tercera derivada

    figure('NumberTitle','off','Name','Tercera derivada de los splines
de grado 5');
    hold on

    aux=reshape(vyd(3,:,:),n-1,m);

    Msd=max(max(aux));
    msd=min(min(aux));

    for i=1:n-1
```

```
        plot(vx(i,1:m),aux(i,1:m));
    end
    axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                       msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
    title('Tercera Derivada de los Splines de Grado 5 Interpolantes');
    xlabel('');
    clear aux;

end


if strcmp(op{5},'1')


    % Dibujamos la cuarta derivada

    figure('NumberTitle','off','Name','Cuarta derivada de los splines de
grado 5');
    hold on

    aux=reshape(vyd(4,:,:),n-1,m);

    Msd=max(max(aux));
    msd=min(min(aux));

    for i=1:n-1
        plot(vx(i,1:m),aux(i,1:m));
    end
    axis(double([vx(1,1)-0.1*(vx(n-1,m)-vx(1,1))
vx(n-1,m)+0.1*(vx(n-1,m)-vx(1,1))                       msd-0.1*(Msd-msd)
Msd+0.1*(Msd-msd)]));
    title('Cuarta Derivada de los Splines de Grado 5 Interpolantes');
    xlabel('');
    clear aux;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   SACAMOS LAS EXPRESIONES DE LOS POLINOMIOS A UN FICHERO
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid=fopen('polinomios_splines.txt','w');

for i=1:n-1
    fprintf(fid,'%s\n\n',char(S(i)));
end

fclose(fid);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%    SACAMOS LOS VALORES INTERPOLADOS EN CADA SUBINTERVALO
%    A UN FICHERO POR COLUMNAS
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid=fopen('resul_valores_interpolados.txt','w');

for j=1:m

    for i=1:n-1
        fprintf(fid,'%16.16f                                    %16.16f
',vx(i,j),vy(i,j));
    end

    fprintf(fid,'\n');
end



fclose(fid);
```

SPLINES5.M (ASOCIADO A LA INTERFAZ GRÁFICA)

```
function varargout = splines5(varargin)
% SPLINES5 MATLAB code for splines5.fig
%      SPLINES5, by itself, creates a new SPLINES5 or raises the existing
%      singleton*.
%
%      H = SPLINES5 returns the handle to a new SPLINES5 or the handle to
%      the existing singleton*.
%
%      SPLINES5('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in SPLINES5.M with the given input
arguments.
%
%      SPLINES5('Property','Value',...) creates a new SPLINES5 or raises
the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before splines5_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to splines5_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help splines5

% Last Modified by GUIDE v2.5 21-Jul-2018 11:11:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```

```
                 'gui_Singleton',  gui_Singleton, ...
                 'gui_OpeningFcn', @splines5_OpeningFcn, ...
                 'gui_OutputFcn',  @splines5_OutputFcn, ...
                 'gui_LayoutFcn',  [] , ...
                 'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before splines5 is made visible.
function splines5_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to splines5 (see VARARGIN)

% Choose default command line output for splines5
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes splines5 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%% CENTRAR INTERFAZ GRÁFICA EN PANTALLA %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

set( handles.output,'Units', 'pixels' );
screenSize = get(0, 'ScreenSize');

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




% --- Outputs from this function are returned to the command line.
function varargout = splines5_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% INCLUIMOS EL ESCUDO DE LA UPCT %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Importamos imagen *.jpg
escudo_upct=imread('escudo.jpg');
axes(handles.escudo);

%imshow(escudo_upct);
imagesc(escudo_upct); axis off



function edit_nodos_x_Callback(hObject, eventdata, handles)
% hObject    handle to edit_nodos_x (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_nodos_x as text
%        str2double(get(hObject,'String')) returns contents of
edit_nodos_x as a double


% --- Executes during object creation, after setting all properties.
function edit_nodos_x_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_nodos_x (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit_nodos_fx_Callback(hObject, eventdata, handles)
% hObject    handle to edit_nodos_fx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_nodos_fx as text
%        str2double(get(hObject,'String')) returns contents of
edit_nodos_fx as a double


% --- Executes during object creation, after setting all properties.
function edit_nodos_fx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_nodos_fx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%      See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton_CrearNodos.
function pushbutton_CrearNodos_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_CrearNodos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

edit crear_nodos.m

[mallado,ordenadasy]=crear_nodos();

set(handles.edit_nodos_x,'String',mallado);
set(handles.edit_nodos_fx,'String',ordenadasy);


% --- Executes on button press in pushbutton_CargarNodos.
function pushbutton_CargarNodos_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_CargarNodos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename,pathname,filterindex]=uigetfile( ...
    {  '*.mat',  'Abscisas  de  los  puntos  usados  para  construir  la
reconstrucción (*.mat)'},'Seleccione un archivo');

if filterindex == 1
    set(handles.edit_nodos_x,'String',filename);
end


[filename,pathname,filterindex]=uigetfile( ...
    {  '*.mat',  'Ordenadas  de  los  puntos  usados  para  construir  la
reconstrucción (*.mat)'},'Seleccione un archivo');

if filterindex == 1
    set(handles.edit_nodos_fx,'String',filename);
end




function edit_m_Callback(hObject, eventdata, handles)
% hObject    handle to edit_m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_m as text
%       str2double(get(hObject,'String')) returns contents of edit_m as
a double
```

```
% --- Executes during object creation, after setting all properties.
function edit_m_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




% --- Executes on button press in radiobutton_2d_izq_apro.
function radiobutton_2d_izq_apro_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton_2d_izq_apro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%    Hint:    get(hObject,'Value')    returns    toggle    state    of
radiobutton_2d_izq_apro

if get(handles.radiobutton_2d_izq_apro,'Value')==1
    set(handles.edit_2d_izq_valor,'Enable','off');
    set(handles.radiobutton_2d_izq_valor,'Value',0);
else
    set(handles.radiobutton_2d_izq_valor,'Value',1);
    set(handles.edit_2d_izq_valor,'Enable','on');
end



% --- Executes on button press in radiobutton_2d_izq_valor.
function     radiobutton_2d_izq_valor_Callback(hObject,     eventdata,
handles)
% hObject    handle to radiobutton_2d_izq_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%    Hint:    get(hObject,'Value')    returns    toggle    state    of
radiobutton_2d_izq_valor


if get(handles.radiobutton_2d_izq_valor,'Value')==1
    set(handles.edit_2d_izq_valor,'Enable','on');
    set(handles.radiobutton_2d_izq_apro,'Value',0);
else
    set(handles.radiobutton_2d_izq_apro,'Value',1);
    set(handles.edit_2d_izq_valor,'Enable','off');
end


function edit_2d_izq_valor_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_2d_izq_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_2d_izq_valor as
text
%              str2double(get(hObject,'String'))  returns  contents  of
edit_2d_izq_valor as a double


% --- Executes during object creation, after setting all properties.
function edit_2d_izq_valor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_2d_izq_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in radiobutton_4d_izq_apro.
function radiobutton_4d_izq_apro_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton_4d_izq_apro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%    Hint:    get(hObject,'Value')    returns    toggle    state    of
radiobutton_4d_izq_apro

if get(handles.radiobutton_4d_izq_apro,'Value')==1
    set(handles.edit_4d_izq_valor,'Enable','off');
    set(handles.radiobutton_4d_izq_valor,'Value',0);
else
    set(handles.radiobutton_4d_izq_valor,'Value',1);
    set(handles.edit_4d_izq_valor,'Enable','on');
end



% --- Executes on button press in radiobutton_4d_izq_valor.
function     radiobutton_4d_izq_valor_Callback(hObject,     eventdata,
handles)
% hObject    handle to radiobutton_4d_izq_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%    Hint:    get(hObject,'Value')    returns    toggle    state    of
radiobutton_4d_izq_valor


if get(handles.radiobutton_4d_izq_valor,'Value')==1
    set(handles.edit_4d_izq_valor,'Enable','on');
    set(handles.radiobutton_4d_izq_apro,'Value',0);
```

```
else
    set(handles.radiobutton_4d_izq_apro,'Value',1);
    set(handles.edit_4d_izq_valor,'Enable','off');
end
```

```
function edit_4d_izq_valor_Callback(hObject, eventdata, handles)
% hObject    handle to edit_4d_izq_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_4d_izq_valor as
text
%        str2double(get(hObject,'String')) returns contents of
edit_4d_izq_valor as a double


% --- Executes during object creation, after setting all properties.
function edit_4d_izq_valor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_4d_izq_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in radiobutton_2d_der_apro.
function radiobutton_2d_der_apro_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton_2d_der_apro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
radiobutton_2d_der_apro


if get(handles.radiobutton_2d_der_apro,'Value')==1
    set(handles.edit_2d_der_valor,'Enable','off');
    set(handles.radiobutton_2d_der_valor,'Value',0);
else
    set(handles.radiobutton_2d_der_valor,'Value',1);
    set(handles.edit_2d_der_valor,'Enable','on');
end
```

```
% --- Executes on button press in radiobutton_2d_der_valor.
function    radiobutton_2d_der_valor_Callback(hObject,    eventdata,
handles)
% hObject    handle to radiobutton_2d_der_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%   Hint:   get(hObject,'Value')   returns   toggle   state   of
radiobutton_2d_der_valor


if get(handles.radiobutton_2d_der_valor,'Value')==1
    set(handles.edit_2d_der_valor,'Enable','on');
    set(handles.radiobutton_2d_der_apro,'Value',0);
else
    set(handles.radiobutton_2d_der_apro,'Value',1);
    set(handles.edit_2d_der_valor,'Enable','off');
end




function edit_2d_der_valor_Callback(hObject, eventdata, handles)
% hObject    handle to edit_2d_der_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_2d_der_valor as
text
%              str2double(get(hObject,'String'))  returns  contents  of
edit_2d_der_valor as a double


% --- Executes during object creation, after setting all properties.
function edit_2d_der_valor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_2d_der_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if       ispc       &&       isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in radiobutton_4d_der_apro.
function radiobutton_4d_der_apro_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton_4d_der_apro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of
radiobutton_4d_der_apro


if get(handles.radiobutton_4d_der_apro,'Value')==1
    set(handles.edit_4d_der_valor,'Enable','off');
    set(handles.radiobutton_4d_der_valor,'Value',0);
else
    set(handles.radiobutton_4d_der_valor,'Value',1);
    set(handles.edit_4d_der_valor,'Enable','on');
end




% --- Executes on button press in radiobutton_4d_der_valor.
function radiobutton_4d_der_valor_Callback(hObject, eventdata,
handles)
% hObject    handle to radiobutton_4d_der_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
radiobutton_4d_der_valor


if get(handles.radiobutton_4d_der_valor,'Value')==1
    set(handles.edit_4d_der_valor,'Enable','on');
    set(handles.radiobutton_4d_der_apro,'Value',0);
else
    set(handles.radiobutton_4d_der_apro,'Value',1);
    set(handles.edit_4d_der_valor,'Enable','off');
end




function edit_4d_der_valor_Callback(hObject, eventdata, handles)
% hObject    handle to edit_4d_der_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_4d_der_valor as
text
%        str2double(get(hObject,'String')) returns contents of
edit_4d_der_valor as a double


% --- Executes during object creation, after setting all properties.
function edit_4d_der_valor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_4d_der_valor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if        ispc        &&        isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end




% --- Executes on button press in checkbox_splines5.
function checkbox_splines5_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_splines5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_splines5


% --- Executes on button press in checkbox_1d.
function checkbox_1d_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_1d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_1d


% --- Executes on button press in checkbox_2d.
function checkbox_2d_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_2d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_2d


% --- Executes on button press in checkbox_3d.
function checkbox_3d_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_3d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_3d


% --- Executes on button press in checkbox_4d.
function checkbox_4d_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_4d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_4d
```

```
% --- Executes on button press in pushbutton_Aplicar.
function pushbutton_Aplicar_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_Aplicar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LEEMOS LOS DATOS INICIALES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Recogemos los nodos

nodosx= get(handles.edit_nodos_x,'String');
nodosx=fliplr(deblank(fliplr(deblank(nodosx)))); % Eliminamos huecos en
blanco

nodosfx= get(handles.edit_nodos_fx,'String');
nodosfx=fliplr(deblank(fliplr(deblank(nodosfx)))); % Eliminamos huecos
en blanco


% Comprobamos que no están vacíos
if isempty(nodosx)==1 || isempty(nodosfx)==1
   uiwait(msgbox('Debe introducir correctamente los nodos por los que
ha de pasar el Spline', 'Mensaje de error',...
       'error','modal'))
   return;
end


% Se genera el vector xa de abscisas usadas para interpolar

load(nodosx)

% Se genera el vector y de ordenadas en las abscisas xa

load(nodosfx)



% Leemos el número de puntos en que se quiere dividir cada subintervalo
% para evaluar los trozos del spline

m= get(handles.edit_m,'String');
m=str2num(m);

if isempty(m)
  uiwait(msgbox('Debe introducir correctamente un valor para m',
'Mensaje de error',...
       'error','modal'))
   return;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CONDICIONES DE CONTORNO
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Inicializamos algunas variables por defecto

d2i='A'; d4i='A'; d2d='A'; d4d='A';


% Frontera Izquierda

% Derivada segunda en la frontera izquierda


if get(handles.radiobutton_2d_izq_valor,'Value')==1

   % Obtenemos la segunda derivada a la izquierda

       aux = get(handles.edit_2d_izq_valor,'String');
       d2i=sscanf(aux, '%f');

       % Comprobamos que no está vacío
       if isempty(d2i)==1
          uiwait(msgbox('Debe introducir un valor para la segunda
derivada en la frontera izquierda.', 'Mensaje de error',...
              'error','modal'))
          return;
       end

end


% Derivada cuarta en la frontera izquierda


if get(handles.radiobutton_4d_izq_valor,'Value')==1

   % Obtenemos la cuarta derivada a la izquierda

       aux = get(handles.edit_4d_izq_valor,'String');
       d4i=sscanf(aux, '%f');

       % Comprobamos que no está vacío
       if isempty(d4i)==1
          uiwait(msgbox('Debe introducir un valor para la cuarta derivada
en la frontera izquierda.', 'Mensaje de error',...
              'error','modal'))
          return;
       end

end


% Frontera Derecha

% Derivada segunda en la frontera derecha
```

```
if get(handles.radiobutton_2d_der_valor,'Value')==1

   % Obtenemos la segunda derivada a la derecha

       aux = get(handles.edit_2d_der_valor,'String');
       d2d=sscanf(aux, '%f');

       % Comprobamos que no está vacío
       if isempty(d2d)==1
           uiwait(msgbox('Debe introducir un valor para la segunda derivada en la frontera derecha.', 'Mensaje de error',...
               'error','modal'))
           return;
       end

end


% Derivada cuarta en la frontera derecha


if get(handles.radiobutton_4d_der_valor,'Value')==1

   % Obtenemos la cuarta derivada a la derecha

       aux = get(handles.edit_4d_der_valor,'String');
       d4d=sscanf(aux, '%f');

       % Comprobamos que no está vacío
       if isempty(d4d)==1
           uiwait(msgbox('Debe introducir un valor para la cuarta derivada en la frontera derecha.', 'Mensaje de error',...
               'error','modal'))
           return;
       end

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LEE LAS OPCIONES DE GRÁFICAS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


opg0=num2str(get(handles.checkbox_splines5,'Value'));
opg1=num2str(get(handles.checkbox_1d,'Value'));
opg2=num2str(get(handles.checkbox_2d,'Value'));
opg3=num2str(get(handles.checkbox_3d,'Value'));
opg4=num2str(get(handles.checkbox_4d,'Value'));

op={opg0,opg1,opg2,opg3,opg4};



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LLAMADA A LA FUNCIÓN QUE CALCULA LOS SPLINES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Borra los ficheros anteriores
% if exist('polinomios_splines.txt','file')
%   delete polinomios_splines.txt;
% end
% if exist('resul_valores_interpolados.txt','file')
%   delete resul_valores_interpolados.txt;
% end



[S,vx,vy]=SplineCincoDatos(nodosx,nodosfx,m,d2i,d4i,d2d,d4d,op);






% --- Executes on button press in pushbutton_Ficheros.
function pushbutton_Ficheros_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_Ficheros (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

edit polinomios_splines.txt;
edit resul_valores_interpolados.txt;




% --- Executes on button press in pushbutton_CerrarGraficas.
function    pushbutton_CerrarGraficas_Callback(hObject,    eventdata,
handles)
% hObject    handle to pushbutton_CerrarGraficas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Cerramos las gráficas
% get( handles.output , 'Tag' ) is the 'Tag' of the GUI
Figures   =   findobj(   'Type',    'Figure'   ,   '-not'   ,   'Tag'   ,
get( handles.output , 'Tag' ) );
NFigures = length( Figures );
for nFigures = 1 : NFigures
  close( Figures( nFigures ) );
end;


% --- Executes on button press in pushbutton_Reset.
function pushbutton_Reset_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_Reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Cerramos las gráficas
% get( handles.output , 'Tag' ) is the 'Tag' of the GUI
Figures   =   findobj(   'Type',    'Figure'   ,   '-not'   ,   'Tag'   ,
get( handles.output , 'Tag' ) );
NFigures = length( Figures );
for nFigures = 1 : NFigures
```

```
  close( Figures( nFigures ) );
end;

% Inicializamos el panel de datos iniciales

set(handles.edit_nodos_x,'String','');
set(handles.edit_nodos_fx,'String','');
set(handles.edit_m,'String','100');


% Inicializamos el panel de condiciones de contorno

% para la derivada segunda por la izquierda

set(handles.radiobutton_2d_izq_apro,'Value',1);
set(handles.radiobutton_2d_izq_valor,'Value',0);
set(handles.edit_2d_izq_valor,'Enable','off');
set(handles.edit_2d_izq_valor,'String','');

% para la derivada cuarta por la izquierda

set(handles.radiobutton_4d_izq_apro,'Value',1);
set(handles.radiobutton_4d_izq_valor,'Value',0);
set(handles.edit_4d_izq_valor,'Enable','off');
set(handles.edit_4d_izq_valor,'String','');

% para la derivada segunda por la derecha

set(handles.radiobutton_2d_der_apro,'Value',1);
set(handles.radiobutton_2d_der_valor,'Value',0);
set(handles.edit_2d_der_valor,'Enable','off');
set(handles.edit_2d_der_valor,'String','');

% para la derivada cuarta por la derecha

set(handles.radiobutton_4d_der_apro,'Value',1);
set(handles.radiobutton_4d_der_valor,'Value',0);
set(handles.edit_4d_der_valor,'Enable','off');
set(handles.edit_4d_der_valor,'String','');


% Inicializamos el panel de graficas

set(handles.checkbox_splines5,'Value',1);
set(handles.checkbox_1d,'Value',0);
set(handles.checkbox_2d,'Value',0);
set(handles.checkbox_3d,'Value',0);
set(handles.checkbox_4d,'Value',0);




% --- Executes on button press in pushbutton_Salir.
function pushbutton_Salir_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_Salir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)


salir=questdlg('¿Desea    salir    del    programa?','Salida    del
Programa','Si','No','No');
switch salir
    case 'Si'
        close all;
    case 'No'
        return;
end





%
---------------------------------------------------------------------
function Ayuda_Callback(hObject, eventdata, handles)
% hObject    handle to Ayuda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


%
---------------------------------------------------------------------
function TFG_Callback(hObject, eventdata, handles)
% hObject    handle to TFG (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

winopen('Tfg.pdf');

%
---------------------------------------------------------------------
function Tutorial_Callback(hObject, eventdata, handles)
% hObject    handle to Tutorial (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

winopen('tutorial.pdf');


%
---------------------------------------------------------------------
function Acerca_Callback(hObject, eventdata, handles)
% hObject    handle to Acerca (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

winopen('AcercaDe.pdf');
```

BIBLIOGRAFÍA

- GALLEGO VALDELLÓS, Irene; TRILLO MOYA, Juan Carlos y ANGOSTO HERNANDEZ, Carlos. Proyecto Fin de Carrera. Splines cúbicos interpolantes. Universidad Politécnica de Cartagena (UPCT).

- RONCERO PEÑA, Blanca; TRILLO MOYA, Juan Carlos. Proyecto Fin de Carrera. Splines cúbicos suavizantes. Universidad Politécnica de Cartagena (UPCT).

- CERÓN COLMENA, Cristina; TRILLO MOYA, Juan Carlos. Estudio sobre esquemas de subdivisión no lineales en mallados no uniformemente espaciados. Universidad Politécnica de Cartagena (UPCT).

- CID ARAÚJO, José Ángel. Interpolación con funciones splines. Universidad de Jaén. (UJA)

- GONZALEZ MORCILLO, Carlos. Splines, Curvas y superficies. Introducción al dibujo de curvas de aproximación e interpolación por Computador. Universidad de Castilla La Mancha (UCLM).

- MR.ZEDZED. batiburrillosubmarino.wordpress.com . Blog vinculado a el Master de Historia de Patrimonio Naval. Universidad de Murcia. Cátedra Naval.