



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

Grado en Ingeniería Mecánica

**Técnicas y estrategias de interpolación
para la resolución de problemas
formulados en términos de EDOs de 2º
orden en el ámbito de la Ingeniería
Industrial**

Autor:

Fernández Ordóñez, Guillermo

Tutor:

Álvarez López, Jorge
Departamento Matemática
Aplicada

Valladolid, julio 2022.





AGRADECIMIENTOS

Me gustaría agradecer a todas las personas que, bien directamente o indirectamente, han colaborado en la realización de este proyecto y en mi formación como ingeniero durante estos cuatro años.

En primer lugar, quiero dar las gracias a mi tutor Jorge Álvarez, por todo su empeño y dedicación continua a la hora de aconsejarme para llevar a cabo el trabajo. Sin su labor llegar al final no habría sido posible.

Quiero agradecer también a mi padre Manuel el apoyo que me ha brindado no sólo durante estos cuatro años de carrera, sino durante toda mi vida.

Tampoco me quiero olvidar de aquellos amigos que me ha dado la Universidad, en especial de Juan y Madi.

Y, por último, querría agradecer especialmente a mi amiga Paula todo el tiempo que ha tenido que aguantarme, motivarme y hacerme creer que podía conseguirlo.

Gracias a todos.



RESUMEN

Infinidad de problemas de Ingeniería y de otras ramas superiores de la ciencia se formulan en términos de Ecuaciones Diferenciales Ordinarias de 2^º orden. En muchas ocasiones, sus soluciones no pueden obtenerse analíticamente, por lo que constituyen un campo de investigación muy interesante en cuanto a métodos numéricos se refiere, ya que no abundan demasiados en comparación con los problemas de 1^{er} orden. Este trabajo se centra en el empleo y desarrollo de técnicas y estrategias de interpolación para la aproximación continua de soluciones de Problemas de Valores Iniciales y Problemas de Valores en la Frontera de 2^º orden. Se emplean métodos de un paso *Runge-Kutta* y *Runge-Kutta-Nyström* que computan soluciones discretas para interpolación posterior mediante estrategias basadas en *splines* y *pseudo-splines*, con continuidad C^1 y C^2 . Todos estos métodos y estrategias se ensayan convenientemente con un buen número de experimentos numéricos para estudiar su precisión y orden.

PALABRAS CLAVE

Ecuaciones Diferenciales Ordinarias, Splines, Estrategias de Interpolación, Métodos Numéricos de un Paso, Método del Disparo Lineal

ABSTRACT

Countless engineering problems and from other branches of science are formulated in terms of 2nd order Ordinary Differential Equations. On many occasions, their solutions cannot be obtained analytically, which is the reason why they constitute a very interesting field of research in terms of numerical methods, since there are not too many of them compared to 1st order problems. This work focuses on the use and development of interpolation techniques and strategies for the continuous approximation of solutions of 2nd order Initial Value Problems and Boundary Value Problems. Single-step *Runge-Kutta* and *Runge-Kutta-Nyström* methods are used to compute discrete solutions for subsequent interpolation using strategies based on *splines* and *pseudo-splines*, with C^1 and C^2 continuity. All of these methods and strategies are conveniently tested with a good number of numerical experiments in order to study their precision and order.

KEY WORDS

Ordinary Differential Equations, Splines, Interpolation Strategies, Single-step Numerical Methods, Linear Shooting Method





ÍNDICES





ÍNDICE DE CONTENIDOS.

CAPÍTULO 1. Introducción, objetivos y estructura del trabajo.....	1
1.1. Introducción y justificación.	3
1.2. Objetivos del TFG.	4
1.3. Estructura de la memoria.....	6
CAPÍTULO 2. Problemas de Valores Iniciales de 2º orden. Nomenclatura en métodos numéricos.	9
2.1. Ecuaciones diferenciales ordinarias de 2º orden.	11
2.1.1. Consideraciones sobre la notación de las derivadas.....	12
2.2. Introducción a los Problemas de Valores Iniciales de 2º orden.	13
2.2.1. Teorema de existencia y unicidad de la solución de un PVI.....	14
2.3. Nomenclatura en métodos numéricos.....	15
CAPÍTULO 3. Métodos numéricos discretos de un paso. 17	
3.1. Introducción a los métodos numéricos de un paso.	19
3.2. Métodos <i>Runge-Kutta</i>	19
3.2.1. Fundamentos teóricos relativos al PVI de 1er orden.....	20
3.2.2. Orden de convergencia de los métodos <i>Runge-Kutta</i>	21
3.2.3. Método de <i>Euler</i> explícito.....	22
3.2.4. Método de <i>Heun</i> de orden 3.....	22
3.2.5. Método <i>Runge-Kutta</i> explícito de orden 4.....	23
3.2.6. Método de <i>Gauss</i> implícito de 3 etapas y orden 6.....	24
3.2.7. Generalización al PVI de 2º orden.....	25
3.2.7.1. Teorema de existencia y unicidad de las soluciones de sistemas de orden p compuestos por PVI de 1er orden.....	26
3.2.7.2. Esquemas <i>Runge-Kutta</i> para resolver PVI de 2º orden mediante sistemas de EDOs de 1er orden.....	27
3.2.8. Pares encajados de métodos <i>Runge-Kutta</i>	28
3.2.9. Métodos <i>Runge-Kutta</i> implementados en solvers para PVI de MATLAB. Problemas rígidos y no rígidos.....	29
3.3. Métodos <i>Runge-Kutta-Nyström</i>	31
3.3.1. Obtención de los métodos <i>Runge-Kutta-Nyström</i> a partir de los métodos <i>Runge-Kutta</i>	31



3.3.2. Métodos *Runge-Kutta-Nyström* para problemas independientes de y' 32

3.3.3. Orden de convergencia de los métodos *Runge-Kutta-Nyström*. 33

3.3.4. Métodos *Runge-Kutta-Nyström* explícitos para el PVI completo de la forma $y''(x) = f(x, y(x), y'(x))$ 34

 3.3.4.1. Método *RKN* directo de orden 3. 34

 3.3.4.2. Método *RKN* transformado de orden 3. 35

 3.3.4.3. Método *RKN* de orden 4. 35

3.3.5. Métodos *Runge-Kutta-Nyström* explícitos para el PVI independiente de y' de la forma $y''(x) = f(x, y(x))$ 36

 3.3.5.1. Método *RKN* simple de orden 4. 36

 3.3.5.2. Método *RKN* simple de orden 5. 37

 3.3.5.3. Método *RKN* especial de orden 5. 37

CAPÍTULO 4. Estrategias de interpolación de soluciones de PVIs empleando *splines*. 39

4.1. Introducción a las estrategias de interpolación. 41

4.2. Utilización de *splines* como estrategia para interpolar soluciones de PVI. 41

4.3. Estrategia empleando *splines* cuadráticos interpoladores. 43

 4.3.1. Justificación del empleo de *splines* cuadráticos. 43

 4.3.2. Resolución del intervalo inicial. Obtención de S_1 en $[t_0, t_1]$, $k = 1$ 44

 4.3.3. Resolución de un intervalo genérico. Obtención de S_k en $[t_{k-1}, t_k]$, $k = 2, 3, \dots, n$ 45

 4.3.4. Resumen del procedimiento. 46

4.4. Estrategia empleando *splines* cúbicos interpoladores. 47

 4.4.1. Justificación del empleo de *splines* cúbicos. 47

 4.4.2. Resolución del intervalo inicial. Obtención de S_1 en $[t_0, t_1]$, $k = 1$ 48

 4.4.3. Resolución de un intervalo genérico. Obtención de S_k en $[t_{k-1}, t_k]$, $k = 2, 3, \dots, n$ 49

 4.4.4. Resumen del procedimiento. 50

CAPÍTULO 5. Estrategias de interpolación de soluciones de PVIs empleando *pseudo-splines*. 53

5.1. Introducción. Definición de *pseudo-spline*. 55

5.2. Pseudo-splines cuárticos interpoladores.....	56
5.2.1. Condiciones adicionales. Estrategia de colocación.	57
5.2.1.1. Particularización al caso lineal.	58
5.2.2. Condiciones adicionales. Estrategia de interpolación intermedia y colocación en el nodo inicial $t_0 = a$	58
5.2.3. Condiciones adicionales. Estrategia de colocación intermedia y colocación en el nodo inicial $t_0 = a$	59
5.2.3.1 Particularización al caso lineal.	59
5.2.4. Resolución del intervalo inicial. Obtención de S_1 en $[t_0, t_1]$, $k = 1$	59
5.2.4.1. Condiciones adicionales. Estrategia de colocación.	61
5.2.4.1.1. Particularización a problemas independientes de y'	61
5.2.4.1.2. Particularización al caso lineal.	62
5.2.4.2. Condiciones adicionales. Estrategia de interpolación intermedia y colocación en el nodo inicial $t_0 = a$	63
5.2.4.3. Condiciones adicionales. Estrategia de colocación intermedia y colocación en el nodo inicial $t_0 = a$	64
5.2.4.3.1. Particularización al caso lineal.	64
5.2.5. Resolución de un intervalo genérico. Obtención de S_k en $[t_{k-1}, t_k]$, $k = 2, 3, \dots, n$	65
5.2.5.1. Condiciones adicionales. Estrategia de colocación.	67
5.2.5.1.1. Particularización a problemas independientes de y'	67
5.2.5.1.2. Particularización al caso lineal.	68
5.2.5.2. Condiciones adicionales. Estrategia de interpolación intermedia.	69
5.2.5.3. Condiciones adicionales. Estrategia de colocación intermedia.	69
5.2.5.3.1. Particularización al caso lineal.	70
5.2.6. Resumen del procedimiento.....	71
5.2.7. Resumen de la implementación en <i>MATLAB</i>	71
5.3. Estrategias de interpolación con polinomios osculadores.	72
5.3.1. Definición del polinomio osculador P de grado m	73
5.3.2. Estrategia de interpolación con <i>pseudo-splines</i> cúbicos basada en los polinomios interpoladores de <i>Hermite</i>	74
5.3.2.1. El polinomio interpolador de <i>Hermite</i>	75



5.3.2.1.1. Teorema de existencia y unicidad del polinomio interpolador de <i>Hermite</i>	75
5.3.2.1.2. Fórmula de diferencias divididas de <i>Newton</i> para polinomios de <i>Hermite</i>	77
5.3.2.2. Resolución de un intervalo genérico k . Obtención de S_k en $[t_{k-1}, t_k]$, $k = 1, 2, \dots, n$	78
5.3.3. Estrategia de interpolación con <i>pseudo-splines</i> quínticos basada en polinomios osculadores.	80
5.3.3.1. Resolución de un intervalo genérico k . Obtención de S_k en $[t_{k-1}, t_k]$, $k = 1, 2, \dots, n$	80
5.3.4. Resumen de la implementación en <i>MATLAB</i>	83
CAPÍTULO 6. Método del disparo para Problemas de Valores en la Frontera lineales.	85
6.1. Introducción a los Problemas de Valores en la Frontera (<i>PVFs</i>) de 2º orden.....	87
6.2. Clasificación de los <i>PVF</i> de 2º orden según la tipología de condiciones de contorno.	87
6.3. Existencia y unicidad de la solución de un <i>PVF</i> de 2º orden.	88
6.3.1. Teorema de existencia y unicidad de la solución. Condiciones de contorno simples.....	88
6.3.2. Nociones sobre la existencia y unicidad de la solución. Condiciones de contorno generales.	89
6.3.2.1. Teorema 1.....	90
6.3.2.1. Teorema 2.....	90
6.4. El método del disparo para <i>PVFs</i> de 2º orden lineales.....	91
6.4.1. Método del disparo lineal. Condiciones de contorno simples.	91
6.4.1.1. Ejemplo. Algoritmo de resolución con nodos equiespaciados.	94
6.4.1.2. Condiciones de contorno simples. Particularización a $y_2(b) = 0$	96
6.4.2. Método del disparo lineal. Condiciones de contorno generales....	97
CAPÍTULO 7. Experimentos numéricos.....	101
7.1. Introducción y estructura del capítulo.	103
7.2. Experimento numérico 1. Vibraciones lineales de 1 grado de libertad.	104
7.2.1. Solución exacta y numérica. Estudio del orden empírico de los métodos discretos.....	107

7.2.1.1. Solución analítica $x(t)$ y primera derivada $\dot{x}(t)$	107
7.2.1.2. Estudio del $ EGT $ para cada método numérico propuesto.	109
7.2.1.2.1. $ EGT $ para el método de <i>Euler</i> explícito.	109
7.2.1.2.2. $ EGT $ de métodos numéricos de orden 3.....	110
7.2.1.2.3. $ EGT $ de métodos numéricos de orden 3.....	111
7.2.1.3. Determinación del orden empírico de los métodos numéricos. Comparación con el orden teórico.....	112
7.2.2. Conclusiones del experimento.....	115
7.3. Experimento numérico 2. Problemas de prueba. Órdenes de convergencia empíricos.	116
7.3.1. $ EGT $ y orden empírico para el método <i>Runge-Kutta</i> de <i>Gauss</i> .119	
7.3.2. $ EGT $ y orden empírico para los métodos <i>Runge-Kutta-Nyström</i> independientes de y'	121
7.3.3. Conclusiones del experimento.....	124
7.4. Experimento numérico 3. Problema de prueba. Ensayos con <i>splines</i> cúbicos.	125
7.4.1. Experimento con <i>splines</i> cúbicos. Error de redondeo del esquema recursivo. Aritmética de coma flotante.	126
7.4.2. Conclusiones del experimento.....	129
7.5. Experimento numérico 4. Deflexión de una viga biapoyada.	130
7.5.1. Solución numérica. Aplicación del método del disparo lineal.	134
7.5.1.1. Obtención de la solución discreta aproximada del <i>PVF</i> para cada malla de ensayo.	135
7.5.1.2. Estrategia de interpolación con <i>splines</i> cuadráticos.	137
7.5.1.3. Estrategia de interpolación con <i>pseudo-splines</i> cúbicos.....	139
7.5.1.4. Estrategia de interpolación con <i>pseudo-splines</i> quínticos. ..	140
7.5.1.5. Comparativa entre estrategias de interpolación.	140
7.5.2. Conclusiones del experimento.....	142
7.6. Experimento numérico 5. Perfil de temperaturas en una aleta anular hiperbólica.	143
7.6.1. Solución numérica. Aplicación del método del disparo lineal y análisis del error.	146
7.6.1.1. Obtención de la solución discreta aproximada del <i>PVF</i> para cada malla de ensayo.	148
7.6.1.2. $ EGT $ cometido en la determinación de la solución discreta del <i>PVF</i> y su primera derivada.....	149

7.6.1.3. Interpolación de la solución discreta mediante el empleo de pseudo-splines cúbicos. Estudio del error.....	151
7.6.2. Conclusiones del experimento.....	155
7.7. Experimento numérico 6. Respuesta temporal forzada de un péndulo no lineal.....	156
7.7.1. Solución numérica discreta con pares encajados de métodos <i>Runge-Kutta</i> . Estudio del coste computacional.....	159
7.7.1.1. Conclusiones principales sobre el coste computacional. Propuesta de estrategias alternativas de resolución.....	165
7.7.2. Solución numérica continua empleando estrategias de interpolación. Tendencia del error.....	166
7.7.2.1. Estrategia de interpolación con <i>splines</i> cuadráticos.....	169
7.7.2.2. Estrategia de interpolación con <i>pseudo-splines</i> quínticos. Reducción porcentual del coste computacional.....	174
7.7.3. Conclusiones finales del experimento.....	180
7.8. Experimento numérico 7. PVI rígido.....	181
7.8.1. Solución numérica discreta con pares encajados de métodos <i>Runge-Kutta</i> . Estudio del coste computacional.....	183
7.8.2. Solución continua empleando <i>pseudo-splines</i> cúbicos. Monitorización del error y coste computacional.....	187
7.8.3. Conclusiones del experimento.....	190
CAPÍTULO 8. Conclusiones y líneas futuras de investigación.....	193
8.1. Resultados y conclusiones generales.....	195
8.2. Líneas futuras de investigación.....	197
BIBLIOGRAFÍA.....	199
ANEXOS A LA MEMORIA.....	207
ANEXO I. Métodos numéricos discretos de un paso.....	209
ANEXO II. Estrategias de interpolación basadas en <i>splines</i>	223
ANEXO III. Estrategias de interpolación basadas en <i>pseudo-splines</i>	227
ANEXO IV. Códigos de los experimentos numéricos.....	239
Experimento numérico 1. Vibraciones lineales de 1 grado de libertad (sección 7.2 de la memoria).....	239
Experimento numérico 2. Problemas de prueba. Órdenes de convergencia empíricos (sección 7.3 de la memoria).....	242



Experimento numérico 3. Problema de prueba. Ensayos con <i>splines</i> cúbicos (sección 7.4 de la memoria).	246
Experimento numérico 4. Deflexión de una viga biapoyada (sección 7.5 de la memoria).	248
Experimento numérico 5. Perfil de temperaturas en una aleta anular hiperbólica (sección 7.6 de la memoria).....	254
Experimento numérico 6. Respuesta temporal de un péndulo no lineal (sección 7.7 de la memoria).	259
Experimento numérico 7. PVI rígido (sección 7.8 de la memoria).	267
ANEXO V. Funciones auxiliares.....	273



ÍNDICE DE FIGURAS.

Figura 2.1. Resumen de la nomenclatura empleada en métodos numéricos.	16
Figura 3.1. Comparativa entre los métodos de <i>Euler</i> , <i>Heun</i> y <i>Runge-Kutta</i> de orden 4.....	24
Figura 3.2. Secuencia de un par encajado de métodos <i>Runge-Kutta</i>	29
Figura 5.1. Imposición de las condiciones de continuidad y de interpolación con <i>pseudo-splines</i> cúbicos.	79
Figura 5.2. Imposición de las condiciones de continuidad y de interpolación con <i>pseudo-splines</i> quínticos.....	82
Figura 6.1. Método del disparo para problemas lineales. Condiciones de contorno simples.....	93
Figura 7.1. Sistema vibratorio “ <i>mkc</i> ” forzado sometido a una excitación armónica.....	104
Figura 7.2. Gráfica de la solución analítica $x(t)$ del <i>PVI</i> (7.6) y primera derivada $x'(t)$. Interpolación lineal por defecto de <i>MATLAB</i>	108
Figura 7.3. Gráfica del $ EGT $ correspondiente al método de <i>Euler</i> explícito aplicado al <i>PVI</i> (7.6) en el intervalo considerado con paso $h = 0.1$	109
Figura 7.4. Gráfica del $ EGT $ correspondiente a los métodos exclusivamente de orden 3 aplicados al <i>PVI</i> (7.6) en el intervalo considerado con paso uniforme $h = 0.1$	110
Figura 7.5. Gráfica del $ EGT $ correspondiente a los métodos exclusivamente de orden 4 aplicados al <i>PVI</i> (7.6) en el intervalo considerado con paso uniforme $h = 0.1$	111
Figura 7.6. Determinación del orden empírico de los métodos numéricos ensayados aplicados al <i>PVI</i> (7.6).....	114
Figura 7.7. Determinación del orden empírico de los métodos “ <i>HRK3s1o</i> ” y “ <i>RKN3t</i> ” aplicados al <i>PVI</i> (7.6).....	115
Figura 7.8. Soluciones analíticas de los <i>PVIs</i> (7.10) y (7.12) en los nodos de la malla x_3	118
Figura 7.9. $ EGT $ correspondiente al método “ <i>GRK6s1o</i> ” para la malla x_1 tras resolver el <i>PVI</i> (7.10).....	119
Figura 7.10. $ EGT $ correspondiente al método “ <i>GRK6s1o</i> ” para la malla x_2 tras resolver el <i>PVI</i> (7.10).....	119
Figura 7.11. Gráfica del $ EGT $ para el método “ <i>GRK6s1o</i> ” aplicado a la resolución del <i>PVI</i> (7.10) empleando diversas mallas de ensayo.....	120
Figura 7.12. Determinación del orden empírico del método “ <i>GRK6s1o</i> ” aplicado a la resolución del <i>PVI</i> (7.10).....	121
Figura 7.13. Gráfica del $ EGT $ correspondiente al método “ <i>RKN4_no_dy</i> ” aplicado al <i>PVI</i> (7.12) en la malla x_3	122
Figura 7.14. Gráfica del $ EGT $ correspondiente a los métodos <i>RKN</i> de orden 5 para problemas independientes de y' aplicados al <i>PVI</i> (7.12) empleando la malla x_3	122
Figura 7.15. Determinación del orden empírico de los métodos <i>RKN</i> para problemas independientes de y' aplicados al <i>PVI</i> (7.12).....	123



Figura 7.16. Gráfica de la solución analítica $y(x)$ del PVI (7.14) en los nodos de la malla x_4 . Interpolación lineal por defecto de MATLAB	126
Figura 7.17. Gráfica de la solución interpolada en los nodos de la malla x_3 mediante el empleo de splines cúbicos. Solución aproximada del PVI (7.14)	128
Figura 7.18. Gráfica de la solución interpolada en los nodos de la malla x_5 mediante el empleo de splines cúbicos. Solución aproximada del PVI (7.14)	129
Figura 7.19. Viga biapoyada sometida a carga transversal senoidal simétrica $q(x)$ y longitudinal uniforme P	130
Figura 7.20. Diagrama de Sólido Libre de una porción arbitraria de viga. ...	131
Figura 7.21. Deflexión de la viga biapoyada interpolada en los nodos de la malla x_1 mediante aplicación del método del disparo lineal. Solución del PVF (7.26)	137
Figura 7.22. Deflexión de la viga biapoyada interpolada en los nodos de la malla x_2 mediante aplicación del método del disparo lineal. Solución del PVF (7.26)	137
Figura 7.23. Zoom de las Figuras 7.21 y 7.22 en el intervalo $[3,5(m)]$	138
Figura 7.24. Zoom de las Figuras 7.21 y 7.22 en el intervalo $[7,8(m)]$	138
Figura 7.25. Deflexión de la viga biapoyada (7.26) en el intervalo $[3,5] (m)$ mediante método del disparo lineal. Comparativa entre mallas e interpolaciones con <i>pseudo-splines</i> cúbicos.	139
Figura 7.26. Deflexión de la viga biapoyada (7.26) en el intervalo $[3,5] (m)$ mediante método del disparo lineal. Comparativa entre mallas e interpolaciones con <i>pseudo-splines</i> quínticos.	140
Figura 7.27. Deflexión de la viga biapoyada (7.26) en el intervalo $[3.6,4.4] (m)$ mediante método del disparo lineal. Comparativa entre estrategias de interpolación en malla x_1	141
Figura 7.28. Deflexión de la viga biapoyada (7.26) en el intervalo $[3.6,4.4] (m)$ mediante método del disparo lineal. Comparativa entre estrategias de interpolación en malla x_2	142
Figura 7.29. Detalle en corte del tubo rodeado por la aleta anular hiperbólica.	144
Figura 7.30. Forma de la solución exacta del PVF (7.34) dada por (7.35). ...	151
Figura 7.31. Solución aproximada del PVF (7.34) interpolada con un <i>pseudo-spline</i> cúbico en los nodos de la malla R_1	152
Figura 7.32. Gráfica del error cometido respecto de (7.35) al interpolar la solución discreta aproximada en la malla R_1 con “ <i>ps3int1</i> ”	152
Figura 7.33. Solución aproximada del PVF (7.34) interpolada con un <i>pseudo-spline</i> cúbico en los nodos de la malla R_2	153
Figura 7.34. Gráfica del error cometido respecto de (7.35) al interpolar la solución discreta aproximada en la malla R_2 con “ <i>ps3int2</i> ”.	153
Figura 7.35. Solución aproximada del PVF (7.34) interpolada con un <i>pseudo-spline</i> cúbico en los nodos de la malla R_3	154

Figura 7.36. Gráfica del error cometido respecto de (7.35) al interpolar la solución discreta aproximada en la malla R_1 con “ps3int3”.....	154
Figura 7.37. Zoom de la Figura 7.34 en el intervalo [0.7,1].....	155
Figura 7.38. Esquema del péndulo no lineal del que se desea obtener su respuesta temporal.....	157
Figura 7.39. Instante $t = 0$ en el que comienzan las oscilaciones del péndulo.....	158
Figura 7.40. Solución numérica discreta del PVI (7.45) mediante solvers de la Tabla 3.3. Tolerancias por defecto de MATLAB para el cómputo del error en cada paso.	160
Figura 7.41. Zoom de la Figura 7.46 en el intervalo [12,18] (s).....	161
Figura 7.42. Solución numérica discreta del PVI (7.45) mediante solvers de la Tabla 3.3. Tolerancias mejoradas para el cómputo del error en cada paso.	161
Figura 7.43. Zoom de la Figura 7.42 en el intervalo [25,30] (s).....	162
Figura 7.44. Evolución global del paso h_k para cada una de las mallas auto computadas con los solvers empleados para aproximar soluciones discretas del PVI (7.45).....	163
Figura 7.45. Evolución del error cometido por los solvers empleados respecto de la solución discreta aproximada de (7.45) dada por “ode89” en los nodos de la malla “t89m”.....	164
Figura 7.46. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_1 por el spline cuadrático “s2int1”.....	169
Figura 7.47. Error de interpolación cometido por el spline cuadrático “s2int1” respecto de la solución discreta muy densificada a evitar computada con “ode89”.....	170
Figura 7.48. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_2 por el spline cuadrático “s2int2”.....	170
Figura 7.49. Error de interpolación cometido por el spline cuadrático “s2int2” respecto de la solución discreta muy densificada a evitar computada con “ode89”.....	171
Figura 7.50. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_3 por el spline cuadrático “s2int3”.....	171
Figura 7.51. Error de interpolación cometido por el spline cuadrático “s2int3” respecto de la solución discreta muy densificada a evitar computada con “ode89”.....	172
Figura 7.52. Zoom de la Figura 7.50 en [20.5,22.5] (s).....	173
Figura 7.53. Zoom de la Figura 7.50 en [25,30] (s).....	173
Figura 7.54. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_1 por el pseudo-spline quintico “ps5int1”.....	175
Figura 7.55. Zooms de la Figura 7.54.....	175
Figura 7.56. Error de interpolación cometido por el pseudo-spline quintico “ps5int1” respecto de la solución discreta muy densificada a evitar computada con “ode89”.....	176
Figura 7.57. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_2 por el pseudo-spline quintico “ps5int2”.....	176



Figura 7.58. Zoom de la **Figura 7.57** en [23,25] (s). 177

Figura 7.59. Error de interpolación cometido por el *pseudo-spline* quíntico “ps5int2” respecto de la solución discreta muy densificada a evitar computada con “ode89” 177

Figura 7.60. Solución continua del **PVI (7.45)** interpolada en los nodos de la malla t_3 por el *pseudo-spline* quíntico “ps5int3” 178

Figura 7.61. Zoom de la **Figura 7.60** en [18,21] (s). 178

Figura 7.62. Error de interpolación cometido por el *pseudo-spline* quíntico “ps5int3” respecto de la solución discreta muy densificada a evitar computada con “ode89” 179

Figura 7.63. Solución exacta (7.49) del **PVI (7.48)**. 182

Figura 7.64. Primera derivada (7.50) de (7.49). 182

Figura 7.65. Zoom de la **Figura 7.64** en [0,0.02] 183

Figura 7.66. Solución discreta aproximada del **PVI (7.48)** en los nodos de la malla auto computada “x23s” mediante el empleo de “ode23s” 184

Figura 7.67. Zoom de la **Figura 7.66** en [0,0.01] 184

Figura 7.68. |EGT| cometido por “ode23s” en la aproximación discreta de soluciones del **PVI (7.48)**. 185

Figura 7.69. Zoom de la **Figura 7.68** en [0,0.01] 185

Figura 7.70. Evolución del paso “h23s” correspondiente a la malla “x23s”. 186

Figura 7.71. Zoom de la **Figura 7.70**. 186

Figura 7.72. Solución continua aproximada del **PVI (7.48)** empleando el trazador “ps3int” y la malla “xint”. 188

Figura 7.73. Zoom de la **Figura 7.72** en [0,0.01] 188

Figura 7.74. Error de interpolación correspondiente al trazador “ps3int” respecto de la solución exacta del **PVI (7.48)** dada por (7.49). 189

Figura 7.75. Zoom de la **Figura 7.74** en [0,0.005]. 189

Figura 7.76. Zoom de la **Figura 7.74** en [0.005,0.01]. 190

ÍNDICE DE TABLAS.

Tabla 3.1. Órdenes de convergencia a esperar de los métodos numéricos <i>Runge-Kutta</i> .	22
Tabla 3.2. Métodos <i>RK</i> para la resolución del <i>PVI</i> de 2º orden.	28
Tabla 3.3. <i>Solvers</i> de <i>MATLAB</i> para problemas no rígidos que implementan métodos <i>Runge-Kutta</i> .	30
Tabla 3.4. <i>Solvers</i> de <i>MATLAB</i> para problemas rígidos que implementan métodos <i>Runge-Kutta</i> .	30
Tabla 3.5. Métodos <i>RKN</i> explícitos para el <i>PVI</i> completo de la forma $y''(x) = f(x, y(x), y'(x))$.	34
Tabla 3.6. Métodos <i>RKN</i> explícitos para el <i>PVI</i> de la forma $y''(x) = f(x, y(x))$.	36
Tabla 4.1. Resumen de condiciones a imponer para determinar unívocamente un <i>spline</i> interpolador.	42
Tabla 4.2. Resumen de condiciones a imponer para determinar unívocamente un <i>spline</i> cuadrático interpolador.	44
Tabla 4.3. Cuadro sinóptico resumen. <i>Splines</i> cuadráticos interpoladores.	46
Tabla 4.4. Resumen de condiciones a imponer para determinar unívocamente un <i>spline</i> cúbico interpolador.	48
Tabla 4.5. Cuadro sinóptico resumen. <i>Splines</i> cúbicos interpoladores.	50
Tabla 5.1. Estructura del Capítulo 5 . <i>Pseudo-splines</i> interpoladores.	56
Tabla 5.2. Resumen de condiciones a imponer para determinar unívocamente un <i>pseudo-spline</i> cuártico interpolador.	57
Tabla 5.3. Cuadro sinóptico resumen. <i>Pseudo-splines</i> cuárticos interpoladores.	71
Tabla 5.4. Resumen de la implementación en <i>MATLAB</i> de estrategias de interpolación basadas en <i>pseudo-splines</i> cuárticos.	72
Tabla 5.5. Particularizaciones relevantes de los polinomios osculadores.	74
Tabla 5.6. Resumen de condiciones a imponer para determinar unívocamente un <i>pseudo-spline</i> cúbico interpolador.	75
Tabla 5.7. Esquema de diferencias divididas con argumentos repetidos. Interpolación de <i>Hermite</i> .	77
Tabla 5.8. Diferencias divididas para el cálculo de S_k mediante el polinomio interpolador de <i>Hermite</i> .	78
Tabla 5.9. Resumen de condiciones a imponer para determinar unívocamente un <i>pseudo-spline</i> quíntico interpolador.	80
Tabla 5.10. Esquema de diferencias divididas con argumentos repetidos para calcular S_k . <i>Pseudo-splines</i> quínticos interpoladores.	81
Tabla 5.11. Expresiones desarrolladas de las diferencias divididas presentadas en la Tabla 5.10 .	82
Tabla 5.12. Resumen de la implementación en <i>MATLAB</i> de estrategias de interpolación basadas en polinomios osculadores (<i>pseudo-splines</i> cúbicos y quínticos).	83



Tabla 6.1. Algoritmo computacional del método del disparo lineal con condiciones de contorno simples.....	94
Tabla 7.1. Resumen de los experimentos numéricos planteados. Capítulo 7	103
Tabla 7.2. Valores numéricos considerados para particularizar el PVI surgido de la EDO (7.4)	106
Tabla 7.3. Valores numéricos de las agrupaciones consideradas para particularizar el PVI surgido de la EDO (7.4)	106
Tabla 7.4. Mallas equiespaciadas progresivamente densificadas en el intervalo $[0,3]$ (s).....	107
Tabla 7.5. Valores numéricos de la solución analítica $x(t)$ de (7.6) y su primera derivada $x'(t)$ con paso uniforme $h = 0.1$	108
Tabla 7.6. Métodos numéricos para la obtención de soluciones discretas aproximadas del PVI (7.6) empleados con su alias de MATLAB	109
Tabla 7.7. Valores numéricos del EGT en los nodos de la malla 1 ($h = 0.1$) correspondiente al método de <i>Euler</i> explícito aplicado al PVI (7.6)	110
Tabla 7.8. Valores numéricos del EGT en los nodos de la malla 1 ($h = 0.1$) correspondientes a los métodos “ <i>HRK3s1o</i> ”, “ <i>RKN3</i> ” y “ <i>RKN3t</i> ” aplicados al PVI (7.6)	111
Tabla 7.9. Valores numéricos del EGT en los nodos de la malla 1 ($h = 0.1$) correspondientes a los métodos “ <i>RK4s1o</i> ” y “ <i>RKN4</i> ” aplicados al PVI (7.6)	112
Tabla 7.10. Valores numéricos del EGT para cada malla de ensayo y correspondientes a cada método ensayado en el nodo extremo $t = 3$ (s) del PVI (7.6)	113
Tabla 7.11. Determinación del orden empírico del método de <i>Euler</i> explícito. EGT en nodo extremo $t = 3$ (s).....	113
Tabla 7.12. Determinación del orden empírico de los métodos ensayados exclusivamente de orden 3 aplicados al PVI (7.6) . EGT en nodo extremo $t = 3$ (s).....	114
Tabla 7.13. Determinación del orden empírico de los métodos ensayados exclusivamente de orden 4 aplicados al PVI (7.6) . EGT en nodo extremo $t = 3$ (s).....	114
Tabla 7.14. Errores de los órdenes empíricos respecto de los teóricos correspondientes a los métodos numéricos ensayados y aplicados al PVI (7.6)	115
Tabla 7.15. Métodos numéricos para la obtención de soluciones discretas aproximadas de los PVIs (7.10) y (7.12) empleados con su alias de MATLAB	117
Tabla 7.16. Mallas equiespaciadas progresivamente densificadas en el intervalo $[0,2]$	117
Tabla 7.17. Valores numéricos de las soluciones de los PVIs (7.10) y (7.12) en los nodos de la malla x_3	118
Tabla 7.18. Determinación del orden empírico del método “ <i>GRK6s1o</i> ”. EGT máximo para cada malla de ensayo.	121

Tabla 7.19. Determinación del orden empírico de los métodos <i>RKN</i> ensayados para problemas independientes de y' aplicados al <i>PVI (7.12)</i> . $ EGT $ en nodo extremo $x = 2$	123
Tabla 7.20. Errores de los órdenes empíricos respecto de los teóricos correspondientes a los métodos <i>RKN</i> ensayados para problemas independientes de y' y aplicados al <i>PVI (7.12)</i>	124
Tabla 7.21. Mallas equiespaciadas en el intervalo $[0,1]$	125
Tabla 7.22. Valores del coeficiente a_{n3} para cada una de las mallas de ensayo y solución exacta en nodos del último intervalo generado $[t_{n-1}, t_n]$	127
Tabla 7.23. Valores numéricos para particularizar la viga biapoyada.....	133
Tabla 7.24. Mallas no equiespaciadas en el intervalo $[0,8]$ (m).....	134
Tabla 7.25. Valores numéricos de las soluciones discretas aproximadas en la malla x_1 de los <i>PVIs (7.27)</i> y <i>(7.28)</i> posteriormente combinadas por disparo lineal para obtener la solución del <i>PVF (7.26)</i>	135
Tabla 7.26. Valores numéricos de las soluciones discretas aproximadas en la malla x_2 de los <i>PVIs (7.27)</i> y <i>(7.28)</i> posteriormente combinadas por disparo lineal para obtener la solución del <i>PVF (7.26)</i>	136
Tabla 7.27. Cuadro sinóptico resumen de estrategias de interpolación adoptadas para el <i>PVF (7.26)</i>	136
Tabla 7.28. Valores numéricos nodales de la deflexión de la viga biapoyada <i>(7.26)</i> y error cometido respecto a la estrategia de interpolación con <i>pseudo-splines</i> quínticos en malla x_1	141
Tabla 7.29. Valores numéricos nodales de la deflexión de la viga biapoyada <i>(7.26)</i> y error cometido respecto a la estrategia de interpolación con <i>pseudo-splines</i> quínticos en malla x_2	142
Tabla 7.30. Significado físico de los términos de la ecuación general de las aletas.	144
Tabla 7.31. Valores numéricos propuestos para la aleta anular.	146
Tabla 7.32. Mallas para abordar la resolución del <i>PVF (7.34)</i>	147
Tabla 7.33. Valores numéricos de las soluciones discretas aproximadas en la malla R_1 de los <i>PVIs (7.37)</i> y <i>(7.38)</i> posteriormente combinadas por disparo lineal para obtener la solución del <i>PVF (7.34)</i>	148
Tabla 7.34. Valores numéricos de las soluciones discretas aproximadas en la malla R_2 de los <i>PVIs (7.37)</i> y <i>(7.38)</i> posteriormente combinadas por disparo lineal para obtener la solución del <i>PVF (7.34)</i>	148
Tabla 7.35. Valores numéricos de las soluciones discretas aproximadas en la malla R_3 de los <i>PVIs (7.37)</i> y <i>(7.38)</i> posteriormente combinadas por disparo lineal para obtener la solución del <i>PVF (7.34)</i>	148
Tabla 7.36. $ EGTs $ proporcionados por el método “ <i>RKN51_no_dy</i> ” aplicado a la resolución del <i>PVF (7.34)</i> por disparo lineal en la malla R_1	150
Tabla 7.37. $ EGTs $ proporcionados por el método “ <i>RKN51_no_dy</i> ” aplicado a la resolución del <i>PVF (7.34)</i> por disparo lineal en la malla R_2	150
Tabla 7.38. $ EGTs $ proporcionados por el método “ <i>RKN51_no_dy</i> ” aplicado a la resolución del <i>PVF (7.34)</i> por disparo lineal en la malla R_3	150



Tabla 7.39. Valores numéricos considerados para particularizar el *PVI (7.43)*. 159

Tabla 7.40. Estadísticas de las mallas auto computadas por los solvers empleados para aproximar soluciones discretas del *PVI (7.45)*. 162

Tabla 7.41. Errores máximos cometidos por los solvers “ode23”, “ode45” y “ode78” respecto a la solución discreta aproximada de (7.45) dada por “ode89” en los nodos de la malla “t89m” 164

Tabla 7.42. Mallas equiespaciadas y poco densificadas para el cómputo de soluciones discretas aproximadas del *PVI (7.45)* mediante “ode89” 166

Tabla 7.43. Valores numéricos de la solución discreta aproximada del *PVI (7.45)* proporcionada por “ode89” en los nodos de la malla t_1 167

Tabla 7.44. Valores numéricos de la solución discreta aproximada del *PVI (7.45)* proporcionada por “ode89” en los nodos de la malla t_2 168

Tabla 7.45. Valores numéricos de la solución discreta aproximada del *PVI (7.45)* proporcionada por “ode89” en los nodos de la malla t_3 168

Tabla 7.46. Cuadro sinóptico resumen de *splines* cuadráticos creados para aproximar soluciones continuas del *PVI (7.45)*. 169

Tabla 7.47. Cuadro sinóptico resumen de *pseudo-splines* quínticos creados para aproximar soluciones continuas del *PVI (7.45)*. 175

Tabla 7.48. Evidencia de la reducción del coste computacional al emplear estrategias de interpolación para resolver de forma continua el *PVI (7.45)*. 179

Tabla 7.49. Estadísticas principales de la malla “x23s” auto computada por “ode23s”. 186

Tabla 7.50. Malla equiespaciada a trozos “xint” que discretiza el dominio continuo del *PVI (7.48)*. 187

Tabla 7.51. Estadísticas principales de la malla “xint” y ahorro de coste computacional respecto a la malla auto computada “x23s” y estrategia única discreta empleando “ode23s”. 190



ÍNDICE DE ECUACIONES.

Ecuación (2.1)	11
Ecuación (2.2)	11
Ecuación (2.3)	11
Ecuación (2.4)	12
Ecuación (2.5)	13
Ecuación (2.6)	13
Ecuación (2.7)	13
Ecuación (2.8)	14
Ecuación (2.9)	14
Ecuación (2.10)	14
Ecuación (2.11)	15
Ecuación (2.12)	15
Ecuación (2.13)	15
Ecuación (2.14)	16
Ecuación (2.15)	16
Ecuación (2.16)	16
Ecuación (3.1)	20
Ecuación (3.2)	20
Ecuación (3.3)	20
Ecuación (3.4)	21
Ecuación (3.5)	21
Ecuación (3.6)	21
Ecuación (3.7)	21
Ecuación (3.8)	22
Ecuación (3.9)	23
Ecuación (3.10)	23
Ecuación (3.11)	24
Ecuación (3.12)	25
Ecuación (3.13)	25
Ecuación (3.14)	25
Ecuación (3.15)	25
Ecuación (3.16)	26
Ecuación (3.17)	26
Ecuación (3.18)	26
Ecuación (3.19)	27
Ecuación (3.20)	27
Ecuación (3.21)	27
Ecuación (3.22)	28
Ecuación (3.23)	29
Ecuación (3.24)	31
Ecuación (3.25)	31
Ecuación (3.26)	31
Ecuación (3.27)	32



Ecuación (3.28)	32
Ecuación (3.29)	32
Ecuación (3.30)	33
Ecuación (3.31)	33
Ecuación (3.32)	33
Ecuación (3.33)	33
Ecuación (3.34)	34
Ecuación (3.35)	34
Ecuación (3.36)	35
Ecuación (3.37)	35
Ecuación (3.38)	35
Ecuación (3.39)	36
Ecuación (3.40)	36
Ecuación (3.41)	37
Ecuación (3.42)	37
Ecuación (3.43)	37
Ecuación (3.44)	38
Ecuación (3.45)	38
Ecuación (4.1).....	42
Ecuación (4.2).....	42
Ecuación (4.3).....	43
Ecuación (4.4).....	43
Ecuación (4.5).....	43
Ecuación (4.6).....	44
Ecuación (4.7).....	44
Ecuación (4.8).....	45
Ecuación (4.9).....	45
Ecuación (4.10)	45
Ecuación (4.11)	45
Ecuación (4.12)	46
Ecuación (4.13)	46
Ecuación (4.14)	46
Ecuación (4.15)	47
Ecuación (4.16)	47
Ecuación (4.17)	47
Ecuación (4.18)	47
Ecuación (4.19)	48
Ecuación (4.20)	49
Ecuación (4.21)	49
Ecuación (4.22)	49
Ecuación (4.23)	49
Ecuación (4.24)	50
Ecuación (4.25)	50
Ecuación (4.26)	50
Ecuación (5.1).....	55



Ecuación (5.2)	56
Ecuación (5.3)	56
Ecuación (5.4)	57
Ecuación (5.5)	58
Ecuación (5.6)	58
Ecuación (5.7)	58
Ecuación (5.8)	58
Ecuación (5.9)	58
Ecuación (5.10)	58
Ecuación (5.11)	59
Ecuación (5.12)	59
Ecuación (5.13)	59
Ecuación (5.14)	60
Ecuación (5.15)	60
Ecuación (5.16)	60
Ecuación (5.17)	60
Ecuación (5.18)	61
Ecuación (5.19)	61
Ecuación (5.20)	61
Ecuación (5.21)	61
Ecuación (5.22)	62
Ecuación (5.23)	62
Ecuación (5.24)	62
Ecuación (5.25)	62
Ecuación (5.26)	63
Ecuación (5.27)	63
Ecuación (5.28)	63
Ecuación (5.29)	64
Ecuación (5.30)	64
Ecuación (5.31)	64
Ecuación (5.32)	64
Ecuación (5.33)	65
Ecuación (5.34)	65
Ecuación (5.35)	66
Ecuación (5.36)	66
Ecuación (5.37)	66
Ecuación (5.38)	66
Ecuación (5.39)	67
Ecuación (5.40)	67
Ecuación (5.41)	67
Ecuación (5.42)	68
Ecuación (5.43)	68
Ecuación (5.44)	68
Ecuación (5.45)	69
Ecuación (5.46)	69



Ecuación (5.47)	69
Ecuación (5.48)	69
Ecuación (5.49)	70
Ecuación (5.50)	70
Ecuación (5.51)	70
Ecuación (5.52)	71
Ecuación (5.53)	73
Ecuación (5.54)	73
Ecuación (5.55)	73
Ecuación (5.56)	74
Ecuación (5.57)	75
Ecuación (5.58)	75
Ecuación (5.59)	76
Ecuación (5.60)	76
Ecuación (5.61)	76
Ecuación (5.62)	77
Ecuación (5.63)	77
Ecuación (5.64)	78
Ecuación (5.65)	78
Ecuación (5.66)	79
Ecuación (5.67)	79
Ecuación (5.68)	81
Ecuación (5.69)	81
Ecuación (6.1).....	88
Ecuación (6.2).....	88
Ecuación (6.3).....	88
Ecuación (6.4).....	89
Ecuación (6.5).....	89
Ecuación (6.6).....	89
Ecuación (6.7).....	90
Ecuación (6.8).....	90
Ecuación (6.9).....	90
Ecuación (6.10)	90
Ecuación (6.11)	90
Ecuación (6.12)	90
Ecuación (6.13)	91
Ecuación (6.14)	91
Ecuación (6.15)	92
Ecuación (6.16)	92
Ecuación (6.17)	92
Ecuación (6.18)	92
Ecuación (6.19)	92
Ecuación (6.20)	92
Ecuación (6.21)	93
Ecuación (6.22)	95



Ecuación (6.23)	95
Ecuación (6.24)	95
Ecuación (6.25)	96
Ecuación (6.26)	96
Ecuación (6.27)	96
Ecuación (6.28)	96
Ecuación (6.29)	96
Ecuación (6.30)	97
Ecuación (6.31)	97
Ecuación (6.32)	97
Ecuación (6.33)	98
Ecuación (6.34)	98
Ecuación (6.35)	98
Ecuación (6.36)	98
Ecuación (6.37)	98
Ecuación (6.38)	99
Ecuación (6.39)	99
Ecuación (6.40)	99
Ecuación (7.1)	105
Ecuación (7.2)	105
Ecuación (7.3)	105
Ecuación (7.4)	105
Ecuación (7.5)	105
Ecuación (7.6)	106
Ecuación (7.7)	107
Ecuación (7.8)	107
Ecuación (7.9)	113
Ecuación (7.10)	116
Ecuación (7.11)	116
Ecuación (7.12)	117
Ecuación (7.13)	117
Ecuación (7.14)	125
Ecuación (7.15)	125
Ecuación (7.16)	126
Ecuación (7.17)	127
Ecuación (7.18)	131
Ecuación (7.19)	132
Ecuación (7.20)	132
Ecuación (7.21)	132
Ecuación (7.22)	132
Ecuación (7.23)	132
Ecuación (7.24)	133
Ecuación (7.25)	133
Ecuación (7.26)	133
Ecuación (7.27)	134



Ecuación (7.28)	135
Ecuación (7.29)	143
Ecuación (7.30)	145
Ecuación (7.31)	145
Ecuación (7.32)	145
Ecuación (7.33)	145
Ecuación (7.34)	146
Ecuación (7.35)	146
Ecuación (7.36)	147
Ecuación (7.37)	147
Ecuación (7.38)	147
Ecuación (7.39)	148
Ecuación (7.40)	149
Ecuación (7.41)	149
Ecuación (7.42)	150
Ecuación (7.43)	157
Ecuación (7.44)	158
Ecuación (7.45)	159
Ecuación (7.46)	160
Ecuación (7.47)	161
Ecuación (7.48)	181
Ecuación (7.49)	181
Ecuación (7.50)	182



CAPÍTULO 1. Introducción, objetivos y estructura del trabajo.



1.1. Introducción y justificación.

Un profesional de la Ingeniería ha de conocer y ser capaz de aplicar estrategias de resolución de problemas múltiples y de distintas índoles que le conduzcan a la consecución de los objetivos marcados en las especificaciones de un proyecto, teniendo la obligación de documentar certeramente los aspectos y pilares fundamentales en los que se sustentan las decisiones profesionales adoptadas. **Las matemáticas juegan un papel determinante en este aspecto**, pues resultan ser una herramienta con frecuencia imprescindible y capaz de trasladar a un marco objetivo todo lo anterior, permitiendo la toma de decisiones posterior.

Como particularidad a lo expuesto, actualmente son infinitud los fenómenos físicos que pueden ser descritos mediante **ecuaciones diferenciales**. A lo largo de la historia, el ser humano ha cultivado un interés especial en este tipo de ecuaciones debido a su inmensa variedad y utilidad [1-3]. Por ello, y **en especial atención a las competencias básicas que ha de tener un ingeniero**, surge la necesidad de conocer en profundidad las metodologías existentes que conducen a la resolución de problemas formulados en términos de las mismas, partiendo de los marcos teóricos matemáticos disponibles y en continuo desarrollo e investigación, tanto por profesionales de las ciencias exactas, como por personas cuyos campos de estudio se nutren directamente de ellas.

El descubrimiento e invención de las ecuaciones diferenciales se remonta a la segunda mitad del siglo XVII [4], cuando el físico y matemático británico *Sir Isaac Newton* (1643 – 1727) y el matemático alemán *Gottfried Leibniz* (1646 – 1716) desarrollaron estudios independientes e inicialmente sin publicaciones oficiales (en el marco del desarrollo del cálculo diferencial) en los que proponían diversos tipos de nuevas ecuaciones que involucraban a su función solución y, al menos, a alguna de sus derivadas. Con el paso de los años, dichos estudios preliminares desembocaron en publicaciones científicas oficiales que datan de 1693 (además del surgimiento del cálculo integral), en donde las ecuaciones inicialmente planteadas ya fueron propuestas y ensayadas para su resolución, sentando las bases de un **nuevo campo matemático de estudio inmenso con infinitud de aplicaciones en otras ciencias que ha motivado el desarrollo de este trabajo**. Otros nombres de célebres matemáticos y científicos que han colaborado enormemente a enriquecer este área del conocimiento son *Leonhard Euler* (1707 – 1783), la familia *Bernoulli*, *Joseph-Louis Lagrange* (1736 – 1813) y *Pierre-Simon Laplace* (1749 – 1827), entre muchos ilustres no mencionados aquí.

No es para nada descabellado afirmar que **la naturaleza puede ser descrita mediante ecuaciones diferenciales** [5]. Por ejemplo, la descripción del movimiento de los cuerpos existentes en todo el universo bebe de este campo de estudio, al mismo tiempo que la electrodinámica también se fundamenta en estas ecuaciones, la mecánica clásica (oscilaciones, campos de deformación...)



y cuántica, los fenómenos difusivos y la termodinámica... Como puede apreciarse, todas estas áreas de conocimiento (entre muchísimas otras) resultan ser de especial interés para la labor llevada a cabo por un ingeniero, cuyo objetivo es **evolucionar hacia un mundo con nuevos paradigmas tecnológicos que permitan mejorar la calidad de vida de las personas**, buscando siempre derribar fronteras del conocimiento responsablemente mediante el uso práctico de las ciencias teóricas madre.

Una de las cuestiones más importantes surgidas tras la elaboración de modelos matemáticos aplicados a la Ingeniería y formulados en términos de ecuaciones diferenciales es la **disponibilidad de métodos lo suficientemente precisos y sistematizables para que puedan ser aplicados a la resolución de dichos problemas eficientemente**. Esto no es para nada sencillo, pues las no linealidades que afloran en muchos de ellos y la frecuente imposibilidad de encontrar soluciones analíticas exactas mediante funciones elementales depositan toda la confianza para su resolución en el **análisis numérico**, otro campo aplicado de las matemáticas de vital importancia en Ingeniería que surge principalmente de la computación y las aproximaciones a las soluciones de los problemas citados.

1.2. Objetivos del TFG.

En atención a todas las consideraciones de **1.1** se ha desarrollado el presente Trabajo Fin de Grado, sirviendo su consecución para avanzar en la investigación relativa al empleo de métodos numéricos y estrategias de resolución de problemas formulados en términos de **Ecuaciones Diferenciales Ordinarias exclusivamente (en adelante abreviadas como EDOs) de 2º orden**, con aplicación directa a la Ingeniería Industrial.

De todo el espectro de ecuaciones diferenciales y, más en concreto, de las **EDOs**, se eligen las de 2º orden porque actualmente se continúa su estudio mediante numerosos esfuerzos investigadores (algunos ejemplos actuales se pueden consultar en [6-10]) y los problemas formulados en términos de las mismas son más difíciles de abordar que aquellos que se definen con **EDOs** de 1^{er} orden, ya que para estas últimas se dispone de muchos más métodos.

Este trabajo concierne al desarrollo y análisis de diversas estrategias de resolución basadas en la utilización de **métodos numéricos para la obtención de soluciones discretas aproximadas y su posterior interpolación con vistas a obtener aproximaciones continuas a las soluciones buscadas**, dejando a un lado las simplificaciones y estudios que han de realizarse cuando el abordaje de las **EDOs** se decide hacer desde un punto de vista más teórico. Por tanto, en concordancia con estas bases, se pueden plantear los siguientes objetivos:

- i. Presentar un marco teórico-práctico concerniente al empleo de métodos numéricos para la obtención de soluciones discretas aproximadas de **Problemas de Valores Iniciales (PVIs)** formulados en términos de **EDOs**

- exclusivamente de 2º orden, cuya principal aportación sea el conocimiento ingenieril de dichos métodos sin perder su fundamento matemático. **Se fija el interés en los métodos Runge-Kutta (RK) generalizados a problemas de 2º orden y en los métodos Runge-Kutta-Nyström (RKN), ambas familias de métodos de un paso.**
- ii. Experimentar con dichos métodos para justificar empíricamente su orden de convergencia mediante experimentos numéricos apropiados y enjuiciar su idoneidad para ser aplicados a problemas particularizados de distinta índole, dentro del espectro abarcado por este trabajo.
 - iii. Presentar un marco de actuación para la resolución de **Problemas de Valores en la Frontera (PVFs)** basado en la aplicación del método del disparo para **problemas exclusivamente lineales**, aprovechando los conocimientos y métodos de resolución discretos para **PVI** de los que se dispone.
 - iv. **Dar una mayor realidad, tanto física como ingenieril**, a las aproximaciones discretas anteriormente citadas mediante el empleo de **estrategias de interpolación** para problemas lineales y no lineales basadas en trazadores polinomiales (**splines** y **pseudo-splines**) especialmente diseñadas para abordar la resolución de **PVIs (y de PVFs sólo lineales por disparo)** de 2º orden, que permitan relajar el esfuerzo computacional necesario respecto a una estrategia única empleando métodos **RK** o **RKN**.
 - v. Aplicar las simplificaciones disponibles y surgidas de los desarrollos matemáticos efectuados más adecuadas para **evitar las mayores no linealidades posibles a la hora de diseñar las estrategias de interpolación citadas**, permitiendo de ese modo un marco teórico más compacto y simplificado, pero manteniendo la naturaleza de los problemas que se deseen resolver.
 - vi. **Adoptar un marco de actuación fundamentado en los métodos discretos y estrategias de interpolación aplicándolos simultáneamente** a problemas relacionados con la Ingeniería Industrial.
 - vii. **Avanzar en el conocimiento de un software de cálculo numérico para ingenieros**, en donde se vuelquen todos los métodos seleccionados y estrategias de interpolación diseñadas para la resolución de los problemas planteados. Se fija el software **MATLAB (desarrollado por la compañía MathWorks®)** como el empleado para poder llevar a cabo todos los programas.

En atención al último objetivo fijado para la realización de este trabajo, se elige el software **MATLAB** dada su idoneidad específica para el cálculo numérico. Este software surgió en los años 60 de la mano del profesor de la *Universidad de Nuevo México* **Dr. Cleve Moler [11]** y su denominación tiene origen en "**MATrix LABoratory** (laboratorio matricial)". Actualmente, es un **potentísimo paquete de cálculo** que goza de muy buena precisión, un lenguaje de alto nivel que facilita la escritura de los códigos, amplio soporte de funciones ya



desarrolladas e implementadas eficazmente y un número de usuarios a nivel mundial muy extenso, sobre todo en el ámbito de la Ingeniería.

1.3. Estructura de la memoria.

Expuestos ya en **1.2** los objetivos del *TFG*, se indica a continuación la estructura de la memoria que ha sido considerada más idónea para el seguimiento adecuado de las líneas de investigación y actuación que se plantean. Se incluye una descripción lo más concisa posible de cada capítulo (comenzando por el **Capítulo 2**), pero que lo identifica claramente, invitando al lector a consultarlo con detenimiento.

- **Capítulo 2. Problemas de Valores Iniciales de 2º orden. Nomenclatura en métodos numéricos.**

En este capítulo se realiza una introducción formal a los *PVIs* de 2º orden. Además, se indica la nomenclatura elegida para desarrollar los métodos numéricos que en este trabajo se plantean.

- **Capítulo 3. Métodos numéricos discretos de un paso.**

En concordancia con lo expuesto en la sección **1.2** de objetivos, se plantean los esquemas para la obtención de soluciones discretas aproximadas de los *PVIs* de 2º orden basados en métodos numéricos *Runge-Kutta* y *Runge-Kutta-Nyström*, ambos de un paso.

- **Capítulo 4. Estrategias de interpolación de soluciones de *PVIs* empleando *splines*.**

Se trata del primer capítulo en el que se desarrollan las estrategias de interpolación de las soluciones discretas obtenidas en el **Capítulo 3** que contiene este trabajo. En concreto, se desarrollan estrategias basadas exclusivamente en trazadores polinomiales al uso, denominados *splines* (cuadráticos y cúbicos).

- **Capítulo 5. Estrategias de interpolación de soluciones de *PVIs* empleando *pseudo-splines*.**

Como continuación al marco teórico de estrategias expuesto en el **Capítulo 4**, se profundiza en el empleo de trazadores polinomiales para resolver *PVIs* de 2º orden de los diversos tipos expuestos en el **Capítulo 1**, ahora empleando *pseudo-splines* (cúbicos, cuárticos y quínticos) con multitud de variantes abordadas y particularizaciones a problemas incompletos, lineales, no lineales... Estas aportaciones pretenden mejorar a las estrategias de interpolación que emplean *splines*.

- **Capítulo 6. Método del disparo para Problemas de Valores en la Frontera lineales.**

Una vez expuesto todo lo relativo a los *PVIs*, se indican los fundamentos de los Problemas de Valores en la Frontera o *PVFs* de 2º orden exclusivamente lineales, teniendo en cuenta distintas tipologías de condiciones de contorno, las cuales desembocan en una metodología de resolución aplicando el método del disparo particularizada a cada

tipo de *PVF* abordado, teniendo su base en la resolución por separado de ciertos *PVIs* empleando las estrategias expuestas en los **Capítulos 3, 4 y 5**.

- **Capítulo 7. Experimentos numéricos.**

Este capítulo incluye todos los experimentos numéricos realizados (la mayoría de ellos contextualizados con fenómenos de estudio en Ingeniería) para probar la estrategia dual de resolución de problemas que nace de la aplicación simultánea de los métodos numéricos del **Capítulo 3**, junto con las estrategias de interpolación de soluciones indicadas en los **Capítulos 4 y 5**. Su principal función es dar al trabajo el carácter práctico fijado en los objetivos de **1.2**, y enjuiciar la viabilidad de todas las estrategias planteadas anteriormente.

- **Capítulo 8. Conclusiones y líneas futuras de investigación.**

Se concluye la memoria del trabajo indicando las conclusiones obtenidas tras la experimentación y desarrollo del marco teórico, verificando la consecución de los objetivos y planteando futuras líneas de investigación que se consideran adecuadas para el correcto seguimiento de los estudios planteados.

- **Bibliografía.**

En esta sección se indican todas las referencias bibliográficas que han sido consultadas para la realización del trabajo y citadas entre líneas del mismo. La asignación de un identificador numérico a cada referencia se realiza empleando el mismo orden de aparición en el texto.

Adicionalmente, se incluyen al final de este documento los siguientes **Anexos**:

- **ANEXO I.** En él se incluyen todos los métodos numéricos *Runge-Kutta* y *Runge-Kutta-Nyström* expuestos en el **Capítulo 3** implementados mediante funciones en **MATLAB**.
- **ANEXO II.** Este anexo contiene todas las funciones definidas en **MATLAB** que implementan las estrategias de interpolación descritas en el **Capítulo 4** y que emplean *splines*.
- **ANEXO III.** Con la misma estructura que el anexo anterior, incluye las implementaciones de las estrategias de interpolación de las soluciones de *PVIs* de 2º orden empleando *pseudo-splines*, en base al marco teórico expuesto en el **Capítulo 5**.
- **ANEXO IV.** Se incluyen todos los códigos de **MATLAB** escritos que computan, mediante la ayuda de las funciones definidas en todos los anexos anteriores, los experimentos numéricos del **Capítulo 7**.
- **ANEXO V.** Contiene funciones auxiliares que han sido necesarias para dar mayor robustez a los códigos de todos los anexos.





CAPÍTULO 2. Problemas de Valores Iniciales de 2^o orden. Nomenclatura en métodos numéricos.



2.1. Ecuaciones diferenciales ordinarias de 2º orden.

Las ecuaciones diferenciales ordinarias (**EDOs**) de 2º orden son ecuaciones que involucran a una única variable independiente (por ejemplo, x), a la solución de la ecuación $y(x)$ y tanto a su primera ($y'(x)$) como a su segunda derivada ($y''(x)$). Este trabajo considera como punto de partida la **EDO** de 2º orden **expresada en forma normal** dada por **(2.1)**:

$$y''(x) = f(x, y(x), y'(x))$$

Ecuación **(2.1)**

La ecuación anterior es una expresión genérica de toda la familia de **EDOs** de 2º orden existente que tienen una forma explícita para la segunda derivada de la solución $y''(x)$, tal y como se puede observar.

Existen varias particularizaciones que se pueden realizar a mayores sobre la expresión dada por **(2.1)**. La ecuación **(2.2)** muestra la forma que toma una **EDO** de 2º orden lineal expresada en forma normal, teniendo en cuenta unos coeficientes genéricos variables p , q y r que acompañan y definen, respectivamente, a la primera derivada y' , la solución y y el término independiente, en un caso no homogéneo.

$$y''(x) = p(x)y'(x) + q(x)y(x) + r(x)$$

Ecuación **(2.2)**

Finalmente, antes de proceder con la teoría básica que concierne a este trabajo en materia de Problemas de Valores Iniciales (**PVIs**) formulados en términos de las ecuaciones presentadas en esta sección, cabe destacar la existencia de una **EDO** con interés especial, particularización una vez más de **(2.1)** y que es independiente de la primera derivada y' . Se tiene **(2.3)**:

$$y''(x) = f(x, y(x))$$

Ecuación **(2.3)**

Sin la imposición de condiciones adicionales que particularicen los problemas que vienen formulados en términos de estas ecuaciones, **las únicas soluciones de las mismas que pueden obtenerse se denominan soluciones generales**. No obstante, como ya se ha mencionado en el **Capítulo 1**, encontrar soluciones de **EDOs** analíticamente no suele ser tarea fácil, puesto que las combinaciones de funciones elementales tienen sus limitaciones y el procedimiento de integración puede ser complicado o imposible, careciendo una gran mayoría de dichas **EDOs** de solución exacta, con el agravante de la presencia de no linealidades en la solución (numérica o analítica) de **(2.1)** y **(2.3)**.



El caso lineal (2.2) es algo más sencillo, pues en ciertas ocasiones puede aplicarse el principio de superposición de soluciones para obtener la solución general de la *EDO*. La solución viene dada en estos casos por la solución general del caso homogéneo y_h y una solución particular del caso no homogéneo y_p (2.4), en el cual se considera $r(x) = 0$.

$$y(x) = y_h(x) + y_p(x)$$

Ecuación (2.4)

2.1.1. Consideraciones sobre la notación de las derivadas.

Un apunte muy importante a la hora de describir fenómenos físicos con aplicaciones a la Ingeniería mediante *EDOs* de la forma (2.1), (2.2) y (2.3), junto con otros tipos de ecuaciones diferenciales no abordados aquí, es la notación utilizada para designar a las derivadas. Mientras que en términos matemáticos suele resultar indiferente emplear un tipo de notación u otra, en Ingeniería el sentido físico ha de ser descrito también por ella. Por tanto, se recuerdan a continuación los distintos tipos de notación de derivadas más utilizados, con el significado habitual de cada uno de ellos.

- i. **Notación de Lagrange.** Se trata de la notación utilizada en este trabajo a la hora de desarrollar todo el marco teórico concerniente a las funciones y ecuaciones de una variable. Resulta ser la forma más empleada y encontrada en los textos matemáticos para designar derivadas. Por ejemplo, la primera derivada de una función genérica g se expresa como g' , la segunda como g'' y así sucesivamente hasta orden n . En Ingeniería, esta notación suele emplearse para describir problemas exclusivamente con variable independiente espacial (en una dimensión, lo habitual es la posición a lo largo de una dirección).
- ii. **Notación de Leibniz.** En esta notación, la derivada de la función genérica $g(x)$ se expresa como $\frac{dg}{dx}$. Del mismo modo, una derivada de orden n se expresaría como $\frac{d^n g}{dx^n}$. Es menos cómoda que la notación de *Lagrange*, pero también se utiliza habitualmente en el cálculo integral, las *EDOs* y el cálculo de varias variables. No existen consideraciones a mayores que le den un significado especial en Ingeniería, pudiéndose utilizar siempre que se desee en la descripción de cualquier fenómeno físico.
- iii. **Notación de Newton.** En contraposición a la notación de *Lagrange*, se trata de la notación empleada para describir fenómenos físicos que presentan evolución temporal, sin ser de uso habitual en los fenómenos con variación espacial descritos por ecuaciones diferenciales. Las derivadas de g se expresan mediante el empleo del mismo número de puntos "acento" que el orden de la derivada que se desea indicar. Así, la primera derivada de g sería \dot{g} , la segunda \ddot{g} ...

La correspondencia entre la notación de *Leibniz* y las otras dos se puede observar en (2.5), para el caso particular de la segunda derivada de g .

$$g''(x) = \frac{d^2g}{dx^2}, \quad \ddot{g} = \frac{d^2g}{dt^2} \rightarrow g = g(t)$$

Ecuación (2.5)

Cuando en este trabajo se aborden problemas temporales en notación de *Newton* empleando estrategias de resolución inicialmente descritas en notación de *Lagrange*, se hará el apunte correspondiente a lo comentado en esta sección.

2.2. Introducción a los Problemas de Valores Iniciales de 2º orden.

Partiendo de las **EDOs** de 2º orden expuestas en la sección **2.1**, más en concreto de la ecuación (2.1), se puede plantear el Problema de Valores Iniciales (**PVI**) dado por (2.6), en donde se considera un intervalo real de la variable independiente $x \in \mathbb{R}$ y sendas condiciones iniciales impuestas sobre la solución y y su primera derivada y' .

$$\begin{cases} y''(x) = f(x, y(x), y'(x)) \\ y(a) = y_0, \quad y'(a) = z_0 \\ x \in [a, b] \end{cases}$$

Ecuación (2.6)

Resulta evidente que, para garantizar la continuidad de la solución en el intervalo $[a, b]$, la función f ha de estar definida en dicho intervalo y ser continua en él. En otras palabras, la segunda derivada y'' está sujeta a las mismas consideraciones comentadas para f , puesto que el **PVI** se expresa en forma normal.

Se puede considerar también el **PVI** lineal dado por (2.7), particularización de (2.6) empleando para su construcción (2.2), teniendo en cuenta que son los coeficientes variables los que ahora condicionan la existencia y la continuidad de la solución en el intervalo considerado $[a, b]$.

$$\begin{cases} y''(x) = p(x)y'(x) + q(x)y(x) + r(x) \\ y(a) = y_0, \quad y'(a) = z_0 \\ x \in [a, b] \end{cases}$$

Ecuación (2.7)

Finalmente, si se emplea la **EDO** en forma normal independiente de y' surge el **PVI** (2.8), con las mismas condiciones iniciales y dominio de la variable independiente que (2.6) y (2.7).

$$\begin{cases} y''(x) = f(x, y(x)) \\ y(a) = y_0, \quad y'(a) = z_0 \\ x \in [a, b] \end{cases}$$

Ecuación (2.8)

Los tres **PVI**s indicados constituyen los problemas de referencia formulados en términos de **EDOs** de 2º orden de los que este trabajo se ocupa. En posteriores capítulos, se harán multitud de referencias explícitamente a ellos, por lo que el lector ha de tenerlos en mente continuamente para comprender todo el material disponible. **Cuando únicamente se haga mención a (2.6), implícitamente se indican referencias a (2.7) y (2.8).**

Antes de comentar las consideraciones relativas a la nomenclatura de los métodos numéricos adoptada para resolver los problemas anteriores, **debe considerarse el siguiente teorema de existencia y unicidad de la solución de un PVI, el cual es condición necesaria (aunque no suficiente) para el correcto desempeño de todas las estrategias desarrolladas en este trabajo, tanto métodos discretos como estrategias de interpolación.**

2.2.1. Teorema de existencia y unicidad de la solución de un PVI.

Considérese ahora, por simplicidad, el **PVI** de 1^{er} orden genérico dado por (2.9):

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(a) = y_0, \quad x \in [a, b] \end{cases}$$

Ecuación (2.9)

La razón de ser de esta simplificación radica en que todas las condiciones relativas a la existencia y unicidad de la solución de (2.9) son perfectamente extrapolables a (2.6), (2.7) y (2.8), siendo además tratadas con mayor frecuencia en la literatura científica [12].

Una función $f(x, y)$ satisface la **condición de Lipschitz** en la variable y en un conjunto $D \in \mathbb{R}^2$, en el cual está definida y es continua, si existe una constante $L > 0$ para la que se cumple (2.10):

$$|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|$$

Ecuación (2.10)

Los dos pares de puntos (x, y_1) y (x, y_2) han de pertenecer al conjunto D . La constante L se denomina constante de **Lipschitz** para la función f en D . Si se supone ahora $D = \{(x, y) \mid a \leq x \leq b, -\infty < y < \infty\}$ manteniendo la existencia y continuidad de f en dicho conjunto, así como el cumplimiento de (2.10) en la variable y , el **PVI (2.9)** tiene solución única $y(x)$ para $x \in [a, b]$.

2.3. Nomenclatura en métodos numéricos.

Como se ha comentado en el **Capítulo 1** de esta memoria, los actores principales del trabajo desarrollado son los métodos numéricos (tanto discretos como estrategias de interpolación de soluciones) empleados para **aproximar** soluciones de problemas formulados en términos de **EDOs** de 2º orden, los cuales ya han sido expuestos en **2.2**.

Por tanto, y dada la necesidad de dar robustez y uniformidad a las expresiones que en posteriores capítulos se desarrollarán, resulta imprescindible establecer una nomenclatura común empleada en todo el trabajo. Comenzando por la discretización del dominio continuo representado por la variable independiente x , se tiene el siguiente **conjunto de $n + 1$ nodos t_k** , ordenado crecientemente y no necesariamente equiespaciado dado por **(2.11)**:

$$x \in [a, b] \rightarrow \{x_k = t_k : k = 0, 1, 2, \dots, n\} \rightarrow \{t_k\}_{k=0}^n, \quad \text{con } \begin{cases} t_0 = a \\ t_n = b \end{cases}$$
$$a = t_0 < t_1 < t_2 < \dots < t_{n-1} < t_n = b$$

Ecuación **(2.11)**

Teniendo en cuenta la expresión anterior, se define el paso h_k como la diferencia entre dos nodos consecutivos t_{k-1} y t_k de **(2.12)**:

$$h_k = t_k - t_{k-1}, \quad k = 1, 2, \dots, n$$

Ecuación **(2.12)**

Nótese que el haber discretizado el intervalo $[a, b]$ con **(2.11)** implica **obtener un número total de n subintervalos contenidos en él**, los cuales se definen en nodos $[t_{k-1}, t_k]$. Cada subintervalo tiene un paso h_k asignado. Por tanto, queda definida la nomenclatura formal de la malla discreta empleada para representar el dominio de la variable independiente continua x de los **PVIs** de **2.2** como un conjunto de nodos perteneciente a dicho dominio y escogido de tal forma que no exista coincidencia de uno con otro. Esto debe hacerse dada la imposibilidad de que un ordenador trabaje con todos los números reales (\mathbb{R} es un conjunto infinito) disponiendo de memoria finita. Si la malla estuviese equiespaciada, **(2.12)** se convertiría en **(2.13)**, con un paso uniforme h :

$$h = \frac{b - a}{n}, \quad t_k = t_{k-1} + h, \quad t_k = t_0 + k \cdot h, \quad k = 1, 2, \dots, n$$

Ecuación **(2.13)**

En cuanto a las soluciones discretas aproximadas que pueden obtenerse de **(2.6)**, **(2.7)** y **(2.8)** en los nodos de **(2.11)** por aplicación de métodos numéricos apropiados, se considera la siguiente notación **(2.14)**:

$$\{y_k: k = 0, 1, 2, \dots, n\} \rightarrow \{y_k\}_{k=0}^n \rightarrow y_k \cong y(t_k)$$

Ecuación (2.14)

Como se tratan problemas formulados en términos de **EDOs** de 2º orden, también se debe disponer de la primera derivada en la solución discreta aproximada. Por tanto, surge (2.15):

$$\{z_k: k = 0, 1, 2, \dots, n\} \rightarrow \{z_k\}_{k=0}^n \rightarrow z_k \cong y'(t_k)$$

Ecuación (2.15)

Finalmente, si en algún momento se dispusiese de los valores de la segunda derivada aproximada en los nodos de (2.11), se tendría la notación dada por (2.16) para dichos valores:

$$\{w_k: k = 0, 1, 2, \dots, n\} \rightarrow \{w_k\}_{k=0}^n \rightarrow w_k \cong y''(t_k)$$

Ecuación (2.16)

En definitiva, valores de la solución evaluados en nodos y expresados entre paréntesis indican el valor exacto que toma dicha solución, mientras que identificadores con subíndices corresponden a aproximaciones en el mismo nodo de la solución exacta del problema, de su primera o de su segunda derivada. Una representación de la nomenclatura adoptada se indica en la **Figura 2.1**.

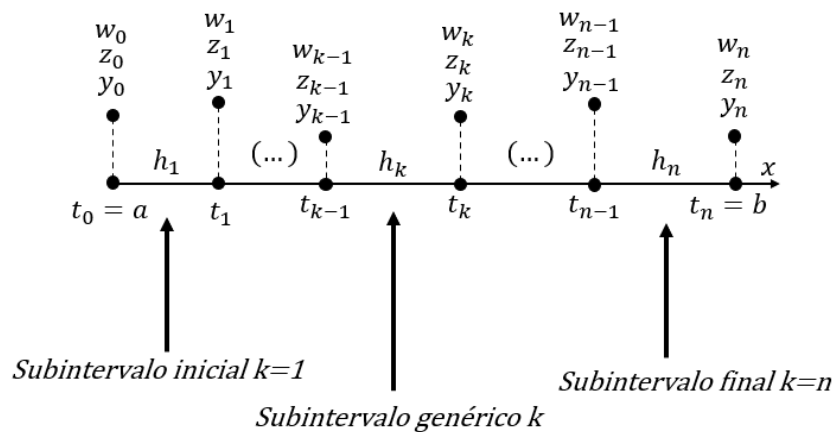


Figura 2.1. Resumen de la nomenclatura empleada en métodos numéricos.



CAPÍTULO 3. Métodos numéricos discretos de un paso.



3.1. Introducción a los métodos numéricos de un paso.

Una vez expuestos todos los Problemas de Valores Iniciales (formulados en términos de **EDOs** de 2º orden) de interés para el desarrollo de este trabajo en el **Capítulo 3**, se puede proceder a introducir lo que constituye uno de los primeros hitos de la investigación a llevar cabo, es decir, dar nociones sobre los **métodos numéricos para la obtención de soluciones discretas aproximadas (2.14)** y primeras derivadas (2.15) en los nodos de la malla definida por (2.11).

Este trabajo se centra en la **experimentación con métodos exclusivamente de un paso**, es decir, algoritmos que necesitan únicamente la aproximación a la solución y_{k-1} o a su primera derivada z_{k-1} en el nodo anterior t_{k-1} para obtener la aproximación (y_k o z_k) en el nodo t_k . Se indican a continuación **las dos familias de métodos a estudiar y a emplear posteriormente**, con una breve descripción de cada uno de ellos:

- i. **Métodos numéricos Runge-Kutta ([3], [12] y [13]).** Su utilización más conocida es para resolver el **PVI** de 1^{er} orden dado por (2.9). No obstante, existen generalizaciones de todos los métodos para que éstos puedan ser aplicados al los **PVIs** de 2º orden expuestos en la sección 2.2, aumentando eso sí el coste computacional y la complejidad algorítmica.
- ii. **Métodos numéricos Runge-Kutta-Nyström.** Especialmente diseñados para resolver de forma directa **PVIs** de 2º orden, constituyen una ventaja sustancial respecto de los métodos *Runge-Kutta*, por su simplicidad y sobre todo por su variedad, ya que existen multitud de métodos específicamente diseñados para problemas particulares, siendo el más común el **PVI** independiente de y' definido en (2.8). Algunas nociones introductorias pueden consultarse en [14] y [15].

A continuación, se indican detalladamente los fundamentos teóricos de ambas familias de métodos, así como multitud de ejemplos que se han considerado para su utilización en este trabajo.

3.2. Métodos Runge-Kutta.

Como ya se ha introducido previamente, de todos los métodos numéricos disponibles en la literatura científica para aproximar las soluciones discretas de un **PVI** de 2º orden este trabajo se centra, por un lado, en emplear métodos numéricos *Runge-Kutta* generalizados respecto del **PVI** de orden 1. Esta familia de métodos fueron desarrollados por primera vez a principios del siglo XX por los matemáticos *Carl Runge* (1856 – 1827) y *Martin Wilhem Kutta* (1867 – 1944). Se aportan (junto a las ya citadas en 3.1) las referencias [16-18] como material más específico de consulta.

3.2.1. Fundamentos teóricos relativos al PVI de 1^{er} orden.

Para comenzar, puede considerarse de momento el PVI de 1^{er} orden (2.9). Manipulándolo, surge la ecuación integral asociada al mismo (3.1):

$$\int_{y_0}^{y(x)} dy = \int_a^x f(z, y(z)) dz \rightarrow y(x) = y_0 + \int_a^x f(z, y(z)) dz$$

Ecuación (3.1)

Un *Runge-Kutta* está basado en la obtención de aproximaciones de la integral de (3.1) mediante métodos de integración numérica apropiados. Estos algoritmos emplean etapas sucesivas basadas en evaluaciones de la función f sin precisar de la evaluación (y correspondiente coste computacional asociado) de sus derivadas. Las s etapas de un método *Runge-Kutta* correspondientes al subintervalo k de la discretización (2.11) (conociendo la aproximación y_{k-1} en t_{k-1}) se definen en (3.2). La aproximación y_k a la solución de (2.9) que produce el método es la dada por (3.3).

$$\begin{cases} k_1 = f\left(t_{k-1} + c_1 h_k, y_{k-1} + h_k \sum_{j=1}^s a_{1j} k_j\right) \\ \vdots \\ k_i = f\left(t_{k-1} + c_2 h_k, y_{k-1} + h_k \sum_{j=1}^s a_{ij} k_j\right) \\ \vdots \\ k_s = f\left(t_{k-1} + c_s h_k, y_{k-1} + h_k \sum_{j=1}^s a_{sj} k_j\right) \end{cases}$$

Ecuación (3.2)

$$y_k = y_{k-1} + h_k \sum_{j=1}^s b_j k_j$$

Ecuación (3.3)

Las constantes a_{ij} agrupadas matricialmente en $\{a_{ij}\}_{i,j=1}^s$ se denominan coeficientes, mientras que $\{c_i\}_{i=1}^s$ son los nodos y $\{b_j\}_{j=1}^s$ los pesos. La denominación de c_i no debe confundirse con su homónima relativa a la malla discreta de (2.11). Las expresiones anteriores se encuentran generalizadas a tamaño de paso variable h_k (2.12). A menudo se suelen imponer condiciones de simplificación sobre los nodos y los pesos, definidas en (3.4):

$$c_i = \sum_{j=1}^s a_{ij}, \quad i = 1, 2, \dots, s, \quad \sum_{j=1}^s b_j = 1$$

Ecuación (3.4)

Una forma muy útil de expresar estos métodos con el objeto de facilitar su construcción y comprensión es el **Tablero de Butcher** [19] (3.5):

$$\begin{array}{c|cccc} c_1 & a_{11} & \cdots & a_{1i} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_i & a_{i1} & \cdots & a_{ii} & \cdots & a_{is} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{si} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_i & \cdots & b_s \end{array}$$

Ecuación (3.5)

De toda la familia *Runge-Kutta*, existen tres subtipos de métodos. Atendiendo al Tablero de *Butcher* de (3.5), se distinguen:

- i. **Métodos RK explícitos:** matriz de coeficientes $\{a_{ij}\}_{i,j=1}^s$ estrictamente triangular inferior con $a_{ij} = 0, j \geq i, j = 1, 2, \dots, s$.
- ii. **Métodos RK semi-implícitos:** si $a_{ij} = 0, j > i, j = 1, 2, \dots, s$.
- iii. **Métodos RK totalmente implícitos:** en todos los demás casos.

3.2.2. Orden de convergencia de los métodos *Runge-Kutta*.

Un método numérico tipo *Runge-Kutta* que aproxima soluciones del *PVI* (2.9) es de orden p si para **problemas suficientemente regulares** se verifican las siguientes condiciones (3.6) relativas a los **Errores Global y Local de Truncamiento** (de ahora en adelante *ELT* y *EGT*, respectivamente):

$$EGT_k = y(t_k) - y_k = O(h^p), \quad h = h_{max}$$

Ecuación (3.6)

Normalmente no se puede evaluar por no disponer de la solución exacta del problema. No obstante, permite determinar el orden de los métodos, los cuales son convergentes si su *EGT* tiende a cero cuando $h_{m\acute{a}x}$ también. El orden también se define como el mayor $p \in \mathbb{N}$ para el que se satisface (3.7):

$$\lim_{h \rightarrow 0} \left(\frac{ELT}{h} \right) = 0$$

Ecuación (3.7)

El error involucrado en (3.7) se denomina **Error Local de Truncamiento (ELT)**, cuyos fundamentos, no tratados aquí, pueden consultarse en [17]. La **Tabla 3.1** recoge información sobre los órdenes de convergencia que presentan los métodos *Runge-Kutta* según sean explícitos o implícitos.

Método RK	Órdenes de convergencia
Explícitos	$p \leq s, \quad \forall s \in [1,5) \quad \quad p < s, \quad \forall s \geq 5$
Implícitos	$p \leq 2s$

Tabla 3.1. Órdenes de convergencia a esperar de los métodos numéricos *Runge-Kutta*.

A continuación, se aportan algunos ejemplos de métodos *Runge-Kutta* obtenidos y estudiados en la literatura científica para aproximar soluciones (de momento) del PVI (2.9) (algunos de los cuales son muy conocidos y pueden deducirse de otros modos) que permiten observar las consideraciones relativas al orden de convergencia dadas por la tabla anterior. Todos los métodos expuestos son particularizaciones de (3.2), (3.3) y (3.5), manteniendo la nomenclatura de los nodos tomados de la malla unidimensional de (2.11).

3.2.3. Método de Euler explícito.

El conocido método de *Euler* o método de las tangentes [12] puede entenderse como un método *Runge-Kutta* explícito de una única etapa y primer orden. El método viene descrito por (3.8):

$$k_1 = f(t_{k-1}, y_{k-1}) \quad \text{a)}$$

$$y_k = y_{k-1} + h_k k_1 \quad \text{b)}$$

$$\frac{0 \mid 0}{\mid 1} \quad \text{c)}$$

Ecuación (3.8)

Este método numérico aproxima la solución de (2.9) mediante el empleo de una línea poligonal. La aproximación que se obtiene será mejor al densificar la malla de (2.11). Además de ser un método *Runge-Kutta*, éste también puede entenderse como una particularización de primer orden de los métodos de *Taylor*, cuyas nociones teóricas también pueden consultarse en [12].

3.2.4. Método de Heun de orden 3.

El método de *Heun* [17] es un método *Runge-Kutta* explícito de tres etapas y, por las consideraciones de la Tabla 1, de tercer orden. El método toma la forma de (3.9):

$$\begin{cases} k_1 = f(t_{k-1}, y_{k-1}) \\ k_2 = f\left(t_{k-1} + \frac{1}{3}h_k, y_{k-1} + \frac{1}{3}h_k k_1\right) \\ k_3 = f\left(t_{k-1} + \frac{2}{3}h_k, y_{k-1} + \frac{2}{3}h_k k_2\right) \end{cases} \quad \text{a)}$$

$$y_k = y_{k-1} + \frac{1}{4} h_k (k_1 + 3k_3) \quad \text{b)}$$

$$\begin{array}{c|ccc} 1/3 & 1/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \end{array} \quad \text{c)}$$

Ecuación (3.9)

A pesar de la mejora del orden de convergencia respecto al método de *Euler*, **no suele ser una práctica común entre los investigadores el emplear métodos *Runge-Kutta* de orden 3**, aunque ha resultado interesante su inclusión en este trabajo.

3.2.5. Método *Runge-Kutta* explícito de orden 4.

Es quizá el método *Runge-Kutta* más universalmente conocido. Se trata de un **método explícito de cuarto orden**. Su demostración puede consultarse en [20], en donde se propone obtener el método de forma sencilla a partir de la aproximación de la integral de (3.1) mediante la regla de *Simpson* $1/3$. Se tiene (3.10):

$$\begin{cases} k_1 = f(t_{k-1}, y_{k-1}) \\ k_2 = f\left(t_{k-1} + \frac{1}{2} h_k, y_{k-1} + \frac{1}{2} h_k k_1\right) \\ k_3 = f\left(t_{k-1} + \frac{1}{2} h_k, y_{k-1} + \frac{1}{2} h_k k_2\right) \\ k_4 = f(t_{k-1} + h_k, y_{k-1} + h_k k_3) \end{cases} \quad \text{a)}$$

$$y_k = y_{k-1} + \frac{1}{6} h_k (k_1 + 2k_2 + 2k_3 + k_4) \quad \text{b)}$$

$$\begin{array}{c|cccc} 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} \quad \text{c)}$$

Ecuación (3.10)

Se adjunta en la **Figura 3.1** una comparación entre los métodos de *Euler* (3.2.3), *Heun* (3.2.4) y el *Runge-Kutta* de orden 4 para observar gráficamente la precisión que son capaces de proporcionar, cuando se conoce la solución analítica de un problema al cual se aplican todos ellos. Esto resulta ser de vital importancia, pues al tratarse de métodos numéricos de un paso la calidad de

las aproximaciones a la solución y_k siempre va a estar influenciada por la calidad de las aproximaciones y_{k-1} .

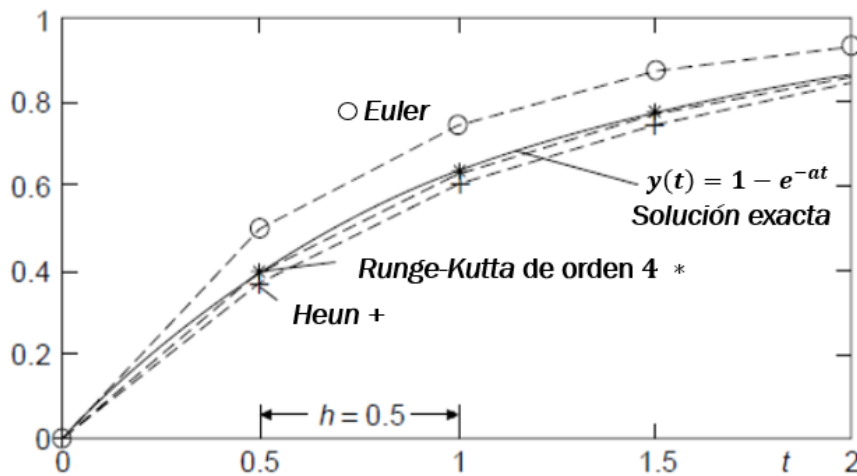


Figura 3.1. Comparativa entre los métodos de Euler, Heun y Runge-Kutta de orden 4.

3.2.6. Método de Gauss implícito de 3 etapas y orden 6.

Como se ha apuntado en la **Tabla 3.1**, emplear métodos Runge-Kutta implícitos permite incrementar el orden de convergencia sin aumentar de forma desproporcionada las etapas en la construcción del método. Tómese este ejemplo (3.11), extraído de [21] y [22], como evidencia de ello.

$$\begin{cases} k_1 = f\left(t_{k-1} + \frac{5 - \sqrt{15}}{10} h_k, y_{k-1} + \frac{5}{36} h_k k_1 + \frac{10 - 3\sqrt{15}}{45} h_k k_2 + \frac{25 - 6\sqrt{15}}{180} h_k k_3\right) \\ k_2 = f\left(t_{k-1} + \frac{1}{2} h_k, y_{k-1} + \frac{10 + 3\sqrt{15}}{72} h_k k_1 + \frac{2}{9} h_k k_2 + \frac{10 - 3\sqrt{15}}{72} h_k k_3\right) \\ k_3 = f\left(t_{k-1} + \frac{5 + \sqrt{15}}{10}, y_{k-1} + \frac{25 + 6\sqrt{15}}{180} h_k k_1 + \frac{10 + 3\sqrt{15}}{45} h_k k_2 + \frac{5}{36} h_k k_3\right) \end{cases} \quad \text{a)}$$

$$y_k = y_{k-1} + \frac{1}{18} h_k (5k_1 + 8k_2 + 5k_3) \quad \text{b)}$$

$\frac{5 - \sqrt{15}}{10}$	$\frac{5}{36}$	$\frac{10 - 3\sqrt{15}}{45}$	$\frac{25 - 6\sqrt{15}}{180}$	c)
$\frac{1}{2}$	$\frac{10 + 3\sqrt{15}}{72}$	$\frac{2}{9}$	$\frac{10 - 3\sqrt{15}}{72}$	
$\frac{5 + \sqrt{15}}{10}$	$\frac{25 + 6\sqrt{15}}{180}$	$\frac{10 + 3\sqrt{15}}{72}$	$\frac{5}{36}$	
	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$	

Ecuación (3.11)

3.2.7. Generalización al PVI de 2º orden.

Los métodos *Runge-Kutta* expuestos anteriormente permiten resolver de forma discreta y aproximada en los nodos de malla (2.11) el PVI (2.9), que es de primer orden. No obstante, este trabajo se centra en las EDOs de 2º orden, por lo que resulta necesario generalizar el esquema *Runge-Kutta* al PVI (2.6). Para ello, se puede partir de un PVI genérico de orden p (3.12):

$$\left\{ \begin{array}{l} y^{(p)}(x) = f(x, y(x), y'(x), \dots, y^{(p-1)}(x)) \\ y(a) = y_0, \\ y'(a) = y'_0, \\ \vdots \\ y^{(p-1)}(a) = y_0^{(p-1)} \\ x \in [a, b] \end{array} \right.$$

Ecuación (3.12)

Se asume que todas las funciones involucradas son continuas en el dominio de la variable independiente x . Ahora, considérese el siguiente sistema de dimensión p (3.13), compuesto por EDOs de 1ª orden, con sus correspondientes condiciones iniciales:

$$\begin{pmatrix} u_1(x) \\ u_2(x) \\ \vdots \\ u_p(x) \end{pmatrix}' = \begin{pmatrix} g_1(x, u_1(x), u_2(x), \dots, u_p(x)) \\ g_2(x, u_1(x), u_2(x), \dots, u_p(x)) \\ \vdots \\ g_p(x, u_1(x), u_2(x), \dots, u_p(x)) \end{pmatrix} \rightarrow \begin{cases} u_1(a) = u_{10} \\ u_2(a) = u_{20} \\ \vdots \\ u_p(a) = u_{p0} \end{cases}, \quad x \in [a, b]$$

Ecuación (3.13)

La expresión equivalente a (3.12) con la forma de (3.13) viene dada por (3.14):

$$\begin{pmatrix} y(x) \\ y_1(x) \\ y_2(x) \\ \vdots \\ y_{p-2}(x) \\ y_{p-1}(x) \end{pmatrix}' = \begin{pmatrix} y_1(x) \\ y_2(x) \\ y_3(x) \\ \vdots \\ y_{p-1}(x) \\ f(x, y(x), y_1(x), y_2(x), \dots, y_{p-2}(x)) \end{pmatrix} \rightarrow \begin{cases} y(a) = y_0 \\ y_1(a) = y'_0 \\ y_2(a) = y''_0 \\ \vdots \\ y_{p-2}(a) = y_0^{(p-2)} \\ y_{p-1}(a) = y_0^{(p-1)} \end{cases}, \quad x \in [a, b]$$

Ecuación (3.14)

Si ahora se particulariza (3.12) a un PVI de 2º orden del tipo (2.6) surge (3.15), el cual es su sistema de EDOs de 1ª orden equivalente:

$$\begin{pmatrix} y(x) \\ y'(x) \end{pmatrix}' = \begin{pmatrix} y'(x) \\ f(x, y(x), y'(x)) \end{pmatrix} \rightarrow \begin{cases} y(a) = y_0 \\ y'(a) = z_0 \end{cases}, \quad x \in [a, b]$$

Ecuación (3.15)



Para poder aplicar un método *Runge-Kutta* inicialmente diseñado para resolver **PVIs** de 1^{er} orden a la resolución aproximada de **PVIs** de 2^o orden es preciso resolver los segundos en su forma dada por (3.15), ya que ésta contiene toda la información necesaria expresada únicamente con **EDOs** de orden 1, a costa eso sí de aumentar la complejidad algorítmica. Las consideraciones inicialmente abordadas en 2.2.1 sobre la existencia y unicidad de la solución de un **PVI** son perfectamente extrapolables a los sistemas de **EDOs** tratados recientemente. Considérese, por tanto, el teorema indicado en la sección 3.2.7.1:

3.2.7.1. Teorema de existencia y unicidad de las soluciones de sistemas de orden p compuestos por **PVIs** de 1^{er} orden.

Tomando como referencia (3.13), la existencia y unicidad de las soluciones de dicho sistema vienen dadas por el cumplimiento de la condición de *Lipschitz* ampliada para funciones de varias variables. Considérese una función $g(x, u_1, \dots, u_p)$ definida y continua en el conjunto D (3.16):

$$D = \{(x, u_1, \dots, u_p) \mid \forall x \in [a, b] \mid \text{con } -\infty < u_i < \infty, \quad i = 1, 2, \dots, p\}$$

Ecuación (3.16)

Esta función satisface la condición de *Lipschitz* en dicho conjunto y en todas su variables si existe una constante $L > 0$ que cumple (3.17) para cualquier (x, u_1, \dots, u_p) y (x, v_1, \dots, v_p) en D .

$$|g(x, u_1, \dots, u_p) - g(x, v_1, \dots, v_p)| \leq L \sum_{j=1}^p |u_j - v_j|$$

Ecuación (3.17)

Además, puede demostrarse que si g y sus primeras derivadas parciales son continuas en D , junto con el cumplimiento de (3.18):

$$\left| \frac{\partial g(u_1, \dots, u_p)}{\partial u_i} \right| \leq L$$

Ecuación (3.18)

para cada $i = 1, 2, \dots, p$ y todas las (x, u_1, \dots, p) en D , entonces g satisface la condición de *Lipschitz* en D con constante de *Lipschitz* L .

Tomando D según (3.16), junto con $g_i(x, u_1, \dots, u_p)$ continua en dicho conjunto y cumpliendo (3.17), el sistema (3.13) tiene una única solución dada por el vector $(u_1(x), \dots, u_p(x))^T$ en $x \in [a, b]$. Puede decirse lo mismo de (3.14) y (3.15), ya que ambas son particularizaciones del caso general. Habiendo garantizado esto último, puede procederse a generalizar los esquemas

planteados para todos los métodos *Runge-Kutta* expuestos previamente. Esto se hace a continuación, en la sección 3.2.7.2.

3.2.7.2. Esquemas *Runge-Kutta* para resolver *PVIs* de 2º orden mediante sistemas de *EDOs* de 1ª orden.

En virtud del teorema de existencia y unicidad de las soluciones, así como del sistema (3.15), se tiene el esquema de etapas dado por (3.19), con las aproximaciones y_k (3.20) que el método proporciona a la solución discreta de (2.6) en (2.11), así como también a su primera derivada z_k (3.21) al tratarse de una generalización a orden 2.

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} k_{11} = z_{k-1} \\ k_{21} = f(t_{k-1}, y_{k-1}, z_{k-1}) \\ \vdots \\ k_{1i} = z_{k-1} + h_k \sum_{j=1}^s a_{ij} k_{2j} \\ k_{2i} = f(t_{k-1} + c_i h_k, y_{k-1} + h_k \sum_{j=1}^s a_{ij} k_{1j}, z_{k-1} + h_k \sum_{j=1}^s a_{ij} k_{2j}) \\ \vdots \\ k_{1s} = z_{k-1} + h_k \sum_{j=1}^s a_{ij} k_{2s} \\ k_{2s} = f(t_{k-1} + c_s h_k, y_{k-1} + h_k \sum_{j=1}^s a_{sj} k_{1j}, z_{k-1} + h_k \sum_{j=1}^s a_{sj} k_{2j}) \end{array} \right. \end{array} \right.$$

Ecuación (3.19)

$$y_k = y_{k-1} + h_k \sum_{j=1}^s b_j k_{1j}$$

Ecuación (3.20)

$$z_k = z_{k-1} + h_k \sum_{j=1}^s b_j k_{2j}$$

Ecuación (3.21)

Los coeficientes, los nodos y los pesos se toman respecto del Tablero de *Butcher* dado por (3.5), al igual que para el caso del *PVI* (2.9). Las etapas k_{1j} son etapas del primer *PVI*, mientras que k_{2j} corresponden al segundo. El carácter *explícito*, *semi-implícito* o *implícito* del método numérico generalizado se determina mediante las mismas consideraciones expuestas en la sección 3.2.1. El orden de los métodos tampoco varía respecto de los estudiados para (2.9).



Todos los métodos *Runge-Kutta* aquí desarrollados se encuentran implementados en *MATLAB* y lógicamente generalizados a la resolución de (2.6) en su forma (3.15). Los códigos pueden consultarse en el ANEXO I a esta memoria, para su posterior utilización en los experimentos numéricos del Capítulo 7. De todos modos, aquí se adjunta la **Tabla 3.2**, la cual es un resumen de todos los códigos disponibles de estos métodos y su alias asignado.

<i>Método RK</i>	<i>Apartado en memoria</i>	<i>Alias en MATLAB</i>
<i>Euler</i> explícito	3.2.3	"ERKs1o"
<i>Heun</i> de orden 3	3.2.4	"HRK3s1o"
<i>RK</i> explícito de orden 4	3.2.5	"RK4s1o"
<i>Gauss</i> implícito de 3 etapas y orden 6	3.2.6	"GRK6s1o"

Tabla 3.2. Métodos *RK* para la resolución del *PVI* de 2º orden.

3.2.8. Pares encajados de métodos *Runge-Kutta*.

Llegados a este punto, conviene dar unas nociones breves sobre el concepto de par encajado. Resulta ser una práctica muy común a la hora de resolver numéricamente *PVIs* emplear métodos numéricos *Runge-Kutta* con esta estrategia ([3] y [23]). Dado que la solución exacta no suele estar disponible (razón de ser principal de un método numérico que obtenga soluciones aproximadas) suelen utilizarse esquemas en los que un método de orden p obtiene la solución del *PVI* al mismo tiempo que otro de orden $p + 1$ se utiliza para evaluar el *ELT* (3.6.b) en combinación con el primero. Es condición necesaria que ambos métodos compartan las mismas etapas k_i . Por tanto, surge el Tablero de *Butcher* (3.22):

c_1	a_{11}	\dots	a_{1i}	\dots	a_{1s}
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
c_i	a_{i1}	\dots	a_{ii}	\dots	a_{is}
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
c_s	a_{s1}	\dots	a_{si}	\dots	a_{ss}
Orden $p \rightarrow b_i$	b_1	\dots	b_i	\dots	b_s
Orden $p + 1 \rightarrow \tilde{b}_i$	\tilde{b}_1	\dots	\tilde{b}_i	\dots	\tilde{b}_s
$b_i - \tilde{b}_i$	$b_1 - \tilde{b}_1$	\dots	$b_i - \tilde{b}_i$	\dots	$b_s - \tilde{b}_s$

Ecuación (3.22)

En ocasiones también se puede utilizar el método de orden $p + 1$ para computar la solución. La definición de las etapas y las aproximaciones que cada método produce vienen dadas por (3.19), (3.20) y (3.21) si el problema es de orden 2 (2.6), o bien (3.2) y (3.3) para el *PVI* (2.9). Una estimación del *ELT* para el método de orden p se da en (3.23):

$$ELT_k \cong \tilde{T}_k = h_k \sum_{j=1}^s (b_j - \tilde{b}_j) k_j$$

Ecuación (3.23)

La **Figura 3.2** representa, a modo de flujograma, la secuencia que sigue un par encajado mediante **ajustes del paso h_k** (2.12) partiendo de una tolerancia prefijada **TOL** .

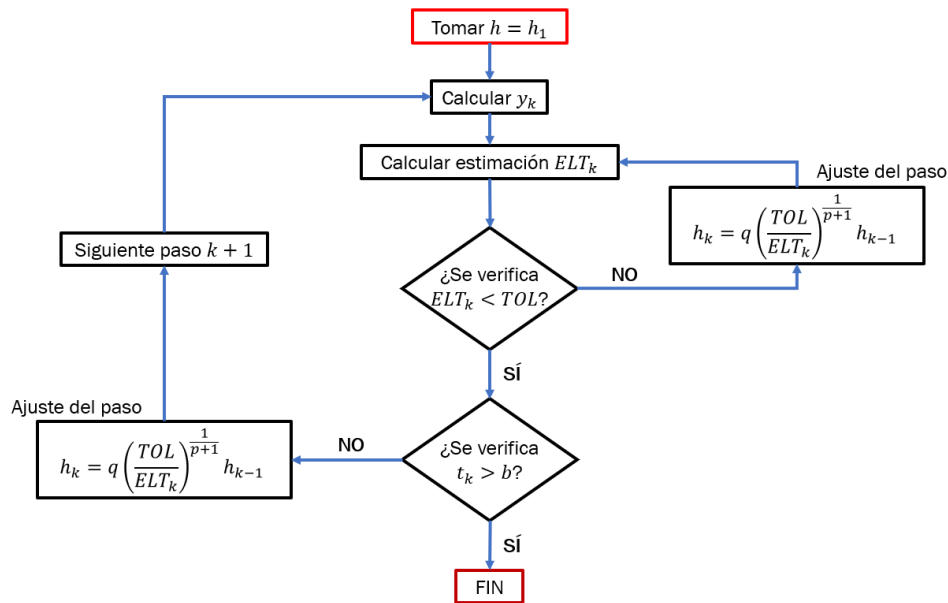


Figura 3.2. Secuencia de un par encajado de métodos *Runge-Kutta*.

La implementación de pares encajados escapa del ámbito de este trabajo. No obstante, y dado que se emplea **MATLAB** para la realización de todos los experimentos numéricos oportunos, **se escogen algunos de los pares disponibles en la librería de solvers del software, expuestos a continuación.**

3.2.9. Métodos *Runge-Kutta* implementados en solvers para PVI de MATLAB. Problemas rígidos y no rígidos.

Para comprender las alternativas ofertadas por **MATLAB** en cuando a pares encajados se refiere, conviene hacer una breve **distinción entre problemas rígidos y no rígidos** (conocidos en la literatura como **stiff** y **no stiff**, respectivamente), ambos formulados en términos de **EDOs**.

Un **PVI rígido** es aquel cuya solución exacta tiene un término que toma la forma de e^{-ct} , siendo c una constante positiva grande. Esta parte de la solución se denomina **transitoria**, mientras que el resto es la **solución de estado estable**. El término transitorio citado decrece a gran velocidad a cero conforme t aumenta. En cambio, su derivada no decae tan rápido (la n -ésima derivada es $c^n e^{-ct}$). **Las características anteriores pueden conllevar a resultados totalmente desastrosos a la hora de aplicar métodos numéricos para obtener su solución.** Por otro lado, un **PVI no rígido** es aquel que no cumple nada de lo citado en este

párrafo, presentando menos problemas de resolución con métodos de un paso. Si el lector quiere profundizar, puede consultar [12] y [24].

Las **Tablas 3.3 y 3.4 (solvers de MATLAB para problemas no rígidos y rígidos, respectivamente)** son una adaptación de la documentación accesible en [25] y [26], y que contiene bibliografía ideal para profundizar el conocimiento, también citada aquí.

<i>Solver</i>	<i>Algoritmo</i>	<i>Utilización</i>
ode45 (precisión media)	<i>RK</i> explícito con par encajado 4(5) de <i>Dormand-Prince</i> [23].	En la mayoría de ocasiones, debería de ser el primer <i>solver</i> a emplear.
ode23 (precisión baja)	<i>RK</i> explícito con par encajado 2(3) de <i>Bogacki-Shampine</i> [27].	Puede ser más eficiente que ode45 en problemas con tolerancias más grandes, o en presencia de rigidez moderada.
ode78 (precisión alta)	<i>RK</i> con par encajado 8(7) más eficiente de <i>Verner</i> [28].	Puede ser más eficiente que ode45 en problemas con soluciones suaves que necesitan elevada precisión.
ode89 (precisión alta)	<i>RK</i> con par encajado 9(8) más robusto de <i>Verner</i> [28].	Puede ser más eficiente que ode78 en problemas muy suaves cuando se integra en intervalos temporales extensos, o cuando las tolerancias son especialmente pequeñas.

Tabla 3.3. Solvers de *MATLAB* para problemas no rígidos que implementan métodos *Runge-Kutta*.

<i>Solver</i>	<i>Algoritmo</i>	<i>Utilización</i>
ode23s (precisión baja)	<i>RK-Rosenbrock</i> modificado de orden 2 [29].	Eficiente en problemas con tolerancias grandes. Este algoritmo computa el Jacobiano en cada paso.
ode23t (precisión baja)	<i>RK</i> trapezoidal que emplea un interpolador libre [30].	Útil en problemas con rigidez moderada y que precisan de una solución libre de amortiguamiento numérico.
ode23tb (precisión baja)	<i>RK</i> implícito <i>TR-BDF2</i> [31].	Eficiente, al igual que ode23s , en problemas con tolerancias grandes.

Tabla 3.4. Solvers de *MATLAB* para problemas rígidos que implementan métodos *Runge-Kutta*.

Algunos de los *solvers* anteriores resultan interesantes para su utilización a la hora de obtener soluciones discretas aproximadas de **PVIs** en este trabajo.

3.3. Métodos *Runge-Kutta-Nyström*.

Tras la extensa presentación y análisis de los distintos métodos numéricos *Runge-Kutta* empleados en este trabajo, la conclusión es evidente: su generalización para resolver el *PVI* de 2º orden es correcta e implementable en un software de cálculo numérico como *MATLAB*, pero la imposibilidad de abordar la resolución de (2.6) sin el empleo de su sistema de *EDOs* de orden 1 equivalente (3.15) aumenta la complejidad algorítmica.

Por ello, en 1925, el matemático finlandés *EJ Nyström* (1895 – 1960) planteó una formulación que permite resolver (2.6) directamente, es decir, sin recurrir a ningún sistema de *EDOs* ni cambios de variable. **Estos métodos alternativos son conocidos como *Runge-Kutta-Nyström*** (de forma abreviada *RKN*), pues se obtienen a partir de los *Runge-Kutta* y conservan ciertas características de los mismos. **La reducción del coste computacional se estima en un 25% globalmente.**

3.3.1. Obtención de los métodos *Runge-Kutta-Nyström* a partir de los métodos *Runge-Kutta*.

Considérese el método *Runge-Kutta* generalizado a sistemas de *EDOs* del tipo (3.15) dado por las expresiones (3.19), (3.20) y (3.21) de la sección 3.2.7.2. Insertando las etapas k_{1i} en las etapas k_{2i} , se obtienen (3.24), (3.25) y (3.26) ([22] y [32]):

$$k_i = f \left(t_{k-1} + c_i h_k, y_{k-1} + c_i h_k z_{k-1} + h_k^2 \sum_{j=1}^s \bar{a}_{ij} k_j, y_{k-1} + h_k \sum_{j=1}^s a_{ij} k_j \right)$$

Ecuación (3.24)

$$y_k = y_{k-1} + h_k z_k + h_k^2 \sum_{i=1}^s \bar{b}_i k_i$$

Ecuación (3.25)

$$z_k = z_{k-1} + h_k \sum_{i=1}^s b_i k_i$$

Ecuación (3.26)

(3.24) corresponde a la etapa i -ésima de un método *Runge-Kutta-Nyström*. Las dos últimas ecuaciones corresponden a la aproximación y_k que el método numérico es capaz de proporcionar a la solución discreta de (2.6) en (2.11) obtenida directamente y a su primera derivada z_k , conservando la notación de coeficientes, nodos y pesos indicada con anterioridad para los métodos *Runge-Kutta*. Estos últimos pueden tomar las formas de (3.27), sin que sea un requisito indispensable el cumplimiento de las mismas.

$$\bar{a}_{ij} = \sum_{k=1}^s a_{ik} a_{kj}, \quad \bar{b}_i = \sum_{j=1}^s b_j a_{ji}$$

Ecuación (3.27)

Con todas las consideraciones anteriores, un método *Runge-Kutta-Nyström* genérico de s etapas viene dado por (3.28):

$$\begin{cases} k_1 = f \left(t_{k-1} + c_1 h_k, y_{k-1} + c_1 h_k z_{k-1} + h_k^2 \sum_{j=1}^s \bar{a}_{1j} k_j, y_{k-1} + h_k \sum_{j=1}^s a_{1j} k_j \right) \\ \vdots \\ k_i = f \left(t_{k-1} + c_i h_k, y_{k-1} + c_i h_k z_{k-1} + h_k^2 \sum_{j=1}^s \bar{a}_{ij} k_j, y_{k-1} + h_k \sum_{j=1}^s a_{ij} k_j \right) \\ \vdots \\ k_s = f \left(t_{k-1} + c_s h_k, y_{k-1} + c_s h_k z_{k-1} + h_k^2 \sum_{j=1}^s \bar{a}_{sj} k_j, y_{k-1} + h_k \sum_{j=1}^s a_{sj} k_j \right) \end{cases}$$

Ecuación (3.28)

Reorganizando toda la información, surge el siguiente Tablero de *Butcher* (3.29), práctica común (al igual que con los métodos *Runge-Kutta*) a la hora de representar de forma compacta y eficiente el método en cuestión.

c_1	\bar{a}_{11}	\bar{a}_{12}	\cdots	\bar{a}_{1s}	a_{11}	a_{12}	\cdots	a_{1s}
c_i	\bar{a}_{i1}	\bar{a}_{i2}	\cdots	\bar{a}_{is}	a_{i1}	a_{i2}	\cdots	a_{is}
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
c_s	\bar{a}_{s1}	\bar{a}_{s2}	\cdots	\bar{a}_{ss}	a_{s1}	a_{s2}	\cdots	a_{ss}
	\bar{b}_1	\bar{b}_2	\cdots	\bar{b}_s	b_1	b_2	\cdots	b_s

Ecuación (3.29)

El carácter *explícito*, *semi-implícito* e *implícito* de los métodos responde también a las consideraciones establecidas para los métodos *Runge-Kutta* en la sección 3 de este capítulo y su Tablero de *Butcher* asociado (3.5), con la única diferencia de que un método *Runge-Kutta-Nyström* presenta un tablero que podría entenderse como expandido, por lo que se tienen dos matrices de coeficientes $\{\bar{a}_{ij}\}$ y $\{a_{ij}\}$.

3.3.2. Métodos *Runge-Kutta-Nyström* para problemas independientes de y' .

Una ventaja importante que también aportan los métodos *RKN* es la existencia de multitud de ellos específicamente diseñados para resolver el *PVI* independiente de la primera derivada y' (2.8). La simplificación algorítmica es grande, pues a pesar de que las evaluaciones de f siguen siendo las mismas,

ahora no ha de evaluarse ninguna expresión relacionada con y' . Esto tiene implicaciones directas en las expresiones de las etapas de los métodos, modificadas respecto a (3.28). La etapa i -ésima ahora viene dada por (3.30):

$$k_i = f \left(t_{k-1} + c_i h_k, y_{k-1} + c_i h_k z_{k-1} + h_k^2 \sum_{j=1}^s \bar{a}_{ij} k_j \right)$$

Ecuación (3.30)

Al igual que el método genérico para resolver *PVIs* completos del tipo (2.6), las aproximaciones conseguidas a la solución discreta y_k y a su primera derivada z_k toman la forma de (3.25) y (3.26), respectivamente. El Tablero de *Butcher* empleado ahora (3.31) es un tablero simplificado respecto a (3.29), pues al no existir dependencia del problema con respecto a y' , desaparece la matriz de coeficientes $\{a_{ij}\}$.

c_1	\bar{a}_{11}	\bar{a}_{12}	\cdots	$\bar{a}_{1(s-1)}$	\bar{a}_{1s}
c_i	\bar{a}_{i1}	\bar{a}_{i2}	\cdots	$\bar{a}_{i(s-1)}$	\bar{a}_{is}
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
c_s	\bar{a}_{s1}	\bar{a}_{s2}	\cdots	$\bar{a}_{s(s-1)}$	\bar{a}_{ss}
	\bar{b}_1	\bar{b}_2	\cdots	\bar{b}_{s-1}	\bar{b}_s
	b_1	b_2	\cdots	b_{s-1}	b_s

Ecuación (3.31)

3.3.3. Orden de convergencia de los métodos *Runge-Kutta-Nyström*.

Un método numérico *RKN* es de orden p si para **problemas suficientemente regulares** se verifican las dos condiciones expuestas a continuación en (3.32) y (3.33), relativas a los *EGT* de la solución y su primera derivada en t_k , respectivamente.

$$EGT_{y,k} = y(y_k) - y_k = O(h^p), \quad h = h_{max}$$

Ecuación (3.32)

$$EGT_{z,k} = y'(t_k) - z_k = O(h^p), \quad h = h_{max}$$

Ecuación (3.33)

También es aplicable la definición del *ELT* aportada en (3.6.b), así como la definición de método convergente. A continuación se indican algunos ejemplos **explícitos** de esta familia de métodos (junto con sus correspondientes referencias bibliográficas), tanto aplicables a *PVI* completos del tipo (2.6), como a los independientes de la primera derivada y' (2.8). Aquellos específicamente diseñados para (2.8) tan sólo pueden ser aplicados en ese tipo de problemas.



Para su posterior utilización en los experimentos numéricos (**Capítulo 7**), se adjuntan los códigos en **MATLAB** de todos ellos en el **ANEXO II**.

3.3.4. Métodos *Runge-Kutta-Nyström* explícitos para el PVI completo de la forma $y''(x) = f(x, y(x), y'(x))$.

La **Tabla 3.5** muestra un resumen sobre los métodos **RKN** implementados en **MATLAB** correspondientes a la tipología que concierne a este apartado. Se indica el alias otorgado a cada uno de ellos para que el lector pueda identificarlos sin problema en los códigos adjuntos en el **ANEXO II**.

Método RKN	Apartado en memoria	Alias en MATLAB®
Directo de orden 3	3.3.4.1	"RKN3"
Transformado de orden 3	3.3.4.2	"RKN3t"
Orden 4	3.3.4.3	"RKN4"

Tabla 3.5. Métodos **RKN** explícitos para el **PVI** completo de la forma $y''(x) = f(x, y(x), y'(x))$.

3.3.4.1. Método **RKN** directo de orden 3.

Este método ha sido consultado en [22]. Las expresiones que toman las etapas, así como las aproximaciones obtenidas a la solución discreta de (2.6) en los nodos de (2.11) y su primera derivada vienen dadas por (3.34):

$$\begin{cases} k_1 = f(t_{k-1}, y_{k-1}, z_{k-1}) \\ k_2 = f\left(t_{k-1} + \frac{1}{2}h_k, y_{k-1} + \frac{1}{2}h_k z_{k-1} + \frac{1}{8}h_k^2 k_1, z_{k-1} + \frac{1}{2}h_k k_1\right) \\ k_3 = f\left(t_{k-1} + \frac{3}{4}h_k, y_{k-1} + \frac{3}{4}h_k z_{k-1} + \frac{9}{32}h_k^2 k_2, z_{k-1} + \frac{3}{4}h_k k_2\right) \end{cases} \quad \text{a)}$$

$$y_k = y_{k-1} + h_k z_{k-1} + h_k^2 \left(\frac{2}{9} k_1 + \frac{1}{6} k_2 + \frac{1}{9} k_3 \right) \quad \text{b)}$$

$$z_k = z_{k-1} + h_k \left(\frac{2}{9} k_1 + \frac{1}{3} k_2 + \frac{4}{9} k_3 \right) \quad \text{c)}$$

Ecuación (3.34)

El Tablero de *Butcher* asociado viene dado por (3.35):

$$\begin{array}{c|cc|cc} 0 & & & & & & \\ \hline 1/2 & 1/8 & & & 1/2 & & \\ 3/4 & 0 & 9/32 & & 0 & 3/4 & \\ \hline & 2/9 & 1/6 & 1/9 & 2/9 & 1/3 & 4/9 \end{array}$$

Ecuación (3.35)

3.3.4.2. Método *RKN* transformado de orden 3.

Disponible en [18] como variante al *RKN* recientemente expuesto (ajustando los coeficientes de la matriz $\{\bar{a}_{ij}\}$), tiene su fundamento en las etapas dadas por (3.36), junto con las aproximaciones pertinentes que derivan de ellas.

$$\left\{ \begin{array}{l} k_1 = f(t_{k-1}, y_{k-1}, z_{k-1}) \\ k_2 = f\left(t_{k-1} + \frac{1}{2}h_k, y_{k-1} + \frac{1}{2}h_k z_{k-1}, z_{k-1} + \frac{1}{2}h_k k_1\right) \\ k_3 = f\left(t_{k-1} + \frac{3}{4}h_k, y_{k-1} + \frac{3}{4}h_k z_{k-1} + \frac{3}{8}h_k^2 k_1, z_{k-1} + \frac{3}{4}h_k k_2\right) \end{array} \right. \quad \text{a)}$$

$$y_k = y_{k-1} + h_k z_{k-1} + h_k^2 \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 \right) \quad \text{b)}$$

$$z_k = z_{k-1} + h_k \left(\frac{2}{9}k_1 + \frac{1}{3}k_2 + \frac{4}{9}k_3 \right) \quad \text{c)}$$

Ecuación (3.36)

Para este método, se tiene el Tablero de *Butcher* siguiente (3.37):

0				1/2		
1/2	0			0	3/4	
3/4	3/8	0		2/9	1/3	4/9
	1/6	1/3	0			

Ecuación (3.37)

3.3.4.3. Método *RKN* de orden 4.

Si se considera que la precisión que proporcionan los métodos expuestos en 3.3.4.1 y 3.3.4.2 no es suficiente, surge la alternativa de emplear un método *Runge-Kutta-Nyström* de orden 4 [17], como el que viene descrito por las etapas dadas en (3.38):

$$\left\{ \begin{array}{l} k_1 = f(t_{k-1}, y_{k-1}, z_{k-1}) \\ k_2 = f\left(t_{k-1} + \frac{1}{2}h_k, y_{k-1} + \frac{1}{2}h_k z_{k-1} + \frac{1}{8}h_k^2 k_1, z_{k-1} + \frac{1}{2}h_k k_1\right) \\ k_3 = f\left(t_{k-1} + \frac{1}{2}h_k, y_{k-1} + \frac{1}{2}h_k z_{k-1} + \frac{1}{8}h_k^2 k_1, z_{k-1} + \frac{1}{2}h_k k_2\right) \\ k_4 = f\left(t_{k-1} + h_k, y_{k-1} + h_k z_{k-1} + \frac{1}{2}h_k^2 k_3, z_{k-1} + h_k k_3\right) \end{array} \right. \quad \text{a)}$$

$$y_k = y_{k-1} + h_k z_{k-1} + h_k^2 \left(\frac{1}{6}k_1 + \frac{1}{6}k_2 + \frac{1}{6}k_3 \right) \quad \text{b)}$$

$$z_k = z_{k-1} + h_k \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right) \quad \text{c)}$$

Ecuación (3.38)

0				
1/2	1/8			
1	0	1/2		
	1/6	1/3	0	
	1/6	2/3	1/6	

Ecuación (3.41)

3.3.5.2. Método RKN simple de orden 5.

Puede incrementarse la precisión obtenida por los métodos de orden 4 empleando un método de orden 5. La descripción detallada de las etapas de este método (consultado en [34]) viene dada por (3.42):

$$\left\{ \begin{array}{l}
 k_1 = f(t_{k-1}, y_{k-1}) \\
 k_2 = f\left(t_{k-1} + \frac{1}{5}h_k, y_{k-1} + \frac{1}{5}h_k z_{k-1} + \frac{1}{50}h_k^2 k_1\right) \\
 k_3 = f\left(t_{k-1} + \frac{2}{3}h_k, y_k + \frac{2}{3}h_k z_{k-1} + h_k^2\left(-\frac{1}{27}k_1 + \frac{7}{27}k_2\right)\right) \\
 k_4 = f\left(t_{k-1} + h_k, y_{k-1} + h_k z_{k-1} + h_k^2\left(\frac{3}{10}k_1 - \frac{2}{35}k_2 + \frac{9}{35}k_3\right)\right)
 \end{array} \right. \quad \text{a)}$$

$$y_k = y_{k-1} + h_k z_{k-1} + h_k^2\left(\frac{1}{24}k_1 + \frac{25}{84}k_2 + \frac{9}{56}k_3\right) \quad \text{b)}$$

$$z_k = z_{k-1} + h_k\left(\frac{1}{24}k_1 + \frac{125}{336}k_2 + \frac{27}{56}k_3 + \frac{5}{48}k_4\right) \quad \text{c)}$$

Ecuación (3.42)

El Tablero de *Butcher* asociado es el dado por (3.43):

0				
0	1/50			
2/3	-1/27	7/27		
1	3/10	-2/35	9/35	
	1/24	25/84	9/56	0
	1/24	125/336	27/56	5/48

Ecuación (3.43)

3.3.5.3. Método RKN especial de orden 5.

Como variante al método expuesto en el apartado anterior, se puede emplear un método especial del mismo orden [35]. Por ello, se tienen las etapas y aproximaciones correspondientes dadas por (3.44):



$$\left\{ \begin{array}{l} k_1 = f(t_{k-1}, y_{k-1}) \\ k_2 = f\left(t_{k-1} + \frac{2}{5}h_k, y_{k-1} + \frac{2}{5}h_k z_{k-1} + \frac{2}{25}h_k^2 k_1\right) \\ k_3 = f\left(t_{k-1} + \frac{2}{3}h_k, y_{k-1} + \frac{2}{3}h_k z_{k-1} + \frac{2}{9}h_k^2 k_1\right) \\ k_4 = f\left(t_{k-1} + \frac{4}{5}h_k, y_{k-1} + \frac{4}{5}h_k z_{k-1} + h_k^2\left(\frac{4}{25}k_1 + \frac{4}{25}k_2\right)\right) \end{array} \right. \quad \text{a)}$$

$$y_k = y_{k-1} + h_k z_{k-1} + h_k^2 \left(\frac{23}{192} k_1 + \frac{25}{64} k_2 - \frac{9}{64} k_3 + \frac{25}{192} k_4 \right) \quad \text{b)}$$

$$z_k = z_{k-1} + h_k \left(\frac{23}{192} k_1 + \frac{125}{192} k_2 - \frac{27}{64} k_3 + \frac{125}{192} k_4 \right) \quad \text{c)}$$

Ecuación (3.44)

En consecuencia, se tiene el Tablero de Butcher de (3.45):

0				
2/5	2/25			
2/3	2/9	0		
4/5	4/25	4/25	0	
	23/192	25/64	-9/64	25/192
	23/192	125/192	-27/64	125/192

Ecuación (3.45)



CAPÍTULO 4. Estrategias de interpolación de soluciones de *PVIs* empleando *splines*.



4.1. Introducción a las estrategias de interpolación.

Tras la extensa exposición realizada en el **Capítulo 3** sobre algunos de los métodos numéricos de un paso disponibles en la literatura científica que obtienen aproximaciones de las soluciones discretas (y de sus derivadas) de problemas formulados en términos de **EDOs** y que toman las formas expuestas en la sección 2 del **Capítulo 2**, surge la necesidad de **obtener funciones adecuadas que garanticen la existencia de una aproximación continua en los intervalos de las variables independientes partiendo de dichas soluciones ya calculadas previamente.**

La cualidad “adecuada” de estas funciones debe entenderse como aquella que les proporciona la **máxima regularidad posible y la mayor precisión a la hora de reproducir, tanto cualitativamente como cuantitativamente, la forma de las soluciones a obtener (así como la de sus derivadas).** Esto no es sencillo, pues han de imponerse una serie de condiciones minuciosamente que sean capaces de materializar las soluciones continuas (mayoritariamente desconocidas inicialmente) y no disponibles hasta que se aplique una estrategia idónea basada en el empleo de métodos numéricos. Un gran número de problemas de Ingeniería no pueden ser resueltos analíticamente, por lo que conocer y ser capaz de adoptar estas estrategias es una competencia extremadamente importante e interesante que todo ingeniero ha de tener.

En atención a las consideraciones anteriores, algunas de las cuales ya se anticipaban en el **Capítulo 1** de esta memoria, se proponen distintas **estrategias de interpolación de soluciones discretas** para **PVIs** del tipo (2.6) en la malla de nodos definida por (2.11) basadas en **trazadores polinomiales, también conocidos como splines.** Evidentemente, y como ya se ha comentado, cuando se cita únicamente (2.6) como problema de referencia para la aplicabilidad de un método, ha de entenderse que (2.7) y (2.8) también son tenidos en cuenta, por ser particularizaciones del caso genérico.

4.2. Utilización de splines como estrategia para interpolar soluciones de PVI.

Los *splines* son una clase muy importante de funciones matemáticas utilizadas en análisis numérico para aproximación. Un *spline* es una una función polinómica definida a trozos descrita como “**lo más suave y regular posible sin tener que ser reducida a un polinomio**” [36].

Durante los años 60 se reconoció que estos trazadores polinomiales podían modelizar matemáticamente fenómenos físicos, siendo empleados en el análisis numérico moderno con muchísima frecuencia al ser fáciles de evaluar. Se incluyen algunas referencias bibliográficas actuales que investigan en su construcción y aplicaciones [37-42]. De entre todas ellas, algunas son:

- Interpolación y aproximación polinomial segmentaria.

- Diseño de curvas y superficies no muy pesadas computacionalmente. Aplicaciones directas en el Diseño Asistido por Ordenador (CAD), rama de la Ingeniería a su vez empleada en automoción, aviación, construcción...
- Procesos industriales basados en teorías de control, modelizados mayoritariamente por ecuaciones diferenciales.
- Múltiples rutinas que involucran procesos de derivación e integración numérica eficientes, así como problemas formulados en la obtención de autovalores y autovectores para operadores.
- Descripción de modelos probabilísticos con aplicaciones físicas, ingenieriles, geológicas e incluso médicas.

En atención a la primera aplicación de los *splines* citada, **su utilización en este trabajo se restringe a la interpolación de las soluciones discretas de PVI** obtenidas por los métodos numéricos del **Capítulo 3** (o exactas si existiesen). Por ello, se recuerda al lector la definición de interpolación. Considérese una función genérica $f(x)$ en dos valores distintos de x (denominados x_0 y x_1 , con $x_0 < x_1$). Ahora, esta función puede ser aproximada en $[x_0, x_1]$ mediante el empleo de otra función $g(x)$ que satisface **(4.1)**:

$$g(x_0) = f(x_0), \quad g(x_1) = f(x_1), \quad g(x) \cong f(x) \rightarrow x \in (x_0, x_1)$$

Ecuación **(4.1)**

Si este concepto se extiende al **PVI (2.6)** y se discretiza su dominio empleando los $n + 1$ nodos de **(2.11)**, surge un *spline* S definido a trozos por polinomios S_k de **grado a lo sumo p** dados por **(4.2)** en cada uno de n subintervalos generados:

$$S_k(x) = \sum_{i=0}^p a_{ki} (x - t_{k-1})^i, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n, \quad a_{ki} \in \mathbb{R}$$

Ecuación **(4.2)**

Ahora, téngase en cuenta que, **para determinar unívocamente dicho *spline*, se han de imponer $(p + 1)n$ condiciones**, compuestas por condiciones de continuidad hasta $S \in C_{[a,b]}^{p-1}$ y de interpolación $S(t_k) = y_k$ (siendo y_k la solución discreta del **PVI (2.6)** en cada nodo de **(2.11)**), así como alguna condición adicional relativa a las condiciones iniciales del problema. Este esquema de imposición se resume en la **Tabla 4.1**:

<i>Tipos de condiciones</i>	<i>Número</i>	<i>Nodos en donde se imponen</i>
Continuidad (regularidad)	$p(n - 1)$	$\{t_k\}_{k=1}^{n-1}$
Interpolación	$n + 1$	$\{t_k\}_{k=0}^n$
Adicionales	$p - 1$	$t_0 = a$
TOTALES		$(p + 1)n$

Tabla 4.1. Resumen de condiciones a imponer para determinar unívocamente un *spline* interpolador.

En este capítulo, **se adoptan dos estrategias** de interpolación de las soluciones discretas del **PVI (2.6)** empleando *splines*, las cuales se citan a continuación y son desarrolladas en posteriores apartados con detalle, además de incluir referencias a las funciones escritas en **MATLAB** para poder realizar los experimentos numéricos del **Capítulo 7**.

- i. Estrategia empleando *splines* cuadráticos interpoladores.
- ii. Estrategia empleando *splines* cúbicos interpoladores.

4.3. Estrategia empleando *splines* cuadráticos interpoladores.

Para obtener la aproximación continua de (2.6) citada con anterioridad, se pueden emplean *splines* cuadráticos. En concordancia con (4.2), un *spline* cuadrático S viene descrito en cada intervalo $[t_{k-1}, t_k]$ por un polinomio S_k , de grado $p \leq 2$ (4.3):

$$S_k(x) = \sum_{i=0}^{p=2} a_{ki} (x - t_{k-1})^i, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n, \quad a_{ki} \in \mathbb{R}$$

Ecuación (4.3)

Desarrollando la expresión anterior y teniendo en cuenta la existencia de $n + 1$ nodos ordenados crecientemente según (2.11), se tiene (4.4):

$$S(x) = \begin{cases} S_1(x) = a_{10} + a_{11}(x - t_0) + a_{12}(x - t_0)^2, & \text{si } x \in [t_0, t_1] \\ S_2(x) = a_{20} + a_{21}(x - t_1) + a_{22}(x - t_1)^2, & \text{si } x \in [t_1, t_2] \\ \dots \\ S_n(x) = a_{n0} + a_{n1}(x - t_{n-1}) + a_{n2}(x - t_{n-1})^2, & \text{si } x \in [t_{n-1}, t_n] \end{cases}$$

Ecuación (4.4)

4.3.1. Justificación del empleo de *splines* cuadráticos.

Utilizar *splines* cuadráticos en la interpolación de soluciones discretas de (2.6) es adecuado, gracias a que se proporcionan aproximaciones de clase C^1 a la solución. El significado físico que estos trazadores aportan viene dado por la continuidad de los mismos y la de su primera derivada. Esto implica poder esperar **buenas aproximaciones a problemas cuya velocidad sea importante de monitorizar**. Las expresiones (4.5) y (4.6) muestran las derivadas en cada intervalo $[t_{k-1}, t_k]$.

$$S'(x) = \begin{cases} S'_1(x) = a_{11} + 2a_{12}(x - t_0), & \text{si } x \in [t_0, t_1] \\ S'_2(x) = a_{21} + 2a_{22}(x - t_1), & \text{si } x \in [t_1, t_2] \\ \dots \\ S'_n(x) = a_{n1} + 2a_{n2}(x - t_{n-1}), & \text{si } x \in [t_{n-1}, t_n] \end{cases}$$

Ecuación (4.5)

$$S''_k(x) = 2a_{k2}, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n$$

Ecuación (4.6)

En los extremos de los intervalos, las derivadas habrán de ser consideradas derivadas laterales (por la derecha en los extremos iniciales y por la izquierda en los finales). Esto es importante, pues como puede observarse en (4.6), la segunda derivada del *spline* cuadrático ha de considerarse constante a trozos. Lógicamente, al no imponer continuidad C^2 , dicha derivada no es continua. **Por tanto, en problemas cuya aceleración resulta importante para su estudio (por ejemplo, oscilatorios), no es muy buena estrategia emplear estos *splines*.**

La determinación total del *spline* de segundo grado se consigue mediante la imposición de las condiciones citadas en la **Tabla 4.1**, con $p = 2$. En total, el número de parámetros a_{ki} a determinar es $3n$, siendo $n + 1$ la cantidad de nodos y n los polinomios de segundo grado S_k en los intervalos generados inicial ($x \in [t_0, t_1]$) y restantes ($x \in [t_1, t_n]$). Corresponde imponer una condición adicional a mayores de las de continuidad e interpolación. Con todo lo anterior, se tiene la **Tabla 4.2**:

Tipo	Número	Forma
Continuidad	$2(n - 1)$	S y S' continuas en $\{t_{k-1}\}_{k=2}^n$ $\begin{cases} S_{k-1}(t_{k-1}) = S_k(t_{k-1}) \rightarrow n - 1 \\ S'_{k-1}(t_{k-1}) = S'_k(t_{k-1}) \rightarrow n - 1 \end{cases}$
Interpolación	$n + 1$	$S(t_k) = y_k$ $(k = 0, 1^{\text{a}} \text{ Cond. Inicial PVI(2.6)})$
Adicionales	$1 \rightarrow 2^{\text{a}} \text{ Cond. Inicial PVI(2.6)}$	$y'(a) = y'(t_0) = z_0$
TOTALES		$3n$

Tabla 4.2. Resumen de condiciones a imponer para determinar unívocamente un *spline* cuadrático interpolador.

4.3.2. Resolución del intervalo inicial. Obtención de S_1 en $[t_0, t_1]$, $k = 1$.

El objetivo en cada intervalo k es determinar los parámetros a_{ki} que proporcionan el *spline* cuadrático interpolador (definido en cada intervalo $[t_{k-1}, t_k]$ por (4.3)) de la solución discreta del **PVI (2.6)**, es decir, obtener un total de 3 parámetros, **los cuales son los coeficientes del trazador en la base de polinomios que también define (4.3)**. El tramo S_1 del *spline* cuadrático S correspondiente al intervalo inicial $[t_0, t_1]$, junto con su derivada primera S'_1 , viene dado por (4.7):

$$S_1(x) = a_{10} + a_{11}(x - t_0) + a_{12}(x - t_0)^2 \quad \text{a)}$$

$$S'_1(x) = a_{11} + 2a_{12}(x - t_0) \quad \text{b)}$$

Ecuación (4.7)

Dado que las condiciones de continuidad sólo se imponen en los **nodos intermedios** $\{t_k\}_{k=1}^{n-1}$ del dominio discretizado con (2.11), no se necesita considerarlas ahora. Se deberán imponer pues, además de las dos condiciones de interpolación en t_0 y t_1 , la **condición adicional citada en la Tabla 4.2 (2ª condición inicial del PVI)** para calcular los parámetros que definen el tramo S_1 del *spline* cuadrático S . Expuesto lo anterior, se tiene (4.8):

$$S_1(t_0) = y(t_0) = y_0 \rightarrow \boxed{a_{10} = y_0} \rightarrow \text{Interp. en } t_0 \text{ y } 1^{\text{a}} \text{ cond. inicial PVI} \quad \text{a)}$$

$$S'_1(t_0) = y'(t_0) = z_0 \rightarrow \boxed{a_{11} = z_0} \rightarrow 2^{\text{a}} \text{ condición inicial PVI} \quad \text{b)}$$

Ecuación (4.8)

El parámetro a_{12} se determina mediante la imposición de una segunda condición de interpolación en el nodo extremo t_1 (4.9):

$$S_1(t_1) = S_1(t_0 + h_1) = a_{10} + h_1 a_{11} + h_1^2 a_{12} = y_1 \rightarrow \boxed{a_{12} = \frac{1}{h_1^2} (y_1 - a_{10} - h_1 a_{11})}$$

Ecuación (4.9)

Por tanto, queda determinado el tramo S_1 del *spline* cuadrático interpolador S de la solución discreta del **PVI (2.6)** correspondiente al primer intervalo del dominio $[t_0, t_1]$. Sustituyendo los valores conocidos de a_{10} y a_{11} dados por (4.8) en (4.9), se obtiene la expresión definitiva para a_{12} (4.10):

$$\begin{cases} a_{10} = y_0 \\ a_{11} = z_0 \end{cases} \rightarrow \boxed{a_{12} = \frac{1}{h_1^2} (y_1 - y_0 - h_1 z_0)}$$

Ecuación (4.10)

4.3.3. Resolución de un intervalo genérico. Obtención de S_k en $[t_{k-1}, t_k]$, $k = 2, 3, \dots, n$.

El esquema progresivo planteado de determinación de parámetros a_{ki} es fácilmente generalizable a todos los intervalos que satisfacen $k = 2, 3, \dots, n$, es decir, si $x \in [t_1, t_n = b]$. El tramo S_k del *spline* cuadrático S , junto con su derivada primera S'_k , viene dado por (4.11):

$$S_k(x) = a_{k0} + a_{k1}(x - t_{k-1}) + a_{k2}(x - t_{k-1})^2 \quad \text{a)}$$

$$S'_k(x) = a_{k1} + 2a_{k2}(x - t_{k-1}) \quad \text{b)}$$

Ecuación (4.11)

Planteando sistemáticamente las condiciones de continuidad en el nodo inicial t_{k-1} y la condición de interpolación en t_k , se tienen (4.12) y (4.13):

$$S_{k-1}(t_{k-1}) = S_k(t_{k-1}) \rightarrow \boxed{a_{k0} = S_{k-1}(t_{k-1})} \quad \text{a)}$$

$$S'_{k-1}(t_{k-1}) = S'_k(t_{k-1}) \rightarrow \boxed{a_{k1} = S'_{k-1}(t_{k-1})} \quad \text{b)}$$

$$S_k(t_k) = y_k \rightarrow a_{k0} + h_k a_{k1} + h_k^2 a_{k2} = y_k \rightarrow \text{Interp. en } t_k \quad \text{c)}$$

Ecuación (4.12)

$$\boxed{a_{k2} = \frac{1}{h_k^2} (y_k - a_{k0} - h_k a_{k1})}$$

Ecuación (4.13)

Una vez resuelto el intervalo k , surgen las siguientes expresiones genéricas (4.14) que permiten calcular los parámetros del intervalo sucesivo $k + 1$. **Recursivamente**, para $k = 1, 2, \dots, n - 1$ (se han incluido los parámetros correspondientes al intervalo inicial), se tiene:

$$\begin{cases} a_{(k+1)0} = a_{k0} + h_k a_{k1} + h_k^2 a_{k2} \\ a_{(k+1)1} = a_{k1} + 2h_k a_{k2} \\ a_{(k+1)2} = y_{k+1} - a_{(k+1)0} - h_{k+1} a_{(k+1)1} \rightarrow h_{k+1} = t_{k+1} - t_k \end{cases}$$

Ecuación (4.14)

4.3.4. Resumen del procedimiento.

El cuadro sinóptico correspondiente a la **Tabla 4.3** resume las condiciones impuestas sobre el *spline* cuadrático (4.4) expuestas con anterioridad, en concordancia con las **Tablas 4.1** y **4.2**.

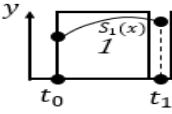
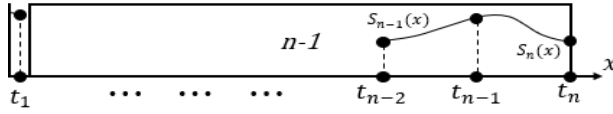
		<i>Distribución de intervalos</i>	
		<i>Inicial</i> (secc. 4.3.2)	<i>Genérico (secc. 4.3.3)</i>
TIPOS DE CONDICIONES			
Continuidad		-	$2(n - 1) \rightarrow (4.12.a), (4.12.b)$
Interpolación		$2 \rightarrow (4.8.a)$ y (4.9)	$n - 1 \rightarrow (4.12.c)$
Adicionales		$1 \rightarrow (4.8.b)$	-
TOTALES		$3 + 3(n - 1) = \boxed{3n}$	

Tabla 4.3. Cuadro sinóptico resumen. *Splines* cuadráticos interpoladores.

En el **ANEXO II** de esta memoria está disponible la función “*spline2int*”, la cual es una implementación en **MATLAB** de la metodología expuesta en estas secciones para poder evaluar los $3n$ parámetros (coeficientes) a_{ki} que

determinan unívocamente el *spline* cuadrático interpolador de la solución discreta del *PVI* (2.6) en los nodos de malla (2.11) y en la base de (4.3).

4.4. Estrategia empleando *splines* cúbicos interpoladores.

Una alternativa al empleo de *splines* cuadráticos para interpolar las soluciones discretas del *PVI* (2.6) en los nodos de malla (2.11) son los *splines* cúbicos, es decir, aquellos con un grado $p \leq 3$, a lo sumo una unidad mayor que la familia de trazadores polinomiales citada en la sección 4.3. En concordancia con (4.2), un *spline* cúbico S viene descrito en cada intervalo $[t_{k-1}, t_k]$ por un polinomio S_k , de grado a lo sumo 3 (4.15):

$$S_k(x) = \sum_{i=0}^{p=3} a_{ki} (x - t_{k-1})^i, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n, \quad a_{ki} \in \mathbb{R}$$

Ecuación (4.15)

Desarrollando la expresión anterior y volviendo a tener en cuenta la existencia de $n + 1$ nodos ordenados crecientemente según (2.11), se tiene (4.16):

$$S(x) = \begin{cases} S_1(x) = a_{10} + a_{11}(x - t_0) + a_{12}(x - t_0)^2 + a_{13}(x - t_0)^3, & \text{si } x \in [t_0, t_1] \\ S_2(x) = a_{20} + a_{21}(x - t_1) + a_{22}(x - t_1)^2 + a_{23}(x - t_1)^3, & \text{si } x \in [t_1, t_2] \\ \dots \\ S_n(x) = a_{n0} + a_{n1}(x - t_{n-1}) + a_{n2}(x - t_{n-1})^2 + a_{n3}(x - t_{n-1})^3, & \text{si } x \in [t_{n-1}, t_n] \end{cases}$$

Ecuación (4.16)

4.4.1. Justificación del empleo de *splines* cúbicos.

Utilizar *splines* cúbicos en la interpolación de las soluciones discretas del *PVI* (2.6) es una alternativa a los *splines* cuadráticos cuando aquellos puedan resultar no del todo satisfactorios, gracias a que ahora se proporcionan aproximaciones de clase C^2 a la solución continua. Las expresiones (4.17) y (4.18) muestran las derivadas en cada intervalo $[t_{k-1}, t_k]$.

$$S'(x) = \begin{cases} S'_1(x) = a_{11} + 2a_{12}(x - t_0) + 3a_{13}(x - t_0)^2, & \text{si } x \in [t_0, t_1] \\ S'_2(x) = a_{21} + 2a_{22}(x - t_1) + 3a_{23}(x - t_1)^2, & \text{si } x \in [t_1, t_2] \\ \dots \\ S'_n(x) = a_{n1} + 2a_{n2}(x - t_{n-1}) + 3a_{n3}(x - t_{n-1})^2, & \text{si } x \in [t_{n-1}, t_n] \end{cases}$$

Ecuación (4.17)

$$S''(x) = \begin{cases} S''_1(x) = 2a_{12} + 6a_{13}(x - t_0), & \text{si } x \in [t_0, t_1] \\ S''_2(x) = 2a_{22} + 6a_{23}(x - t_1), & \text{si } x \in [t_1, t_2] \\ \dots \\ S''_n(x) = 2a_{n2} + 6a_{n3}(x - t_{n-1}), & \text{si } x \in [t_{n-1}, t_n] \end{cases}$$

Ecuación (4.18)

Lógicamente, en los extremos de los intervalos, las derivadas habrán de ser consideradas derivadas laterales (por la derecha en los extremos iniciales y por la izquierda en los finales). Ahora se tiene una segunda derivada continua en todo el dominio del **PVI (2.6)**, al haber impuesto continuidad C^2 del *spline*, lo cual implica una **mejora respecto a lo comentado anteriormente a la hora de aproximar aceleraciones en problemas cuya importancia sea relevante (ligadas a la segunda derivada)**.

En concordancia con la **Tabla 4.1**, la determinación total del *spline* de tercer grado se consigue mediante la imposición de las condiciones cuya naturaleza se indican en ella. Además, se han de imponer dos condiciones adicionales. Por tanto, considérese la **Tabla 4.4**:

<i>Tipo</i>	<i>Número</i>	<i>Forma</i>
Continuidad	$3(n - 1)$	$S, S' \text{ y } S'' \text{ continuas en } \{t_{k-1}\}_{k=2}^n$ $\begin{cases} S_{k-1}(t_{k-1}) = S_k(t_{k-1}) \rightarrow n - 1 \\ S'_{k-1}(t_{k-1}) = S'_k(t_{k-1}) \rightarrow n - 1 \\ S''_{k-1}(t_{k-1}) = S''_k(t_{k-1}) \rightarrow n - 1 \end{cases}$
Interpolación	$n + 1$	$S(t_k) = y_k$ ($k = 0, 1^{\text{a}}$ Cond. Inicial PVI(2.6))
Adicionales		$\rightarrow 2^{\text{a}}$ Cond. Inicial PVI(2.6) : $y'(a) = y'(t_0) = z_0$ \rightarrow Colocación en el nodo inicial $t_0 = a$: $S''(t_0) = f(t_0, y_0, z_0)$
TOTALES		$4n$

Tabla 4.4. Resumen de condiciones a imponer para determinar unívocamente un *spline* cúbico interpolador.

4.4.2. Resolución del intervalo inicial. Obtención de S_1 en $[t_0, t_1]$, $k = 1$.

Para determinar los parámetros (coeficientes) a_{1i} que proporcionan el primer tramo S_1 del *spline* cúbico interpolador S de la solución discreta del **PVI (2.6)** se han de imponer las condiciones citadas en la **Tabla 4.4**. En total, el número de parámetros a calcular es 4. El tramo S_1 viene dado por **(4.19)**, junto con sus derivadas primera (S'_1) y segunda (S''_1):

$$S_1(x) = a_{10} + a_{11}(x - t_0) + a_{12}(x - t_0)^2 + a_{13}(x - t_0)^3 \quad \text{a)}$$

$$S'_1(x) = a_{11} + 2a_{12}(x - t_0) + 3a_{13}(x - t_0)^2 \quad \text{b)}$$

$$S''_1(x) = 2a_{12} + 6a_{13}(x - t_0) \quad \text{c)}$$

Ecuación **(4.19)**

Procediendo con la imposición de condiciones citada, se tiene **(4.20)**:

$$S_1(t_0) = y(t_0) = y_0 \rightarrow \boxed{a_{10} = y_0} \rightarrow \text{Interp. en } t_0 \text{ y 1ª cond. inicial PVI} \quad a)$$

$$S_1'(t_0) = y'(t_0) = z_0 \rightarrow \boxed{a_{11} = z_0} \rightarrow \text{2ª condición inicial PVI} \quad b)$$

$$S_1''(t_0) = f(t_0, y(t_0), y'(t_0)) = f(t_0, y_0, z_0) \\ \rightarrow \boxed{a_{12} = \frac{1}{2}f(t_0, y_0, z_0)} \rightarrow \text{Colocación en el nodo inicial} \quad c)$$

Ecuación (4.20)

El parámetro a_{13} restante se determina mediante una segunda condición de interpolación (4.21) en el nodo extremo t_1 :

$$S_1(t_1) = S_1(t_0 + h_1) = a_{10} + h_1 a_{11} + h_1^2 a_{12} + h_1^3 a_{13} = y_1$$

$$\rightarrow \boxed{a_{13} = \frac{1}{h_1^3}(y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12})}$$

Ecuación (4.21)

Por tanto, queda determinado el tramo del *spline* cúbico interpolador de la solución discreta del **PVI (2.6)** correspondiente al primer intervalo del dominio $[t_0, t_1]$. Sustituyendo los valores conocidos de a_{10} , a_{11} y a_{12} dados por (4.20) en (4.21), surge la expresión definitiva para a_{13} (4.22):

$$\left\{ \begin{array}{l} a_{10} = y_0 \\ a_{11} = z_0 \\ a_{12} = \frac{1}{2}f(t_0, y_0, z_0) \end{array} \right. \rightarrow \boxed{a_{13} = \frac{1}{h_1^3} \left(y_1 - y_0 - h_1 z_0 - \frac{h_1^2}{2} f(t_0, y_0, z_0) \right)}$$

Ecuación (4.22)

4.4.3. Resolución de un intervalo genérico. Obtención de S_k en $[t_{k-1}, t_k]$, $k = 2, 3, \dots, n$.

Al igual que en la sección 4.3.3, el esquema progresivo planteado para la determinación de parámetros (coeficientes) a_{ki} se puede generalizar a todos los intervalos contenidos en $[t_1, t_n = b]$. El tramo S_k del *spline* cúbico S , junto con sus derivadas primera (S_k') y segunda (S_k''), viene dado por (4.23):

$$S_k(x) = a_{k0} + a_{k1}(x - t_{k-1}) + a_{k2}(x - t_{k-1})^2 + a_{k3}(x - t_{k-1})^3 \quad a)$$

$$S_k'(x) = a_{k1} + 2a_{k2}(x - t_{k-1}) + 3a_{k3}(x - t_{k-1})^2 \quad b)$$

$$S_k''(x) = 2a_{k2} + 6a_{k3}(x - t_{k-1}) \quad c)$$

Ecuación (4.23)

Planteando sistemáticamente las condiciones de continuidad en el nodo inicial t_{k-1} y la condición de interpolación en t_k , se tienen (4.24) y (4.25):

$$S_{k-1}(t_{k-1}) = S_k(t_{k-1}) \rightarrow \boxed{a_{k0} = S_{k-1}(t_{k-1})} \quad \text{a)}$$

$$S'_{k-1}(t_{k-1}) = S'_k(t_{k-1}) \rightarrow \boxed{a_{k1} = S'_{k-1}(t_{k-1})} \quad \text{b)}$$

$$S''_{k-1}(t_{k-1}) = S''_k(t_{k-1}) \rightarrow \boxed{a_{k2} = \frac{1}{2} S''_{k-1}(t_{k-1})} \quad \text{c)}$$

$$S_k(t_k) = y_k \rightarrow a_{k0} + h_k a_{k1} + h_k^2 a_{k2} + h_k^3 a_{k3} = y_k \rightarrow \text{Interp. en } t_k \quad \text{d)}$$

Ecuación (4.24)

$$\boxed{a_{k3} = \frac{1}{h_k^3} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2})}$$

Ecuación (4.25)

Además, planteando nuevamente un **esquema recursivo**, surgen las siguientes expresiones genéricas (4.26) que permiten calcular los parámetros del intervalo sucesivo $k + 1$.

$$\left\{ \begin{array}{l} a_{(k+1)0} = a_{k0} + h_k a_{k1} + h_k^2 a_{k2} + h_k^3 a_{k3} \\ a_{(k+1)1} = a_{k1} + 2h_k a_{k2} + 3h_k^2 a_{k3} \\ a_{(k+1)2} = a_{k2} + 3h_k a_{k3} \\ a_{(k+1)3} = y_{k+1} - a_{(k+1)0} - h_{k+1} a_{(k+1)1} - h_{k+1}^2 a_{(k+1)2} \rightarrow h_{k+1} = t_{k+1} - t_k \end{array} \right.$$

Ecuación (4.26)

4.4.4. Resumen del procedimiento.

Préstese atención al cuadro sinóptico de la **Tabla 4.5**, el cual resume las condiciones impuestas sobre el **spline cúbico (4.17)** y desarrolladas en los anteriores apartados.

Distribución de intervalos

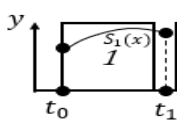
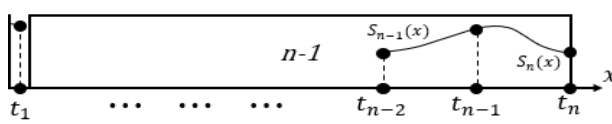
	<i>Inicial</i> (secc. 4.4.2)	<i>Genérico</i> (secc. 4.4.3)
TIPOS DE CONDICIONES		
Continuidad	-	$3(n - 1) \rightarrow (4.24.a), (4.24.b) \text{ y } (4.24.c)$
Interpolación	$2 \rightarrow (4.20.a) \text{ y } (4.21)$	$n - 1 \rightarrow (4.24.d)$
Adicionales	$2 \rightarrow (4.20.b) \text{ y } (4.20.c)$	-
TOTALES		$4 + 4(n - 1) = \boxed{4n}$

Tabla 4.5. Cuadro sinóptico resumen. Splines cúbicos interpoladores.



En el **ANEXO II** se dispone de la función “**spline3int**”, la cual es una implementación en **MATLAB** de la metodología expuesta para poder evaluar los $4n$ parámetros a_{ki} que determinan unívocamente el *spline* cúbico interpolador de la solución discreta del **PVI (2.6)** en los nodos de malla **(2.11)**. **Se recuerda una vez más que dichos parámetros coinciden con los coeficientes del spline en cada tramo a determinar y en la base de polinomios definida en (4.15)**, al igual que ocurría con los correspondientes a los *splines* cuadráticos de la sección **4.3**.





CAPÍTULO 5. Estrategias de interpolación de soluciones de *PVIs* empleando *pseudo-* *splines*.



5.1. Introducción. Definición de *pseudo-spline*.

Si el lector ha consultado con atención el **Capítulo 4** algunas de las referencias bibliográficas que se proponen en él, habrá comprendido la enorme utilidad que los *splines* a día de hoy tienen no sólo en interpolación y aproximación polinomial (razón de ser de su uso en este trabajo), sino en muchas otras áreas de la ciencia también citadas y documentadas bibliográficamente.

No obstante, el uso de trazadores polinomiales no queda para nada restringido a los *splines*. De hecho, éstos presentan algunas restricciones en cuanto a su definición formal, pues para ser considerados como tal ha de imponerse una continuidad que obedezca a $S \in C_{[a,b]}^{p-1}$ (p es el grado máximo), considerando la forma y base en cada intervalo del dominio (2.11) como la dada por (4.2). La repercusión que puede tener lo anterior al emplearlos en las estrategias de interpolación de las soluciones de **PVIs** que están formulados en términos de **EDOs** de 2º orden (**Capítulo 2**) es significativa, pues no siempre se consiguen buenos resultados (aproximaciones a velocidades, aceleraciones...) empleando las condiciones impuestas por la **Tabla 4.1**.

Por lo tanto, en este capítulo se mantiene la formalidad de los trazadores polinomiales y el uso para el que son concebidos en el trabajo, pero se realiza una generalización de los *splines* a otra familia conocida como *pseudo-splines*. Un *pseudo-spline* no impone continuidad $S \in C_{[a,b]}^{p-1}$, sino que ésta ha de ser ajustada al problema en cuestión, permitiendo más libertad a la hora de adecuar la aproximación a la solución a interpolar. Por ello, de ahora en adelante, se denotará un trazador con esas características como \tilde{S} , para poder diferenciarlo claramente de los *splines* S . Algunos esfuerzos investigadores sobre estos trazadores se indican en las referencias [43-45].

Un *pseudo-spline* de grado a lo sumo p se define en cada intervalo $[t_{k-1}, t_k]$ por un polinomio que toma la forma de (5.1), expresión similar a (4.2) pero con la nueva notación del trazador empleado ahora introducida:

$$\tilde{S}_k(x) = \sum_{i=0}^p a_{ki}(x - t_{k-1})^i, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n, \quad a_{ki} \in \mathbb{R}$$

Ecuación (5.1)

Las distintas estrategias de interpolación abordadas en este capítulo (e implementadas en **MATLAB** en el **ANEXO IV**) pasan por el **empleo de pseudo-splines cuárticos interpoladores** de las soluciones de los **PVI** de 2º orden del **Capítulo 2** (con estrategias diversas para cada uno de los **PVI** tipo presentados), así como otras alternativas basadas en **polinomios osculadores (pseudo-splines cúbicos y quínticos)**. En resumen, se tiene la estructura general marcada por la **Tabla 5.1**, la cual ha decidido aportarse para que el lector pueda situar correctamente cada estrategia en su lugar y contexto adecuado.

<i>Estrategia</i>	<i>Variantes particulares</i>	<i>Sección</i>
<i>Pseudo-splines</i> <i>cuárticos (5.2)</i>	Colocación	5.2.1
	Interpolación intermedia y colocación inicial	5.2.2
	Colocación intermedia y colocación en el nodo inicial	5.2.3
<i>Polinomios</i> <i>osculadores</i> <i>(5.3)</i>	<i>Pseudo-splines</i> cúbicos	5.3.2
	<i>Pseudo-splines</i> quínticos	5.3.3

Tabla 5.1. Estructura del **Capítulo 5**. *Pseudo-splines* interpoladores.

5.2. *Pseudo-splines* cuárticos interpoladores.

Un *pseudo-spline* cuártico toma la forma dada por (5.1), con la particularidad de que el grado toma valores $p \leq 4$. Desarrollando dicha expresión particularizada en cada intervalo de los n definidos por los $n + 1$ nodos de (2.11), surge (5.2):

$$\tilde{S}(x) = \begin{cases} \tilde{S}_1(x) = a_{10} + a_{11}(x - t_0) + a_{12}(x - t_0)^2 + a_{13}(x - t_0)^3 + a_{14}(x - t_0)^4, & \text{si } x \in [t_0, t_1] \\ \tilde{S}_2(x) = a_{20} + a_{21}(x - t_1) + a_{22}(x - t_1)^2 + a_{23}(x - t_1)^3 + a_{24}(x - t_1)^4, & \text{si } x \in [t_1, t_2] \\ \dots \\ \tilde{S}_n(x) = a_{n0} + a_{n1}(x - t_{n-1}) + a_{n2}(x - t_{n-1})^2 + a_{n3}(x - t_{n-1})^3 + a_{n4}(x - t_{n-1})^4, & \text{si } x \in [t_{n-1}, t_n] \end{cases}$$

Ecuación (5.2)

Asimismo, la primera (\tilde{S}'_k) y segunda (\tilde{S}''_k) derivada en cada intervalo $[t_{k-1}, t_k]$ vienen dadas por (5.3) y (5.4). En dichas expresiones también se insertan las derivadas del *pseudo-spline* en todo el dominio ((5.3.b) y (5.4.b)). Es importante considerar en los extremos de cada intervalo derivadas estrictamente laterales (por la derecha en los extremos iniciales t_{k-1} y por la izquierda en los finales t_k).

$$\tilde{S}'_k(x) = \sum_{i=1}^{p=4} i \cdot a_{ki}(x - t_{k-1})^{i-1}, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n, \quad a_{ki} \in \mathbb{R} \quad \text{a)}$$

$$\tilde{S}'(x) = \begin{cases} \tilde{S}'_1(x) = a_{11} + 2a_{12}(x - t_0) + 3a_{13}(x - t_0)^2 + 4a_{14}(x - t_0)^3, & \text{si } x \in [t_0, t_1] \\ \tilde{S}'_2(x) = a_{21} + 2a_{22}(x - t_1) + 3a_{23}(x - t_1)^2 + 4a_{24}(x - t_1)^3, & \text{si } x \in [t_1, t_2] \\ \dots \\ \tilde{S}'_n(x) = a_{n1} + 2a_{n2}(x - t_{n-1}) + 3a_{n3}(x - t_{n-1})^2 + 4a_{n4}(x - t_{n-1})^3, & \text{si } x \in [t_{n-1}, t_n] \end{cases} \quad \text{b)}$$

Ecuación (5.3)

$$\tilde{S}''_k(x) = \sum_{i=2}^{p=4} i(i-1) \cdot a_{ki}(x - t_{k-1})^{i-2}, \quad \text{si } x \in [t_{k-1}, t_k], \quad k = 1, 2, \dots, n, \quad a_{ki} \in \mathbb{R} \quad \text{a)}$$

$$\tilde{S}''(x) = \begin{cases} \tilde{S}_1''(x) = 2a_{12} + 6a_{13}(x - t_0) + 12a_{14}(x - t_0)^2, & \text{si } x \in [t_0, t_1] \\ \tilde{S}_2''(x) = 2a_{22} + 6a_{23}(x - t_1) + 12a_{24}(x - t_1)^2, & \text{si } x \in [t_1, t_2] \\ \dots \\ \tilde{S}_n''(x) = 2a_{n2} + 6a_{n3}(x - t_{n-1}) + 12a_{n4}(x - t_{n-1})^2, & \text{si } x \in [t_{n-1}, t_n] \end{cases} \quad b)$$

Ecuación (5.4)

Para determinar unívocamente dicho *pseudo-spline* cuártico \tilde{S} , se imponen **condiciones de continuidad** ($\tilde{S} \in C_{[a,b]}^2$) y de **interpolación** ($\tilde{S}(t_k) = y_k$), así como condiciones adicionales dependientes de las estrategias de resolución expuestas con detalle en los sucesivos apartados y citadas brevemente en la **Tabla 5.1**. El número total de parámetros (coeficientes) a calcular es $5n$. A su vez, el **número total de las condiciones adicionales asciende a $n + 2$ (ligadas a parámetros libres)**, quedándose en $n + 1$ tras aplicar la 2ª condición inicial del **PVI** en términos de la primera derivada $\tilde{S}'_1(t_0) = \tilde{S}'_1(a) = z_0$. Resumiendo, se tiene la **Tabla 5.2**:

Tipo	Número	Forma
Continuidad	$3(n - 1)$	S, S' y S'' continuas en $\{t_{k-1}\}_{k=2}^n$ $\begin{cases} S_{k-1}(t_{k-1}) = S_k(t_{k-1}) \rightarrow n - 1 \\ S'_{k-1}(t_{k-1}) = S'_k(t_{k-1}) \rightarrow n - 1 \\ S''_{k-1}(t_{k-1}) = S''_k(t_{k-1}) \rightarrow n - 1 \end{cases}$
Interpolación	$n + 1$	$S(t_k) = y_k$ $(k = 0, 1^{\text{ª}} \text{ Cond. Inicial PVI(2.6)})$
Adicionales	$n + 2$	Diversas dependiendo de la estrategia
TOTALES		$5n$

Tabla 5.2. Resumen de condiciones a imponer para determinar unívocamente un *pseudo-spline* cuártico interpolador.

A continuación, se presenta la metodología apropiada para cada estrategia de imposición de las $n + 1$ condiciones adicionales (restada ya la 2ª condición inicial), particularizada cuando procede a cada uno de los tipos de **PVI** expuestos en el **Capítulo 2**.

5.2.1. Condiciones adicionales. Estrategia de colocación.

Una primera opción que se puede utilizar es el **método de colocación en los nodos de (2.11)**, ya que al existir un total de $n + 1$, se consigue la determinación unívoca del *pseudo-spline* cuártico que interpola a la solución discreta del **PVI** en cuestión, combinando las condiciones adicionales con las de continuidad e interpolación. Este método consiste en **imponer que el trazador empleado \tilde{S} satisfaga la EDO del problema propuesto en cada nodo t_k** . Por tanto, y empleando el **PVI** completo (2.6) se tiene (5.5):

$$y''(x) = f(x, y(x), y'(x)) \rightarrow \tilde{S}''(t_k) = f(t_k, \tilde{S}(t_k), \tilde{S}'(t_k))$$

Ecuación (5.5)

Puede comprobarse fácilmente que, para $k = 0$, se tiene la **condición de colocación expresada en términos de las condiciones iniciales** del PVI abordado (5.6), tal y como ocurría con el *spline* cúbico interpolador (sección 4.4 del Capítulo 4).

$$\tilde{S}_1''(t_0) = f(t_0, y(t_0), y'(t_0)) = f(t_0, y_0, z_0)$$

Ecuación (5.6)

5.2.1.1. Particularización al caso lineal.

Si se utiliza la particularización de (2.6) dada por el PVI lineal (2.7), se tienen las expresiones (5.7) y (5.8), las cuales son respectivamente particularizaciones lineales de (5.5) y (5.6):

$$y''(x) = p(x)y'(x) + q(x)y(x) + r(x) \rightarrow \tilde{S}''(t_k) = p_k\tilde{S}'(t_k) + q_k\tilde{S}(t_k) + r_k$$

Ecuación (5.7)

$$\tilde{S}_1''(t_0) = p_0\tilde{S}_1'(t_0) + q_0\tilde{S}_1(t_0) + r_0 = p_0z_0 + q_0y_0 + r_0$$

Ecuación (5.8)

Posteriormente se verá que también existe una particularización especial para problemas independientes de la primera derivada y' del tipo (2.8). Nótese que se ha introducido la notación simplificadora indicada en (5.9), empleada de ahora en adelante cuando proceda en los desarrollos teóricos adjuntos.

$$p_k = p(t_k), \quad q_k = q(t_k), \quad r_k = r(t_k), \quad k = 0, 1, \dots, n$$

Ecuación (5.9)

5.2.2. Condiciones adicionales. Estrategia de interpolación intermedia y colocación en el nodo inicial $t_0 = a$.

Esta segunda opción arroja la posibilidad de interpolar en nodos intermedios de los intervalos generados según las soluciones discretas y_k . Se consideran nodos intermedios $t_{k-1/2}$ en cada intervalo y aproximaciones a la solución en los mismos, obtenidos con el mismo método (o solución exacta si existiese) que proporciona y_k . Además, se utiliza una condición de colocación en el nodo inicial ya abordada en (5.6). La nueva solución discreta en los nodos intermedios (5.10) toma la forma siguiente:

$$\left\{ t_{k-1/2} : k = 1, 2, \dots, n \right\}, \quad \left\{ y_{k-1/2} : k = 1, 2, \dots, n \right\} \rightarrow y_{k-1/2} \cong y\left(t_{k-1/2}\right)$$

Ecuación (5.10)

En otras palabras, ahora se utiliza un paso mitad respecto del definido en (2.12), denotado como $1/2 h_k = t_{k-1/2} - t_k$ y empleado para situar (5.10), de tal forma que ésta se sigue expresando en función de las posiciones de (2.11) y el paso h_k , manteniendo el acoplamiento entre ambas. Las nuevas **condiciones de interpolación intermedia surgidas** se expresan en (5.11), las cuales deben combinarse con (5.6) para completar la estrategia.

$$\tilde{S}_k \left(t_{k-\frac{1}{2}} \right) = y_{k-\frac{1}{2}}, \quad k = 1, 2, \dots, n$$

Ecuación (5.11)

5.2.3. Condiciones adicionales. Estrategia de colocación intermedia y colocación en el nodo inicial $t_0 = a$.

Manteniendo ahora la nueva sucesión de nodos intermedios dada en (5.10), se imponen condiciones de **colocación sobre la ecuación diferencial en cada nodo $t_{k-\frac{1}{2}}$** . Se obtiene (5.12):

$$\tilde{S}_k'' \left(t_{k-\frac{1}{2}} \right) = f \left(t_{k-\frac{1}{2}}, \tilde{S} \left(t_{k-\frac{1}{2}} \right), \tilde{S}' \left(t_{k-\frac{1}{2}} \right) \right), \quad k = 1, 2, \dots, n$$

Ecuación (5.12)

La expresión anterior, en conjunto con la colocación inicial de (5.6), completa las $n + 1$ condiciones adicionales disponibles y necesarias para determinar unívocamente el *pseudo-spline* cuártico por esta estrategia.

5.2.3.1 Particularización al caso lineal.

Utilizando el *PVI* lineal (2.7), se tiene (5.13) como forma particularizada lineal de (5.12).

$$\tilde{S}_k'' \left(t_{k-\frac{1}{2}} \right) = p_{k-\frac{1}{2}} \tilde{S}_k' \left(t_{k-\frac{1}{2}} \right) + q_{k-\frac{1}{2}} \tilde{S}_k \left(t_{k-\frac{1}{2}} \right) + r_{k-\frac{1}{2}}, \quad k = 1, 2, \dots, n$$

Ecuación (5.13)

Para completar la imposición de condiciones, se añade (5.8).

5.2.4. Resolución del intervalo inicial. Obtención de \tilde{S}_1 en $[t_0, t_1]$, $k = 1$.

El objetivo en **cada intervalo $k = 1, \dots, n$** es calcular los parámetros a_{ki} que determinan unívocamente el *pseudo-spline* cuártico interpolador de la solución del *PVI* y_k , es decir, obtener un total de 5 parámetros (coeficientes). Dado que las condiciones de continuidad se definen en **los nodos intermedios del dominio y se imponen en el nodo inicial del intervalo en cuestión**, no es necesario considerarlas al determinar los coeficientes correspondientes al primer intervalo $[t_0, t_1]$. Por tanto, para encontrar los parámetros que definen



el trazador \tilde{S}_1 , se imponen las **condiciones iniciales del PVI**, junto con otras **condiciones dependientes de la estrategia utilizada en la resolución** (ver **5.2.1**, **5.2.2** y **5.2.3**). El tramo del *pseudo-spline* cuártico correspondiente al primer intervalo viene descrito, junto con sus derivadas primera (\tilde{S}'_1) y segunda (\tilde{S}''_1), por **(5.14)** (forma desarrollada de **(5.1)** para $k = 1$).

$$\tilde{S}_1(x) = a_{10} + a_{11}(x - t_0) + a_{12}(x - t_0)^2 + a_{13}(x - t_0)^3 + a_{14}(x - t_0)^4 \quad \text{a)}$$

$$\tilde{S}'_1(x) = a_{11} + 2a_{12}(x - t_0) + 3a_{13}(x - t_0)^2 + 4a_{14}(x - t_0)^3 \quad \text{b)}$$

$$\tilde{S}''_1(x) = 2a_{12} + 6a_{13}(x - t_0) + 12a_{14}(x - t_0)^2 \quad \text{c)}$$

Ecuación **(5.14)**

Las condiciones comunes a todas las estrategias particularizadas al intervalo inicial (con $t_0 = a$) vienen dadas por **(5.15)**, y han sido planteadas en concordancia con la **Tabla 5.2**.

$$\tilde{S}_1(t_0) = y(t_0) = y_0 \rightarrow \boxed{a_{10} = y_0} \rightarrow \text{Interp. en } t_0 \text{ (1ª cond. inicial PVI)} \quad \text{a)}$$

$$\tilde{S}'_1(t_0) = y'(t_0) = z_0 \rightarrow \boxed{a_{11} = z_0} \rightarrow \text{2ª Condición inicial PVI} \quad \text{b)}$$

$$\tilde{S}''_1(t_0) = f(t_0, y(t_0), y'(t_0)) = f(t_0, y_0, z_0) \rightarrow \boxed{a_{12} = \frac{1}{2}f(t_0, y_0, z_0)} \rightarrow \text{Coloc. inicial} \quad \text{c)}$$

$$\tilde{S}_1(t_1) = \tilde{S}_1(t_0 + h_1) = a_{10} + h_1 a_{11} + h_1^2 a_{12} + h_1^3 a_{13} + h_1^4 a_{14} = y_1 \rightarrow \text{Interp. en } t_1 \quad \text{d)}$$

Ecuación **(5.15)**

Nótese que **(5.15.c)** corresponde a **(5.6)** y que el paso toma un valor $h_1 = t_1 - t_0$. Es necesario determinar los parámetros restantes a_{13} y a_{14} . Reorganizando **(5.15.d)**, se tiene **(5.16)**:

$$h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12}$$

Ecuación **(5.16)**

Puede particularizarse la ecuación anterior a los valores conocidos de a_{10} , a_{11} y a_{12} para obtener **(5.17)**:

$$h_1^3 a_{13} + h_1^4 a_{14} = y_1 - y_0 - h_1 z_0 - \frac{1}{2} h_1^2 f(t_0, y_0, z_0)$$

Ecuación **(5.17)**

Se precisa una ecuación más para poder calcular a_{13} y a_{14} . Dicho cálculo se efectúa mediante la resolución de un **Sistema 2x2 Compatible y Determinado**. Las formas de obtener la 2ª ecuación necesaria para determinar unívocamente el tramo \tilde{S}_1 del *pseudo-spline* cuártico \tilde{S} son diversas, pues **ya se conocen**

distintas estrategias de imposición de la condición adicional restante correspondiente al intervalo inicial tras haber impuesto (5.15) (secciones 5.2.1, 5.2.2 y 5.2.3). La forma de la expresión resultante también depende del tipo de problema (PVI general (2.6) o lineal (2.7), con ciertos matices también expuestos posteriormente para el PVI independiente de la primera derivada y' (2.8)), por lo que es de interés realizar una distinción múltiple en el procedimiento a aplicar.

5.2.4.1. Condiciones adicionales. Estrategia de colocación.

La ecuación (5.18) muestra la **condición de colocación** que debe imponerse en t_1 surgida de (5.5), junto con la ya expuesta anteriormente en t_0 (5.6), para determinar completamente el polinomio \tilde{S}_1 de grado $p \leq 4$ correspondiente al primer intervalo.

$$\tilde{S}_1''(t_1) = f(t_1, \tilde{S}_1(t_1), \tilde{S}_1'(t_1))$$

Ecuación (5.18)

La ecuación puede reescribirse en (5.19), haciendo uso de la condición de interpolación impuesta en t_1 (5.15.d):

$$\tilde{S}_1''(t_1) = f(t_1, y_1, \tilde{S}_1'(t_1))$$

Ecuación (5.19)

El **Sistema 2x2** a resolver para el cálculo de a_{13} y a_{14} viene dado por (5.20), y está compuesto por (5.16) y (5.19):

$$\begin{cases} h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\ \tilde{S}_1''(t_1) = f(t_1, y_1, \tilde{S}_1'(t_1)) \end{cases}$$

Ecuación (5.20)

La resolución del sistema anterior se realiza mediante algún método apropiado. En un caso general, cabe la posibilidad de que sea no lineal.

5.2.4.1.1. Particularización a problemas independientes de y' .

Abordar la resolución de (5.20) implica distinguir un caso especial y muy interesante. Este caso concierne al **PVI no lineal e independiente de y'** dado por (2.8), el cual ya se sabe que es un caso particular de (2.6). Reescribiendo (5.20) para definir el **Sistema 2x2** que le corresponde a (2.8) y así calcular a_{13} y a_{14} , se tiene (5.21):

$$\begin{cases} h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\ \tilde{S}_1''(t_1) = f(t_1, y_1) \end{cases}$$

Ecuación (5.21)

La segunda derivada $\tilde{S}_1''(t_1)$ de $\tilde{S}_1(t_1)$ es más que conocida, pues simplemente es necesario particularizar (5.14.c) al nodo extremo del intervalo inicial. Además, obtener $f(t_1, y_1)$ es inmediato. Con estas dos consideraciones, se reescribe (5.21) en (5.22):

$$\begin{cases} h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\ 6h_1 a_{13} + 12h_1^2 a_{14} = -2a_{12} + f(t_1, y_1) \end{cases}$$

Ecuación (5.22)

Se ha obtenido un **Sistema 2x2 lineal** partiendo de un problema no lineal sin hacer ninguna suposición en el desarrollo, algo que para nada suele ser habitual conseguir y que era uno de los objetivos marcados en este trabajo (Capítulo 1). Las expresiones explícitas de a_{13} y a_{14} se obtienen a continuación (5.23):

$$\begin{aligned} a_{14} &= \frac{1}{12h_1^2} (-2a_{12} - 6h_1 a_{13} + f(t_1, y_1)) \rightarrow \\ a_{13} &= \frac{1}{h_1^3} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} - h_1^4 a_{14}) = \\ &= \frac{2}{h_1^3} \left(y_1 - a_{10} - h_1 a_{11} - \frac{5}{6} h_1^2 a_{12} - \frac{1}{12} h_1^2 f(t_1, y_1) \right) \rightarrow \end{aligned}$$

$a_{13} = \frac{2}{h_1^3} \left(-a_{10} - h_1 a_{11} - \frac{5}{6} h_1^2 a_{12} + \left(y_1 - \frac{1}{12} h_1^2 f(t_1, y_1) \right) \right)$	a)
$a_{14} = \frac{1}{h_1^4} \left(a_{10} + h_1 a_{11} + \frac{2}{3} h_1^2 a_{12} + \left(\frac{1}{6} h_1^2 f(t_1, y_1) - y_1 \right) \right)$	b)

Ecuación (5.23)

5.2.4.1.2. Particularización al caso lineal.

Ahora, se asume un PVI de tipo (2.7), y se reescribe (5.18) en (5.24):

$$\tilde{S}_1''(t_1) = p_1 \tilde{S}_1'(t_1) + q_1 \tilde{S}_1(t_1) + r_1$$

Ecuación (5.24)

Utilizando (5.16) conjuntamente con la ecuación anterior, se obtiene un **Sistema 2x2** (5.25), particularización lineal de (5.21).

$$\begin{cases} h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\ A \cdot a_{13} + B \cdot a_{14} = p_1 a_{11} + 2(p_1 h_1 - 1) a_{12} + q_1 y_1 + r_1 \end{cases}$$

$$A = 3h_1(2 - p_1 h_1), \quad B = 4h_1^2(3 - p_1 h_1)$$

Ecuación (5.25)

Se pueden obtener expresiones explícitas para a_{13} y a_{14} . Despejando a_{14} de la primera ecuación de (5.25) y operando, se obtiene (5.26):

$$\begin{aligned}
 a_{14} &= \frac{1}{h_1^4} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} - h_1^3 a_{13}) \rightarrow \\
 &\rightarrow \left(A - \frac{B}{h_1} \right) a_{13} + \frac{B}{h_1^4} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12}) = \\
 &= p_1 a_{11} + 2(p_1 h_1 - 1) a_{12} + q_1 y_1 + r_1 \rightarrow \\
 &\rightarrow \left(\frac{A h_1 - B}{h_1} \right) a_{13} = \frac{B}{h_1^4} a_{10} + \left(p_1 + \frac{B}{h_1^3} \right) a_{11} + \left(2p_1 h_1 - 2 + \frac{B}{h_1^2} \right) a_{12} + \left(q_1 - \frac{B}{h_1^4} \right) y_1 + r_1 \rightarrow
 \end{aligned}$$

$ \begin{aligned} a_{13} &= \left(\frac{h_1}{A h_1 - B} \right) \left(\frac{B}{h_1^4} a_{10} + \left(p_1 + \frac{B}{h_1^3} \right) a_{11} + \left(2p_1 h_1 - 2 + \frac{B}{h_1^2} \right) a_{12} \right. \\ &\quad \left. + \left(q_1 - \frac{B}{h_1^4} \right) y_1 + r_1 \right) \end{aligned} $	a)
$ a_{14} = \frac{1}{h_1^4} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} - h_1^3 a_{13}) $	b)

Ecuación (5.26)

5.2.4.2. Condiciones adicionales. Estrategia de interpolación intermedia y colocación en el nodo inicial $t_0 = a$.

Ahora ha de imponerse la condición de interpolación intermedia en $t_{1/2}$, a partir de la previamente determinada solución nodal $y_{1/2}$ (5.10). Particularizando (5.15.a) con $x = t_{1/2}$ y aplicando (5.11) para $k = 1$, se tiene (5.27):

$$\tilde{s}_1(t_{1/2}) = a_{10} + \frac{h_1}{2} a_{11} + \frac{h_1^2}{4} a_{12} + \frac{h_1^3}{8} a_{13} + \frac{h_1^4}{16} a_{14} = y_{1/2}$$

Ecuación (5.27)

Agrupando en un **Sistema 2x2** la ecuación anterior con (5.16), surge (5.28):

$$\begin{cases}
 h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\
 \frac{h_1^3}{8} a_{13} + \frac{h_1^4}{16} a_{14} = y_{1/2} - a_{10} - \frac{h_1}{2} a_{11} - \frac{h_1^2}{4} a_{12}
 \end{cases}$$

Ecuación (5.28)

Mediante una sencilla estrategia de eliminación, pueden obtenerse expresiones explícitas para a_{13} y a_{14} . Procediendo, se obtiene (5.29):

$$\begin{cases}
 [h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12}] \times (1) \\
 \left[\frac{h_1^3}{8} a_{13} + \frac{h_1^4}{16} a_{14} = y_{1/2} - a_{10} - \frac{h_1}{2} a_{11} - \frac{h_1^2}{4} a_{12} \right] \times (-16) \rightarrow
 \end{cases}$$



$$\begin{aligned} \rightarrow h_1^3 a_{13} + h_1^4 a_{14} - 2h_1^3 a_{13} - h_1^4 a_{14} &= \\ = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} - 16y_{1/2} + 16a_{10} + 8h_1 a_{11} + 4h_1^2 a_{12} &\rightarrow \end{aligned}$$

$a_{13} = \frac{1}{h_1^3} (16y_{1/2} - y_1 - 15a_{10} - 7h_1 a_{11} - 3h_1^2 a_{12})$	a)
$a_{14} = \frac{1}{h_1^4} (y_1 - 16y_{1/2} + 14a_{10} + 6h_1 a_{11} + 2h_1^2 a_{12})$	b)

Ecuación (5.29)

Esta estrategia **no requiere ninguna particularización** al *PVI* lineal (2.7), salvo la condición de colocación en el nodo inicial expresada en (5.8), pues al estar utilizando únicamente la condición adicional de interpolación intermedia en $y_{1/2}$, no se involucra a ninguna estrategia de colocación dependiente de la forma de la ecuación diferencial que define el *PVI*.

5.2.4.3. Condiciones adicionales. Estrategia de colocación intermedia y colocación en el nodo inicial $t_0 = a$.

Esta última estrategia de imposición de condiciones adicionales implica particularizar a $k = 1$ la ecuación (5.13), para obtener (5.30) como condición de interpolación intermedia en el intervalo inicial $[t_0, t_1]$.

$$\tilde{S}_1''(t_{1/2}) = f\left(t_{1/2}, \tilde{S}_1(t_{1/2}), \tilde{S}_1'(t_{1/2})\right)$$

Ecuación (5.30)

Como ocurre en apartados anteriores, agrupando ahora (5.30) con (5.16), se tiene un **Sistema 2x2** dado por (5.31):

$$\begin{cases} h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\ \tilde{S}_1''(t_{1/2}) = f\left(t_{1/2}, \tilde{S}_1(t_{1/2}), \tilde{S}_1'(t_{1/2})\right) \end{cases}$$

Ecuación (5.31)

La resolución del sistema anterior se realiza mediante algún método apropiado. **En un caso general, cabe la posibilidad de que sea no lineal.**

5.2.4.3.1. Particularización al caso lineal.

Asumiendo un *PVI* del tipo (2.7), se reescribe (5.30) en (5.32):

$$\tilde{S}_1''(t_{1/2}) = p_{1/2} \tilde{S}_1'(t_{1/2}) + q_{1/2} \tilde{S}_1(t_{1/2}) + r_{1/2}$$

Ecuación (5.32)

Utilizando nuevamente (5.16) en conjunto con la ecuación anterior, se obtiene un **Sistema 2x2 (5.33)**, particularización lineal de (5.31):

$$\begin{cases} h_1^3 a_{13} + h_1^4 a_{14} = y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} \\ A a_{13} + B a_{14} = q_{1/2} a_{10} + \left(p_{1/2} + \frac{1}{2} q_{1/2} h_1\right) a_{11} + \left(p_{1/2} h_1 + \frac{1}{4} q_{1/2} h_1^2 - 2\right) a_{12} + r_{1/2} \end{cases}$$

$$A = 3h_1 - \frac{3}{4} p_{1/2} h_1^2 - \frac{1}{8} q_{1/2} h_1^3, \quad B = 3h_1^2 - \frac{1}{2} p_{1/2} h_1^3 - \frac{1}{16} q_{1/2} h_1^4$$

Ecuación (5.33)

Se pueden obtener expresiones explícitas para a_{13} y a_{14} . Despejando a_{14} de la primera ecuación de (5.33) y operando, se tiene (5.34):

$$\begin{aligned} a_{14} &= \frac{1}{h_1^4} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} - h_1^3 a_{13}) \rightarrow \\ &\rightarrow \left(A - \frac{B}{h_1}\right) a_{13} + \frac{B}{h_1^4} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12}) \\ &= p_{1/2} (a_{11} + h_1 a_{12}) + q_{1/2} \left(a_{10} + \frac{1}{2} h_1 a_{11} + \frac{1}{4} h_1^2 a_{12}\right) + r_{1/2} - 2a_{12} \rightarrow \\ &\rightarrow \left(\frac{A h_1 - B}{h_1}\right) a_{13} = q_{1/2} a_{10} + \left(p_{1/2} + \frac{1}{2} q_{1/2} h_1\right) a_{11} + \left(p_{1/2} h_1 - \frac{1}{4} q_{1/2} h_1^2 + \frac{B}{h_1^2} - 2\right) a_{12} + r_{1/2} \rightarrow \end{aligned}$$

$a_{13} = \left(\frac{h_1}{A h_1 - B}\right) \left(\left(q_{1/2} + \frac{B}{h_1^4}\right) a_{10} + \left(p_{1/2} + \frac{1}{2} q_{1/2} h_1 + \frac{B}{h_1^3}\right) a_{11} + \left(p_{1/2} + q_{1/2} \frac{h_1^2}{4} + \frac{B}{h_1^2} - 2\right) a_{12} + \left(r_{1/2} - \frac{B y_1}{h_1^4}\right) \right)$	a)
$a_{14} = \frac{1}{h_1^4} (y_1 - a_{10} - h_1 a_{11} - h_1^2 a_{12} - h_1^3 a_{13})$	b)

Ecuación (5.34)

5.2.5. Resolución de un intervalo genérico. Obtención de \tilde{S}_k en $[t_{k-1}, t_k]$, $k = 2, 3, \dots, n$.

El esquema progresivo planteado para calcular los parámetros a_{ki} de (5.1) es **fácilmente generalizable** a todos los intervalos que satisfacen $k = 2, 3, \dots, n$, es decir, si $x \in [t_1, t_n]$. El tramo \tilde{S}_k del *pseudo-spline* cuártico \tilde{S} tomado expandiendo (5.1) ($p \leq 4$) es el dado por (5.35), junto con sus derivadas primera (\tilde{S}'_k) y segunda (\tilde{S}''_k).

$$\tilde{S}_k(x) = a_{k0} + a_{k1}(x - t_{k-1}) + a_{k2}(x - t_{k-1})^2 + a_{k3}(x - t_{k-1})^3 + a_{k4}(x - t_{k-1})^4 \quad \text{a)}$$

$$\tilde{S}'_k(x) = a_{k1} + 2a_{k2}(x - t_{k-1}) + 3a_{k3}(x - t_{k-1})^2 + 4a_{k4}(x - t_{k-1})^3 \quad \text{b)}$$



$$\tilde{S}_k''(x) = 2a_{k2} + 6a_{k3}(x - t_{k-1}) + 12a_{k4}(x - t_{k-1})^2 \quad \text{c)}$$

Ecuación (5.35)

Planteando sistemáticamente las condiciones de continuidad en el nodo inicial t_{k-1} y la condición de interpolación (indicadas en la **Tabla 5.2**) en el nodo extremo t_k , se tiene (5.36):

$$\tilde{S}_{k-1}(t_{k-1}) = \tilde{S}_k(t_{k-1}) \rightarrow \boxed{a_{k0} = \tilde{S}_{k-1}(t_{k-1})} \quad \text{a)}$$

$$\tilde{S}'_{k-1}(t_{k-1}) = \tilde{S}'_k(t_{k-1}) \rightarrow \boxed{a_{k1} = \tilde{S}'_{k-1}(t_{k-1})} \quad \text{b)}$$

$$\tilde{S}''_{k-1}(t_{k-1}) = \tilde{S}''_k(t_{k-1}) \rightarrow \boxed{a_{k2} = \frac{1}{2} \tilde{S}''_{k-1}(t_{k-1})} \quad \text{c)}$$

$$\tilde{S}_k(t_k) = y_k \rightarrow a_{k0} + h_k a_{k1} + h_k^2 a_{k2} + h_k^3 a_{k3} + h_k^4 a_{k4} = y_k \rightarrow \text{Interp. en } t_k \quad \text{d)}$$

Ecuación (5.36)

Reorganizando (5.36.d), surge (5.37):

$$h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2}$$

Ecuación (5.37)

Puede particularizarse la ecuación anterior a los valores conocidos de a_{k0} , a_{k1} y a_{k2} para obtener (5.38):

$$h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - \tilde{S}_{k-1}(t_{k-1}) - h_1 \tilde{S}'_{k-1}(t_{k-1}) - \frac{1}{2} h_1^2 \tilde{S}''_{k-1}(t_{k-1})$$

Ecuación (5.38)

Al igual que ocurría en la resolución del intervalo inicial $k = 1$ (sección 5.2.4), se precisa de una ecuación más para poder calcular a_{k3} y a_{k4} . Dicho cálculo se efectúa mediante la resolución de un **Sistema 2x2 Compatible y Determinado**. Las formas de obtener la 2ª ecuación necesaria para determinar unívocamente el tramo \tilde{S}_k del *pseudo-spline* cuártico \tilde{S} son diversas. La forma de la expresión resultante también depende del tipo de problema (*PVI* general (2.6) o lineal (2.7), con ciertas particularizaciones al caso independiente de y' (2.8) como en 5.2.4), por lo que es de interés realizar una distinción múltiple en el procedimiento a aplicar.

La determinación sistemática de parámetros a_{ki} definida en (5.36) puede transformarse en un **esquema recursivo** (5.39) para clarificar el procedimiento en el intervalo sucesivo $k + 1$ al intervalo desarrollado k . **Nótese que el razonamiento recursivo es solamente válido para $k = 1, \dots, n - 1$, pues la discretización no abarca más intervalos. Se tiene:**

$$\boxed{a_{(k+1)0} = a_{k0} + h_k a_{k1} + h_k^2 a_{k2} + h_k^3 a_{k3} + h_k^4 a_{k4}} \quad a)$$

$$\boxed{a_{(k+1)1} = a_{k1} + 2h_k a_{k2} + 3h_k^2 a_{k3} + 4h_k^3 a_{k4}} \quad b)$$

$$\boxed{a_{(k+1)2} = a_{k2} + 3h_k a_{k3} + 6h_k^2 a_{k4}} \quad c)$$

$$h_{k+1}^3 a_{(k+1)3} + h_{k+1}^4 a_{(k+1)4} = y_{k+1} - a_{(k+1)0} - h_{k+1} a_{(k+1)1} - h_{k+1}^2 a_{(k+1)2} \quad d)$$

Ecuación (5.39)

5.2.5.1. Condiciones adicionales. Estrategia de colocación.

Se plantea la condición de colocación en t_k dada por (5.5) junto con la interpolación de (5.39.d) en cada intervalo $k = 2, \dots, n$. Combinando estas expresiones con (5.37), se obtiene el **Sistema 2x2** que conduce al cálculo de los parámetros restantes a_{k3} y a_{k4} para determinar \tilde{S}_k . Se tiene (5.40):

$$\begin{cases} h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} \\ \tilde{S}_k''(t_k) = f(t_k, y_k, \tilde{S}_k'(t_k)) \end{cases}$$

Ecuación (5.40)

La resolución del sistema anterior se realiza mediante algún método apropiado. **En un caso general, cabe la posibilidad de que sea no lineal.** A continuación, se indica la forma de proceder, similar al intervalo inicial (5.2.4.1) pero partiendo de condiciones definidas mediante el esquema recursivo de (5.39), tanto para la particularización a problemas que no dependen de y' como al problema lineal.

5.2.5.1.1. Particularización a problemas independientes de y' .

La estrategia de colocación en intervalo genérico presenta también la ventaja de calcular los parámetros restantes (ahora a_{k3} y a_{k4}) de un caso genérico no lineal independiente de y' (**PVI** del tipo (2.8)) resolviendo un **Sistema 2x2** estrictamente lineal. Utilizando una forma particularizada de (5.40) a este tipo de problemas, se tiene (5.41):

$$\begin{cases} h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} \\ 6a_{k3} h_k + 12a_{k4} h_k^2 = -2a_{k2} - f(t_k, y_k) \end{cases}$$

Ecuación (5.41)

Se ha sustituido el valor de $\tilde{S}_k''(t_k)$ utilizando (5.35.c) teniendo en cuenta que $f(t_k, y_k)$ es conocida. Las expresiones explícitas de a_{k3} y a_{k4} se obtienen a continuación (5.42):

$$a_{k4} = \frac{1}{12h_k^2} (-2a_{k2} - 6h_k a_{k3} + f(t_k, y_k)) \rightarrow$$

$$\begin{aligned}
 a_{k3} &= \frac{1}{h_k^3} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} - h_k^4 a_{k4}) = \\
 &= \frac{2}{h_k^3} \left(y_k - a_{k0} - h_k a_{k1} - \frac{5}{6} h_k^2 a_{k2} - \frac{1}{12} h_k^2 f(t_k, y_k) \right) \rightarrow
 \end{aligned}$$

$$a_{k3} = \frac{2}{h_k^3} \left(-a_{k0} - h_k a_{k1} - \frac{5}{6} h_k^2 a_{k2} + \left(y_k - \frac{1}{12} h_k^2 f(t_k, y_k) \right) \right) \quad \text{a)}$$

$$a_{k4} = \frac{1}{h_k^4} \left(a_{k0} + h_k a_{k1} + \frac{2}{3} h_k^2 a_{k2} + \left(\frac{1}{6} h_k^2 f(t_k, y_k) - y_k \right) \right) \quad \text{b)}$$

Ecuación (5.42)

5.2.5.1.2. Particularización al caso lineal.

Ahora, utilizando el PVI (2.7), se reescribe (5.8) en (5.43):

$$\tilde{S}_k''(t_k) = p_k \tilde{S}_k'(t_k) + q_k \tilde{S}_k(t_k) + r_k$$

Ecuación (5.43)

 Utilizando (5.37) conjuntamente con la ecuación anterior y (5.36.d) para sustituir $\tilde{S}_k(t_k)$, se obtiene un **Sistema 2x2 (5.44)**, particularización lineal de (5.40).

$$\begin{cases}
 h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} \\
 A \cdot a_{k3} + B \cdot a_{k4} = p_k a_{k1} + 2(p_k h_k - 1) a_{k2} + q_k y_k + r_k
 \end{cases}$$

$$A = 3(2h_k - p_k h_k^2), \quad B = 4(3h_k^2 - p_k h_k^3)$$

Ecuación (5.44)

 Ya se pueden obtener las expresiones explícitas de a_{k3} y a_{k4} . Despejando a_{k4} de la primera ecuación de (5.44) y operando, se obtiene (5.45):

$$\begin{aligned}
 a_{k4} &= \frac{1}{h_k^4} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} - h_k^3 a_{k3}) \rightarrow \\
 &\rightarrow \left(A - \frac{B}{h_k} \right) a_{k3} + \frac{B}{h_k^4} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2}) = \\
 &= p_k a_{k1} + 2(p_k h_k - 1) a_{k2} + q_k y_k + r_k \rightarrow \\
 &\rightarrow \left(\frac{Ah_k - B}{h_k} \right) a_{k3} = \frac{B}{h_k^4} a_{k0} + \left(p_k + \frac{B}{h_k^3} \right) a_{k1} + \left(2p_k h_k - 2 + \frac{B}{h_k^2} \right) a_{k2} + \left(q_k - \frac{B}{h_k^4} \right) y_k + r_k \rightarrow
 \end{aligned}$$

$$a_{k3} = \left(\frac{h_k}{Ah_k - B} \right) \left(\frac{B}{h_k^4} a_{k0} + \left(p_k + \frac{B}{h_k^3} \right) a_{k1} + \left(2p_k h_k - 2 + \frac{B}{h_k^2} \right) a_{k2} + \left(q_k - \frac{B}{h_k^4} \right) y_k + r_k \right) \quad \text{a)}$$

$$a_{k4} = \frac{1}{h_k^4} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} - h_k^3 a_{k3}) \quad b)$$

Ecuación (5.45)

5.2.5.2. Condiciones adicionales. Estrategia de interpolación intermedia.

Considerar un intervalo genérico $k = 2, \dots, n$ distinto del inicial $k = 1$ implica no tener en cuenta la condición de colocación inicial (5.6) definida en t_0 (sección 5.2.1). Se debe imponer la condición de interpolación intermedia en $t_{k-1/2}$, a partir de la previamente determinada solución nodal $y_{k-1/2}$ (5.10). Particularizando (5.35.a) con $x = t_{k-1/2}$ y aplicando (5.11), se tiene (5.46):

$$\tilde{S}_k \left(t_{k-\frac{1}{2}} \right) = a_{k0} + \frac{h_k}{2} a_{k1} + \frac{h_k^2}{4} a_{k2} + \frac{h_k^3}{8} a_{k3} + \frac{h_k^4}{16} a_{k4} = y_{k-\frac{1}{2}}$$

Ecuación (5.46)

Agrupando en un **Sistema 2x2** la ecuación anterior con (5.37), surge (5.47):

$$\begin{cases} h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} \\ \frac{h_k^3}{8} a_{k3} + \frac{h_k^4}{16} a_{k4} = y_{k-\frac{1}{2}} - a_{k0} - \frac{h_k}{2} a_{k1} - \frac{h_k^2}{4} a_{k2} \end{cases}$$

Ecuación (5.47)

Este sistema puede ser resuelto mediante una sencilla estrategia de eliminación, y pueden obtenerse expresiones explícitas para a_{k3} y a_{k4} . Finalmente, se consigue (5.48):

$$a_{k3} = \frac{1}{h_k^3} \left(16y_{k-\frac{1}{2}} - y_k - 15a_{k0} - 7h_k a_{k1} - 3h_k^2 a_{k2} \right) \quad a)$$

$$a_{k4} = \frac{1}{h_k^4} \left(y_k - 16y_{k-\frac{1}{2}} + 14a_{k0} + 6h_k a_{k1} + 2h_k^2 a_{k2} \right) \quad b)$$

Ecuación (5.48)

Como ya se comentó en 5.2.2 para el intervalo inicial $[t_0, t_1]$, esta estrategia **no requiere ninguna particularización** al PVI lineal (2.7), pues al estar utilizando únicamente la condición adicional de interpolación intermedia en $y_{k-1/2}$, no se involucra a ninguna estrategia de colocación dependiente de la forma de la EDO que define el PVI.

5.2.5.3. Condiciones adicionales. Estrategia de colocación intermedia.

Si se vuelve a involucrar a (5.37) con otra ecuación surgida de la imposición de una condición adicional, en este caso colocación intermedia dada por (5.12),

se vuelve a obtener un **Sistema 2x2** para calcular los parámetros restantes a_{k3} y a_{k4} (5.49).

$$\begin{cases} h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} \\ \tilde{S}_k'' \left(t_{k-\frac{1}{2}} \right) = f \left(t_{k-\frac{1}{2}}, \tilde{S}_k \left(t_{k-\frac{1}{2}} \right), \tilde{S}_k' \left(t_{k-\frac{1}{2}} \right) \right) \end{cases}$$

Ecuación (5.49)

La resolución del sistema anterior se realiza mediante algún método apropiado. Una vez más, **en un caso general, cabe la posibilidad de que sea no lineal.**

5.2.5.3.1. Particularización al caso lineal.

Volviendo a asumir un **PVI** de tipo (2.7), se reescribe (5.12) en (5.50):

$$\tilde{S}_k'' \left(t_{k-\frac{1}{2}} \right) = p_{k-\frac{1}{2}} \tilde{S}_k' \left(t_{k-\frac{1}{2}} \right) + q_{k-\frac{1}{2}} \tilde{S}_k \left(t_{k-\frac{1}{2}} \right) + r_{k-\frac{1}{2}}$$

Ecuación (5.50)

Utilizando (5.37) conjuntamente con la ecuación anterior, se obtiene un **Sistema 2x2** (5.51), particularización lineal de (5.49).

$$\begin{cases} h_k^3 a_{k3} + h_k^4 a_{k4} = y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} \\ A a_{k3} + B a_{k4} = q_{k-\frac{1}{2}} a_{k0} - \left(p_{k-\frac{1}{2}} + \frac{1}{2} q_{k-\frac{1}{2}} h_k \right) a_{k1} + \left(p_{k-\frac{1}{2}} h_k + \frac{1}{4} q_{k-\frac{1}{2}} h_k^2 - 2 \right) a_{k2} + r_{k-\frac{1}{2}} \end{cases}$$

$$A = 3h_{k-1} - \frac{3}{4} p_{k-\frac{1}{2}} h_k^2 - \frac{1}{8} q_{k-\frac{1}{2}} h_k^3, \quad B = 3h_k^2 - \frac{1}{2} p_{k-\frac{1}{2}} h_k^3 - \frac{1}{16} q_{k-\frac{1}{2}} h_k^4$$

Ecuación (5.51)

Las expresiones explícitas de a_{k3} y a_{k4} ya se pueden obtener. Despejando a_{k4} de la primera ecuación de (5.51) y operando, surge (5.52):

$$a_{k4} = \frac{1}{h_k^4} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} - h_k^3 a_{k3}) \rightarrow$$

$$\begin{aligned} \rightarrow \left(A - \frac{B}{h_k} \right) a_{k3} + \frac{B}{h_k^4} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2}) \\ = p_{k-\frac{1}{2}} (a_{k1} + h_k a_{k2}) + q_{k+\frac{1}{2}} \left(a_{k0} + \frac{1}{2} h_k a_{k1} + \frac{1}{4} h_k^2 a_{k2} \right) + r_{k-\frac{1}{2}} - 2a_{k2} \rightarrow \end{aligned}$$

$$\rightarrow \left(\frac{A h_k - B}{h_k} \right) a_{k3} = q_{k-\frac{1}{2}} a_{k0} + \left(p_{k-\frac{1}{2}} + \frac{1}{2} q_{k-\frac{1}{2}} h_k \right) a_{k1} + \left(p_{k-\frac{1}{2}} h_k - \frac{1}{4} q_{k-\frac{1}{2}} h_k^2 + \frac{B}{h_k^2} - 2 \right) a_{k2} + r_{k-\frac{1}{2}} \rightarrow$$

$$a_{k3} = \left(\frac{h_k}{A h_k - B} \right) \left(\left(q_{k-\frac{1}{2}} - \frac{B}{h_k^4} \right) a_{k0} + \left(p_{k-\frac{1}{2}} + \frac{1}{2} q_{k-\frac{1}{2}} h_k + \frac{B}{h_k^3} \right) a_{k1} + \left(p_{k-\frac{1}{2}} h_k + \frac{1}{4} q_{k-\frac{1}{2}} h_k^2 + \frac{B}{h_k^2} - 2 \right) a_{k2} + \left(r_{k-\frac{1}{2}} - \frac{B y_k}{h_k^4} \right) \right) \quad \text{a)}$$

$$a_{k4} = \frac{1}{h_k^4} (y_k - a_{k0} - h_k a_{k1} - h_k^2 a_{k2} - h_k^3 a_{k3}) \quad b)$$

Ecuación (5.52)

5.2.6. Resumen del procedimiento.

El cuadro sinóptico correspondiente a la **Tabla 5.3** resume las condiciones impuestas sobre el pseudo-spline cuártico (5.2) expuestas en los anteriores apartados, con el objeto de resumir la vasta presentación que se ha realizado.

<i>Distribución de intervalos</i>		
	<i>Inicial (secc. 5.2.4)</i>	<i>Genérico (secc. 5.2.5)</i>
TIPOS DE CONDICIONES COMUNES		
Continuidad	-	$3(n-1) \rightarrow$ Tabla 5.2
Interpolación	$2 \rightarrow$ (5.15.a) y (5.15.d)	$n-1 \rightarrow$ Tabla 5.2 o (5.36.d)
Adicionales comunes	$2 \rightarrow$ (5.15.b) y (5.15.c)	-
TOTALES COMUNES	$4 + 4(n-1) = 4n$	
CONDICIONES ADICIONALES DEPENDIENTES DE LA ESTRATEGIA		
Colocación en el nodo inicial (5.15.b) presente en todas por definirse en t_0 , aunque en rigor también pertenece a la definición de cada estrategia.		
Colocación	$1 \rightarrow$ (5.18) indep. de y' (5.22) lin. (5.24)	$n-1 \rightarrow$ (5.5) indep. de y' (5.41) lin. (5.40)
Interpolación intermedia	$1 \rightarrow$ (5.27)	$n-1 \rightarrow$ (5.46)
Colocación intermedia	$1 \rightarrow$ (5.30) lin. (5.32)	$n-1 \rightarrow$ (5.12) lin. (5.50)
TOTALES	$4n + n = \boxed{5n}$	

Tabla 5.3. Cuadro sinóptico resumen. Pseudo-splines cuárticos interpoladores.

5.2.7. Resumen de la implementación en MATLAB.

El lector habrá podido comprobar que existen multitud de estrategias disponibles para interpolar las soluciones discretas de los PVI (2.6), (2.7) y (2.8) expuestos en el **Capítulo 2** (y calculadas bien de forma exacta o utilizando métodos numéricos aproximados como los del **Capítulo 3**) empleando la malla (2.11) y trazadores polinomiales basados en pseudo-splines cuárticos que proporcionen aproximaciones continuas.



Por tanto, y en concordancia con el marco teórico planteado sobre ellos, se indican en la **Tabla 5.4** las funciones disponibles en el **ANEXO III** de esta memoria (con su nombre o alias asignado en el programa) que implementan todas las estrategias planteadas para problemas lineales (2.7) e independientes de la primera derivada y' (2.8). El **PVI no lineal genérico (2.6)** ha de ser abordado individualmente, por lo que no se dispone ninguna rutina universal para él basada en los esquemas recursivos desarrollados. No obstante, los conceptos que deben aplicarse son similares y también responden al marco teórico recientemente expuesto.

Todas las funciones tienen un único objetivo, obtener los parámetros (coeficientes) α_{ki} del **pseudo-spline** cuártico interpolador (5.2) de la solución discreta del **PVI** en cada intervalo en el que viene descrito por un polinomio de, a lo sumo, cuarto grado.

<i>Estrategia</i>	<i>Sección memoria</i>	<i>Alias en MATLAB</i>
Colocación independiente de y'	5.2.1 (genérico), 5.2.4.1.1 y 5.2.5.1.1	"pspline4int_col1"
Colocación lineal	5.2.1 (genérico), 5.2.4.1.2 y 5.2.5.1.2	"pspline4int_col2"
Interpolación intermedia	5.2.2 (genérico), 5.2.4.2 y 5.2.5.2	"psplinet4int_intm"
Colocación intermedia lineal	5.2.3 (genérico), 5.2.4.3.1 y 5.2.5.3.1	"psplinet4int_colm_lin"

Tabla 5.4. Resumen de la implementación en **MATLAB** de estrategias de interpolación basadas en *pseudo-splines* cuárticos.

5.3. Estrategias de interpolación con polinomios osciladores.

Expuesto el extenso marco teórico concerniente al empleo de *pseudo-splines* cuárticos para interpolar soluciones discretas de los **PVI** definidos en el **Capítulo 2** de esta memoria, debe resaltarse un matiz importante que para nada resulta ser un hándicap, pero sí es una cuestión matizable y mejorable. El lector habrá podido comprobar que hasta ahora, incluyendo todas las estrategias de interpolación empleando *splines* (cuadráticos y cúbicos) del **Capítulo 4**, tan sólo se aprovechan los datos de las soluciones discretas y_k (aproximadas o exactas $y(t_k)$) en los nodos de la malla (2.11).

No obstante, y sin perjuicio a las condiciones impuestas que garantizan la concordancia de la aproximación con la solución continua de los **PVI**, puede mejorarse el esquema mediante la inserción de condiciones de interpolación también sobre la primera (z_k) o segunda derivada (w_k), las cuales casan con la importancia que pueden llegar a tener las aproximaciones a dichas derivadas en la descripción de fenómenos físicos. Estos datos están disponibles, bien sea

por la aplicación de un método numérico del **Capítulo 3** o por disponer de las soluciones exactas. Es importante denotar que la segunda derivada discreta w_k (o exacta $y''(t_k)$) se obtiene aprovechando la expresión en forma normal de las **EDOs** que componen los **PVIs**, mediante una **sencilla estrategia de colocación** indicada (en nodos de (2.11)), por (5.53), y utilizando (2.6) como problema genérico a abordar.

$$\{w_k\}_{k=0}^n \rightarrow w_k = f(t_k, y_k, z_k) \cong y''(t_k)$$

Ecuación (5.53)

Dicho esto, las estrategias de interpolación que se aprovechan de toda la información disponible en los nodos de la malla citada y desarrolladas en este trabajo corresponden a **pseudo-splines** construidos en base a la teoría de **polinomios osculadores**. Cuantas más condiciones concordantes con la realidad física del problema a resolver, mejores resultados finales han de ser los esperados. Se procede a continuación al desarrollo de los trazadores polinomiales que cumplen las características citadas de forma introductoria en estos párrafos, y siempre manteniendo el **PVI (2.6)** como referencia única, sin entrar en particularizaciones como las realizadas en la sección 5.2 de este capítulo.

5.3.1. Definición del polinomio osculador P de grado m .

Los polinomios osculadores pueden considerarse una generalización de los polinomios de *Taylor* y de *Lagrange* (entre otros), los cuales no son abordados en ninguna sección de esta memoria, pero cuyo marco teórico y aplicaciones pueden consultarse en [12, 46-48].

Se considera la discretización de x correspondiente al **PVI (2.6)** y dada por (2.11) y un conjunto de números enteros no negativos definido en (5.54):

$$m_0, m_1, \dots, m_n \rightarrow \{m_k\}_{k=0}^n \mid m = \max\{\{m_k\}_{k=0}^n\}$$

Ecuación (5.54)

El polinomio osculador que aproxima una función genérica $g \in C^m$ en los nodos $\{t_k\}_{k=0}^n$ del intervalo $x \in [a, b]$ (con $t_0 = a$ y $t_n = b$) es el **polinomio de mínimo grado que presenta los mismos valores que g y todas sus derivadas de orden menor que o igual a m_k en cada t_k de (2.11)**. El grado del polinomio osculador será a lo sumo el máximo dado por (5.55):

$$M = n + \sum_{k=0}^n m_k$$

Ecuación (5.55)

La razón de ser de lo anterior radica en las condiciones que deben satisfacerse, cuyo número asciende a $(n + 1) + \sum_{k=0}^n m_k$, y también en que un polinomio de



grado como máximo M tiene $M + 1$ coeficientes que pueden utilizarse para hacer que se satisfagan. Por tanto, partiendo de (2.11) y (5.54) se define el polinomio osculador que aproxima a g como el polinomio Q de menor grado que cumple (5.56):

$$Q(x) \mid Q^{(j)}(t_k) = g^{(j)}(t_k), \quad k = 0, 1, \dots, n, \quad j = 0, 1, \dots, m_k$$

Ecuación (5.56)

La **Tabla 5.5** resume las particularizaciones que se pueden realizar de Q para conseguir otros polinomios conocidos en la literatura científica, con especial atención a las dos últimas, las cuales son las empleadas en este trabajo.

<i>Consideración</i>	<i>Polinomio osculador particularizado</i>
$n = 0$	(2.11) se reduce a $x_0 = t_0$. Se obtiene el m_0 -ésimo polinomio de <i>Taylor</i> que aproxima a g en un entorno de t_0 [12].
$m_k = 0,$ $k = 0, 1, \dots, n$	Obtención del polinomio interpolador de <i>Lagrange</i> de g en (2.11) [12].
$m_k = 1,$ $k = 0, 1, \dots, n$	Obtención del polinomio interpolador de <i>Hermite</i> de g en (2.11) empleado en 5.3.2 para la interpolación con <i>pseudo-splines</i> cúbicos.
$m_k = 2,$ $k = 0, 1, \dots, n$	Obtención del polinomio osculador interpolador de g en (2.11) empleado en 5.3.3 para la interpolación con <i>pseudo-splines</i> quínticos.

Tabla 5.5. Particularizaciones relevantes de los polinomios osculadores.

5.3.2. Estrategia de interpolación con *pseudo-splines* cúbicos basada en los polinomios interpoladores de *Hermite*.

Hasta ahora, cuando se han expuesto las estrategias de interpolación de la solución discreta y_k del *PVI* (2.6) utilizando trazadores polinomiales cúbicos, éstos se definían como *splines* (Capítulo 4), puesto que cumplían los requisitos en cuanto a continuidad de los mismos y sus derivadas, de modo que $S \in C^2$.

La estrategia presentada en esta sección concierne al empleo de *pseudo-splines* cúbicos, los cuales son trazadores del mismo grado que los anteriores pero cuya regularidad viene descrita por $\tilde{S} \in C^1$, siendo necesarias $4n$ condiciones a imponer para su cálculo, con $n + 1$ nodos (2.11) y n intervalos. La determinación unívoca de \tilde{S} se consigue mediante la imposición de condiciones de continuidad y de interpolación (incluyendo la primera derivada). Se tiene la **Tabla 5.6**:

<i>Tipo</i>	<i>Número</i>	<i>Forma</i>
Continuidad	$2(n - 1)$	S y S' continuas en $\{t_{k-1}\}_{k=2}^n$ $\begin{cases} \tilde{S}_{k-1}(t_{k-1}) = \tilde{S}_k(t_{k-1}) \rightarrow n - 1 \\ \tilde{S}'_{k-1}(t_{k-1}) = \tilde{S}'_k(t_{k-1}) \rightarrow n - 1 \end{cases}$
Interpolación	$2(n + 1)$	Interpolación en $\{t_k\}_{k=0}^n$ $\begin{cases} \tilde{S}(t_k) = y_k \rightarrow \text{Interpolación de } \tilde{S} \rightarrow n + 1 \\ \tilde{S}'(t_k) = z_k \rightarrow \text{Interpolación de } \tilde{S}' \rightarrow n + 1 \end{cases}$
TOTALES		$4n$

Tabla 5.6. Resumen de condiciones a imponer para determinar unívocamente un *pseudo-spline* cúbico interpolador.

Estas condiciones se consiguen imponer mediante el empleo del **polinomio interpolador de Hermite** en cada intervalo k utilizando dos nodos, los cuales son los extremos t_{k-1} y t_k de $[t_{k-1}, t_k]$. **Nótese que ahora ya no se tiene un esquema recursivo, como el que surgía para todas las estrategias del Capítulo 4 y los pseudo-splines cuárticos de este.** Cada intervalo se resuelve de forma independiente al anterior y sin influir en el posterior.

5.3.2.1. El polinomio interpolador de Hermite.

Como ya se ha remarcado en la **Tabla 5.5**, el polinomio interpolador de *Hermite* resulta ser una **particularización de los polinomios osculadores cuando $m_k = 1$** , siendo $k = 0, 1, \dots, n$. Esto garantiza la concordancia del polinomio con la solución discreta y_k en los nodos de (2.11) y también con la de sus primeras derivadas z_k .

5.3.2.1.1. Teorema de existencia y unicidad del polinomio interpolador de Hermite.

Dada una función $g \in C^1$ definida en $x \in [c, d]$ y un conjunto de nodos $\{x_j\}_{j=0}^n$ distintos (con $x_0 = c$ y $x_n = d$), el único polinomio de menor grado que concuerda con g y g' en x_j es el polinomio de Hermite H de grado como máximo $2n + 1$ dado por (5.57):

$$H_{2n+1}(x) = \sum_{j=0}^n g(x_j)H_{n,j}(x) + \sum_{j=0}^n g'(x_j)\hat{H}_{n,j}(x)$$

Ecuación (5.57)

Se definen los j -ésimos coeficientes del polinomio de *Lagrange* de grado n como $L_{n,j}$ en (5.58):

$$L_{n,j}(x) = \frac{(x - x_0) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}$$

Ecuación (5.58)



Ahora, utilizando (5.58), surgen las expresiones para $H_{n,j}$ y $\hat{H}_{n,j}$ (5.59):

$$H_{n,j}(x) = \left(1 - 2(x - x_j)L'_{n,j}(x_j)\right) L_{n,j}^2(x) \quad \text{a)}$$

$$\hat{H}_{n,j}(x) = (x - x_j)L_{n,j}^2(x) \quad \text{b)}$$

Ecuación (5.59)

Además, si $g \in C^{2n+2}$ en $x \in [c, d]$, se tiene (5.60):

$$g(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \dots (x - x_n)^2}{(2n + 2)!} g^{(2n+2)}(\xi(x)) \quad \text{a)}$$

$$\xi(x) \in (\text{mín}\{x, c, \dots, d\}, \text{máx}\{x, c, \dots, d\}) \quad \text{b)}$$

Ecuación (5.60)

La expresión anterior se define para algunos (mayoritariamente desconocidos) $\xi(x)$ y está compuesta por una aproximación a g dada por el polinomio de Hermite H_{2n+1} (5.59) junto con un término de error (segundo sumando).

Corolario. Polinomio interpolador de Hermite en un intervalo $[t_{k-1}, t_k]$, $k = 1, 2, \dots, n$.

Considerando el tramo del *pseudo-spline* cúbico \tilde{S} dado por el polinomio $\tilde{S}_k \in C^\infty$ en el intervalo $x \in [t_{k-1}, t_k]$, se construye éste último empleando el polinomio interpolador de Hermite definido en (5.57) con $n = 1$ (dos nodos t_{k-1} y t_k). Empleando la discretización de la variable independiente x descrita en (2.11), junto con la solución discreta $\{y_k\}_{k=0}^n$ del PVI (2.6) y su primera derivada $\{z_k\}_{k=0}^n$, se tiene (5.61):

$$\tilde{S}_k(x) = \sum_{j=k-1}^k y_j H_{1,j}(x) + \sum_{j=k-1}^k z_j \hat{H}_{1,j}(x)$$

Ecuación (5.61)

Nótese que se ha obtenido un polinomio de grado $p \leq 3$ ($2n + 1$) en virtud de la definición del *pseudo-spline* cúbico \tilde{S} . Además, los coeficientes del polinomio de Lagrange de grado n dados en (5.58) se particularizan a grado unitario en (5.62), junto con las agrupaciones dadas por (5.59).

$$L_{1,k-1}(x) = \frac{t_k - x}{h_k}, \quad L_{1,k}(x) = \frac{x - t_{k-1}}{h_k} \quad \text{a)}$$

$$H_{1,j}(x) = \left(1 - 2(x - t_j)L'_{1,j}(t_j)\right) L_{1,j}^2(x) \quad \text{b)}$$

$$\hat{H}_{1,j}(x) = (x - t_j)L_{n,j}^2(x) \quad c)$$

Ecuación (5.62)

5.3.2.1.2. Fórmula de diferencias divididas de Newton para polinomios de Hermite.

Computacionalmente, la obtención de (5.57) utilizando polinomios de Lagrange y sus derivadas resulta ser complicada. Por ello, se emplea la **fórmula de diferencias divididas de Newton con argumentos repetidos** [49] para obtener el polinomio de Hermite. La razón de ser de este método radica en la concordancia entre la n -ésima diferencia dividida y la n -ésima derivada de la función g , a la cual (5.57) interpola en $\{x_j\}_{j=0}^n$. Con el objeto de materializar esta conexión, debe considerarse la nueva sucesión de nodos repetidos definida en (5.63):

$$\{v_{2j} = v_{2j+1} = x_j\}_{j=0}^n$$

Ecuación (5.63)

Para facilitar la comprensión, la definición de las diferencias divididas surgidas se aporta en la **Tabla 5.7**, mediante un sencillo ejemplo particular empleando dos nodos ($n = 1$ o dos nodos repetidos, que son los necesarios en el desarrollo posterior). El esquema resultante [12] es el siguiente:

v	$g(v)$	$D.D. \ 0/1$	$D.D. \ 0/2$	$D.D. \ 0/3$
$v_0 = x_0$	$g[v_0] = \boxed{g(x_0)}$	$g[v_0, v_1] = \boxed{g'(x_0)}$	$g[v_0, v_1, v_2] = \boxed{\frac{g[v_1, v_2] - g[v_0, v_1]}{v_2 - v_0}}$	$g[v_0, v_1, v_2, v_3] = \boxed{\frac{g[v_1, v_2, v_3] - g[v_0, v_1, v_2]}{v_3 - v_0}}$
$v_1 = x_0$	$g[v_1] = g(x_0)$	$g[v_1, v_2] = \frac{g[v_2] - g[v_1]}{v_2 - v_1}$	$g[v_1, v_2, v_3] = \frac{g[v_2, v_3] - g[v_1, v_2]}{v_3 - v_1}$	
$v_2 = x_1$	$g[v_2] = g(x_1)$	$g[v_2, v_3] = g'(x_1)$		
$v_3 = x_1$	$g[v_3] = g(x_1)$			

Tabla 5.7. Esquema de diferencias divididas con argumentos repetidos. Interpolación de Hermite.

Las expresiones encuadradas no son baladíes, pues permiten obtener unívocamente el polinomio interpolador de Hermite (de grado a lo sumo 3) de g en $\{x_j\}_{j=0}^3$ (5.64):

$$H_3(x) = g[v_0] + \sum_{i=1}^3 g[v_0, \dots, v_i] \cdot (x - v_0) \cdot (x - v_1) \dots (x - v_{i-1})$$

Ecuación (5.64)

Finalmente, extendiendo el esquema de la **Tabla 5.7** a $\{x_j\}_{j=0}^n$, se determina la expresión general del polinomio interpolador de *Hermite* de g con diferencias divididas de *Newton*, el cual es un polinomio de grado $p \leq (2n + 1)$ según (5.65):

$$H_{2n+1}(x) = g[v_0] + \sum_{j=1}^{2n+1} g[v_0, \dots, v_j] \cdot (x - v_0) \cdot (x - v_1) \dots (x - v_{j-1})$$

Ecuación (5.65)

5.3.2.2. Resolución de un intervalo genérico k . Obtención de \tilde{S}_k en $[t_{k-1}, t_k]$, $k = 1, 2, \dots, n$.

Una vez expuestas las consideraciones teóricas sobre el esquema de diferencias divididas con argumentos repetidos, se puede proceder a determinar unívocamente el tramo \tilde{S}_k del *pseudo-spline* cúbico \tilde{S} en el intervalo $[t_{k-1}, t_k]$.

Para ello, y en virtud de la correspondencia existente entre el polinomio interpolador de *Hermite* de y_k en los dos nodos extremos del intervalo genérico y \tilde{S}_k dada por el **Corolario 5.3.2.1.1**, se considera el esquema de la **Tabla 5.8**. Este esquema es una particularización del de la **Tabla 5.7** a la solución discreta del **PVI (2.6)** (en vez de la función genérica g) y su primera derivada construidas en base a (2.11), fijando los nodos que interesan en t_{k-1} y t_k .

v	$\{y_k\}_{k-1}^k$	D.D. o/1	D.D. o/2	D.D. o/3
$v_0 = t_{k-1}$	$y[v_0] = \boxed{y_{k-1}}$	$y[v_0, v_1] = \boxed{z_{k-1}}$	$y[v_0, v_1, v_2] = \frac{1}{h_k^2}(y_k - y_{k-1} - h_k z_{k-1})$	$y[v_0, v_1, v_2, v_3] = \frac{2}{h_k^3}(y_{k-1} - y_k) + \frac{1}{h_k^2}(z_k + z_{k-1})$
$v_1 = t_{k-1}$	$y[v_1] = y_{k-1}$	$y[v_1, v_2] = \frac{y_k - y_{k-1}}{h_k}$	$y[v_1, v_2, v_3] = \frac{1}{h_k^2}(h_k z_k - y_k + y_{k-1})$	
$v_2 = t_k$	$y[v_2] = y_k$	$y[v_2, v_3] = z_k$		
$v_3 = t_k$	$y[v_3] = y_k$			

Tabla 5.8. Diferencias divididas para el cálculo de \tilde{S}_k mediante el polinomio interpolador de *Hermite*.

Nótese que se ha hecho uso del paso h_k definido en (2.12). Obtener \tilde{S}_k es inmediato, pues solamente es necesario aplicar su correspondencia con (5.57) dada en (5.61), pero **utilizando las diferencias divididas encuadradas $y[v_0 \dots v_3]$ calculadas en la **Tabla 5.8** y argumentos repetidos hasta tercer orden según (5.65).** Por tanto, se tiene (5.66):

$$\tilde{S}_k(x) = y[v_0] + \sum_{j=1}^3 y[v_0, \dots, v_j] \cdot (x - v_0) \cdot (x - v_1) \dots (x - v_{j-1})$$

Ecuación (5.66)

Desarrollando, surge la expresión definitiva del tramo \tilde{S}_k correspondiente al pseudo-spline cúbico interpolador \tilde{S} (5.67):

$$\begin{aligned} \tilde{S}_k(x) = & y_{k-1} + z_{k-1} \cdot (x - t_{k-1}) + \frac{1}{h_k^2} (y_k - y_{k-1} - h_k z_{k-1}) \cdot (x - t_{k-1})^2 \\ & + \left[\frac{2}{h_k^3} (y_{k-1} - y_k) + \frac{1}{h_k^2} (z_k + z_{k-1}) \right] \cdot (x - t_{k-1})^2 \cdot (x - t_k) \end{aligned}$$

Ecuación (5.67)

Una característica muy importante del esquema planteado es la **imposición implícita de las condiciones de continuidad e interpolación** definidas en la **Tabla 5.6** que determinan unívocamente el pseudo-spline \tilde{S} en cada intervalo (caso de estudio con \tilde{S}_k). El esquema de la **Tabla 5.8** impone continuidad e interpolación sistemáticamente cuando éste es aplicado a cada intervalo, debido a que intervalos sucesivos $k + 1$ comparten el nodo t_k , y en dicho nodo se interpola a y_k tanto con \tilde{S}_{k+1} como con \tilde{S}_k . Al mismo tiempo, la interpolación de la primera derivada z_k en dicho nodo se realiza con \tilde{S}'_{k+1} y \tilde{S}'_k , lo que implica estar considerando implícitamente continuidad C^1 . Si se aplica el procedimiento al intervalo $k - 1$, éste es análogo, pues el razonamiento expuesto se traslada a t_{k-1} , y así sucesivamente si se desea expandir por la izquierda o por la derecha de k . En definitiva, se tiene la **Figura 5.1**:

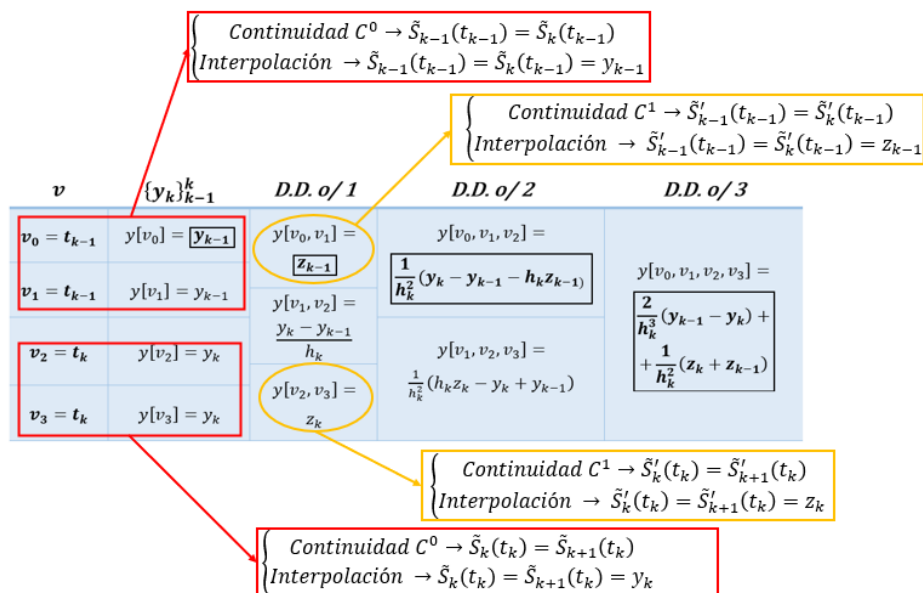


Figura 5.1. Imposición de las condiciones de continuidad y de interpolación con pseudo-splines cúbicos.



5.3.3. Estrategia de interpolación con *pseudo-splines* quínticos basada en polinomios osculadores.

Los *pseudo-splines* cúbicos de la sección 5.3.2 que interpolan en (2.11) a la solución discreta y_k del PVI (2.6) y a su derivada z_k son una buena herramienta cuando se desea imponer continuidad C^1 e interpolación conjunta hasta la primera derivada, lo cual es ventajoso para conseguir regularidad del trazador sin involucrar a la forma del PVI, que puede complicarse si éste presenta no linealidades. No obstante, el cálculo de dichos trazadores no utiliza en ningún momento la imposición de condiciones ligadas a la segunda derivada discreta w_k (5.53). Para poder utilizar estas últimas, se precisa el empleo de *pseudo-splines* quínticos.

Los *pseudo-splines* quínticos considerados se definen como trazadores polinomiales de grado a lo sumo 5 y $\tilde{S} \in C^2$, además de ser interpoladores en (2.11) de y_k , z_k y w_k . El número total de condiciones a imponer para su determinación unívoca asciende a $6n$, con $n + 1$ nodos y n intervalos. Cada tramo de \tilde{S} en $[t_{k-1}, t_k]$ y $k = 1, 2, \dots, n$ viene descrito por \tilde{S}_k . Las condiciones de continuidad e interpolación a imponer son las indicadas por la **Tabla 5.9**:

<i>Tipo</i>	<i>Número</i>	<i>Forma</i>
Continuidad	$3(n - 1)$	$\tilde{S}, \tilde{S}' \text{ y } \tilde{S}'' \text{ continuas en } \{t_{k-1}\}_{k=2}^n$ $\begin{cases} \tilde{S}_{k-1}(t_{k-1}) = \tilde{S}_k(t_{k-1}) \rightarrow n - 1 \\ \tilde{S}'_{k-1}(t_{k-1}) = \tilde{S}'_k(t_{k-1}) \rightarrow n - 1 \\ \tilde{S}''_{k-1}(t_{k-1}) = \tilde{S}''_k(t_{k-1}) \rightarrow n - 1 \end{cases}$
Interpolación	$3(n + 1)$	$\text{Interpolación en } \{t_k\}_{k=0}^n$ $\begin{cases} \tilde{S}(t_k) = y_k \rightarrow \text{Interpolación de } \tilde{S} \rightarrow n + 1 \\ \tilde{S}'(t_k) = z_k \rightarrow \text{Interpolación de } \tilde{S}' \rightarrow n + 1 \\ \tilde{S}''(t_k) = w_k \rightarrow \text{Interpolación de } \tilde{S}'' \rightarrow n + 1 \end{cases}$
TOTALES		$6n$

Tabla 5.9. Resumen de condiciones a imponer para determinar unívocamente un *pseudo-spline* quíntico interpolador.

Para imponer de forma sistemática todas las condiciones definidas en las expresiones anteriores y calcular unívocamente \tilde{S} , se emplea en cada intervalo $[t_{k-1}, t_k]$ el polinomio osculador particularizado a todos los valores de (5.54) $m_k = 2$ comentado en la última fila de la **Tabla 5.5**, que determina el tramo \tilde{S}_k de \tilde{S} .

5.3.3.1. Resolución de un intervalo genérico k . Obtención de \tilde{S}_k en $[t_{k-1}, t_k]$, $k = 1, 2, \dots, n$.

Como ya se ha mencionado, se considera el polinomio osculador Q ([12], [50] y [51]) con todos los enteros no negativos $m_k = 2$, dos nodos (t_{k-1} y t_k) y grado $p \leq 5$ (en virtud de (5.55)) perteneciente a la familia que satisface (5.56) y que

interpola en (2.11) a y_k, z_k y w_k del PVI (2.6). Por tanto, este polinomio puede utilizarse para obtener \tilde{S}_k en $[t_{k-1}, t_k]$, tal y como se indica en (5.68):

$$\tilde{S}_k(x) = Q_5(x) = y[v_0] + \sum_{j=1}^5 y[v_0, \dots, v_j] \cdot (x - v_0) \cdot (x - v_1) \dots (x - v_{j-1})$$

Ecuación (5.68)

En la expresión anterior, se considera la sucesión de nodos triplicados v_{3j} dada por (5.69) y modificada respecto a la de nodos duplicados v_{2j} (5.63).

$$\{v_{3j} = v_{3j+1} = v_{3j+2} = t_j\}_{j=0}^n$$

Ecuación (5.69)

El esquema en diferencias divididas de Newton con argumentos repetidos adoptado ahora para evaluar $y[v_0, \dots, v_j]$ corresponde a la Tabla 5.10, utilizando (5.69) con $n = 1$ (nodos extremos del intervalo k , t_{k-1} y t_k triplicados cada uno).

v	$\{y_k\}_{k-1}^k$	D.D. o/1	D.D. o/2	D.D. o/3	D.D. o/4	D.D. o/5
$v_0 = t_{k-1}$	$y[v_0] = \boxed{y_{k-1}}$	$y[v_0, v_1] = \boxed{z_{k-1}}$				
$v_1 = t_{k-1}$	$y[v_1] = y_{k-1}$	$y[v_1, v_2] = z_{k-1}$	$y[v_0, v_1, v_2] = \boxed{\frac{w_{k-1}}{2}}$	$\boxed{y[v_0, v_1, v_2, v_3]}$		
$v_2 = t_{k-1}$	$y[v_2] = y_{k-1}$	$y[v_2, v_3] = \frac{y_k - y_{k-1}}{h_k}$	$y[v_1, v_2, v_3]$	$y[v_1, v_2, v_3, v_4]$	$\boxed{y[v_0, v_1, v_2, v_3, v_4]}$	
$v_3 = t_k$	$y[v_3] = y_k$	$y[v_3, v_4] = z_k$	$y[v_2, v_3, v_4]$	$y[v_2, v_3, v_4, v_5]$	$y[v_1, v_2, v_3, v_4, v_5]$	
$v_4 = t_k$	$y[v_4] = y_k$		$y[v_3, v_4, v_5] = \frac{w_k}{2}$			
$v_5 = t_k$	$y[v_5] = y_k$	$y[v_4, v_5] = z_k$				$\boxed{y[v_0, v_1, v_2, v_3, v_4, v_5]}$

Tabla 5.10. Esquema de diferencias divididas con argumentos repetidos para calcular \tilde{S}_k . Pseudo-splines cuánticos interpoladores.

En la tabla anterior, se ha vuelto a hacer uso de (2.12). Las expresiones correspondientes a las diferencias divididas no desarrolladas se obtienen en la Tabla 5.11:

Nomenclatura	Expresión
$y[v_1, v_2, v_3]$	$\frac{1}{h_k^2}(y_k - y_{k-1} - h_k z_{k-1})$
$y[v_2, v_3, v_4]$	$\frac{1}{h_k^2}(y_{k-1} - y_k + h_k z_k)$
$y[v_0, v_1, v_2, v_3]$	$\frac{1}{h_k^3}(y_k - y_{k-1} - h_k z_{k-1}) - \frac{w_{k-1}}{2h_k}$
$y[v_1, v_2, v_3, v_4]$	$\frac{1}{h_k^3}(2(y_{k-1} - y_k) + h_k(z_{k-1} + z_k))$
$y[v_2, v_3, v_4, v_5]$	$\frac{w_k}{2h_k} - \frac{1}{h_k^3}(y_{k-1} - y_k + h_k z_k)$
$y[v_0, v_1, v_2, v_3, v_4]$	$\frac{w_{k-1}}{2h_k^2} + \frac{1}{h_k^4}(3(y_{k-1} - y_k) + h_k(2z_{k-1} + z_k))$
$y[v_1, v_2, v_3, v_4, v_5]$	$\frac{w_k}{2h_k^2} - \frac{1}{h_k^4}(3(y_{k-1} - y_k) + h_k(z_{k-1} + 2z_k))$
$y[v_0, v_1, v_2, v_3, v_4, v_5]$	$\frac{w_k - w_{k-1}}{h_k^3} - \frac{3}{h_k^5}(2(y_{k-1} - y_k) + h_k(z_{k-1} + z_k))$

Tabla 5.11. Expresiones desarrolladas de las diferencias divididas presentadas en la **Tabla 5.10.**

Ahora, ya se puede desarrollar (5.68) para obtener la expresión definitiva correspondiente al tramo \tilde{S}_k en $[t_{k-1}, t_k]$ del *pseudo-spline* quíntico \tilde{S} . La forma final de \tilde{S}_k se omite, ya que es trivial sustituir las diferencias divididas. Emplear *pseudo-splines* quínticos $\tilde{S} \in C^2$ garantiza poder imponer sistemáticamente en el esquema expuesto todas las condiciones de continuidad e interpolación de la **Tabla 5.9**. Extendiendo el razonamiento de la **Figura 5.1** a continuidad C^2 e interpolación sobre w_k , se tiene la **Figura 5.2**:

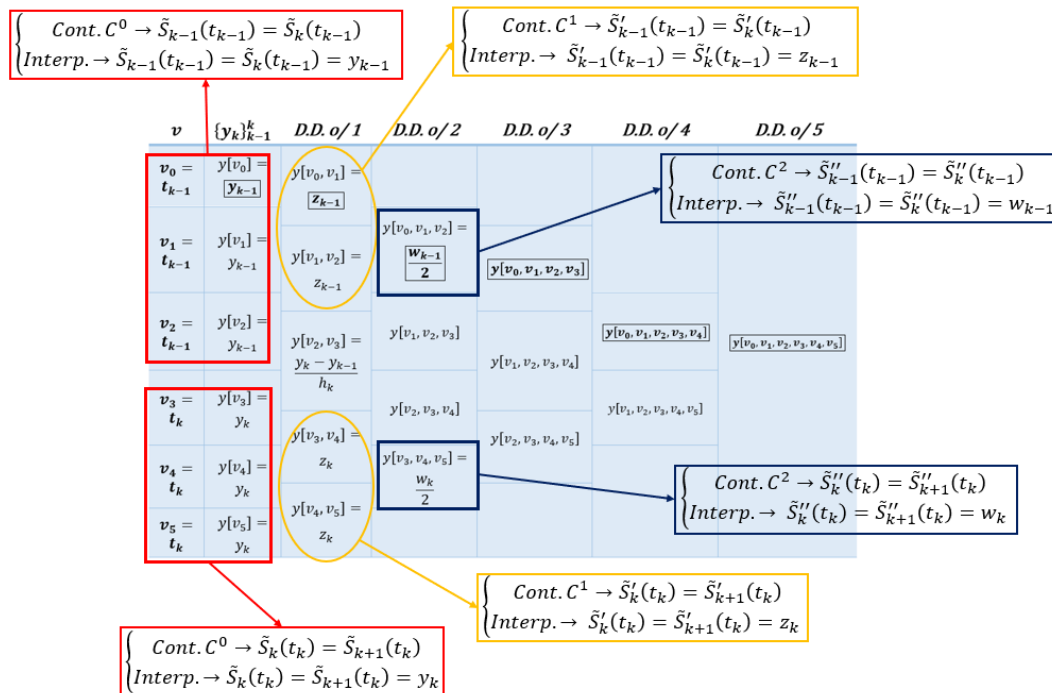


Figura 5.2. Imposición de las condiciones de continuidad y de interpolación con *pseudo-splines* quínticos.

5.3.4. Resumen de la implementación en *MATLAB*.

Al igual que con los *pseudo-splines* cuárticos interpoladores de y_k en (2.11) expuestos en la sección 5.2 de este capítulo, se ha realizado la correspondiente implementación en *MATLAB* de las estrategias de interpolación basadas en polinomios osculadores (*pseudo-splines* cúbicos y quínticos) tomando como referencia el *PVI* (2.6). En concordancia con el marco teórico expuesto para ellos, se aporta la **Tabla 5.12**, la cual recoge las funciones de interés ligadas a su estrategia (y evaluación) y que pueden ser consultadas en el **ANEXO III** de esta memoria.

<i>Estrategia</i>	<i>Sección memoria</i>	<i>Alias en MATLAB</i>
<i>Pseudo-splines</i> cúbicos (<i>Hermite</i>)	5.3.2	<i>"pspline3int"</i>
		<i>"eval_pspline3int"</i>
<i>Pseudo-splines</i> quínticos	5.3.3.	<i>"pspline5int"</i>
		<i>"eval_pspline5int"</i>

Tabla 5.12. Resumen de la implementación en *MATLAB* de estrategias de interpolación basadas en polinomios osculadores (*pseudo-splines* cúbicos y quínticos).





CAPÍTULO 6. Método del disparo para Problemas de Valores en la Frontera lineales.



6.1. Introducción a los Problemas de Valores en la Frontera (PVFs) de 2^o orden.

Hasta ahora se han abordado distintas estrategias de resolución de **Problemas de Valores Iniciales (PVI)** formulados en términos de **EDOs de 2^o orden**, tanto para el caso general no lineal (2.6) como para su particularización al problema lineal (2.7), pasando por un caso especial como es el independiente de la primera derivada y' (2.8). En el **Capítulo 3** se presentaban métodos numéricos de un paso *Runge-Kutta* y *Runge-Kutta-Nyström* para la obtención de soluciones en la malla discreta (2.11), mientras que en los **Capítulos 4 y 5** se aportaban estrategias de interpolación de dichas soluciones mediante el empleo de trazadores polinomiales (*splines* y *pseudo-splines*).

No obstante, el **PVI** no abarca la totalidad de problemas formulados en términos de **EDOs de 2^o orden**, pues se precisan dos condiciones que no necesariamente han de ser impuestas en el nodo inicial (lo que correspondería a la 1^a y la 2^a condición inicial) del intervalo considerado $[a, b]$, con $\{t_k\}_{k=0}^n$, siendo $t_0 = a$ y $t_n = b$.

Por regla general, la mayoría de problemas físicos e ingenieriles que dependen más de la posición que del tiempo se describen utilizando ecuaciones diferenciales con condiciones impuestas en más de un punto ([12] y [52]). Si el problema viene descrito por una **EDO** de 2^o orden y si dichas condiciones corresponden a los nodos inicial a y extremo b , se obtiene un **Problema de Valores en la Frontera o PVF**. Las condiciones a imponer en dichos nodos se denominan **condiciones de contorno**. A partir de estas condiciones se pueden realizar distinciones entre **PVFs**, las cuales se determinan en la sección 6.2.

Este capítulo concierne a la presentación de una metodología de resolución de **Problemas de Valores en la Frontera de 2^o orden basada en la aplicación del método del disparo para problemas estrictamente lineales (EDO del tipo (2.2))**. Este método se desarrolla en profundidad en la sección 6.4.

6.2. Clasificación de los PVF de 2^o orden según la tipología de condiciones de contorno.

Como ya se ha apuntado, las condiciones de contorno juegan un papel determinante a la hora de clasificar la naturaleza del **Problema de Valor en la Frontera de 2^o orden** a abordar. Surgen al materializar matemáticamente las hipótesis físicas en las que se sustenta el problema a resolver en sus extremos, por lo que deben de ser tratadas con especial atención para plantear un problema con rigor.

Considérense los **PVFs de 2^o orden** expresados en forma normal siguientes. El primero de ellos (6.1), está sujeto a **condiciones de contorno simples**, mientras que el segundo (6.2) presenta **condiciones de contorno más generales**.



$$\begin{cases} y''(x) = f(x, y(x), y'(x)) \\ y(a) = y_a, \quad y(b) = y_b \\ x \in [a, b] \end{cases}$$

Ecuación (6.1)

$$\begin{cases} y''(x) = f(x, y(x), y'(x)) \\ p_{11}y(a) + p_{12}y'(a) = p_{13}, \quad x \in [a, b] \\ p_{21}y(b) + p_{22}y'(b) = p_{23} \end{cases}$$

Ecuación (6.2)

Los coeficientes $\{p_{1j}\}_{j=1}^3$ y $\{p_{2j}\}_{j=1}^3$ de las condiciones de contorno de (6.2) son constantes y reales.

6.3. Existencia y unicidad de la solución de un PVF de 2º orden.

Una forma adecuada para comenzar el abordaje de un **PVF de 2º orden** consiste en analizar la existencia y unicidad de su solución. **Una gran cantidad de problemas del ámbito de la Ingeniería planteados matemáticamente exigen una solución única (previamente a plantear el problema final ya se han tenido en cuenta posibilidades alternativas pero rechazadas)**, la cual suele considerarse como la idónea u optimizada, cuando han sido tenidos en cuenta todos los criterios e hipótesis en los que se sustenta el análisis del fenómeno de estudio.

6.3.1. Teorema de existencia y unicidad de la solución. Condiciones de contorno simples.

El siguiente teorema establece las premisas que garantizan la existencia y unicidad de la solución para un **PVF de 2º orden** con condiciones de contorno del tipo expuesto en el problema (6.1). Supóngase que la función f de dicho problema es continua en el conjunto D , definido como (6.3):

$$D = \left\{ (x, y, y') \mid \forall x \in [a, b] \mid \text{con} \begin{cases} -\infty < y < \infty \\ -\infty < y' < \infty \end{cases} \right\}$$

Ecuación (6.3)

Garantizando también la continuidad en D de las derivadas parciales $\frac{\partial f}{\partial y}$ y $\frac{\partial f}{\partial y'}$, si se cumple:

- i. $\frac{\partial f}{\partial y}(x, y, y') > 0, \quad \forall (x, y, y') \in D, y$
- ii. Existe una constante M , con $\left| \frac{\partial f}{\partial y'}(x, y, y') \right| \leq M, \quad \forall (x, y, y') \in D,$

entonces existe la solución del **PVF (6.1)** y ésta es única.

Corolario. Particularización al PVF lineal.

Si ahora se considera el **PVF de 2º orden** con condiciones de contorno del tipo **(6.1)** en su particularización lineal, se tiene **(6.4)**:

$$\begin{cases} y''(x) = p(x)y'(x) + q(x)y(x) + r(x) \\ y(a) = y_a, \quad y(b) = y_b \\ x \in [a, b] \end{cases}$$

Ecuación **(6.4)**

El teorema expuesto anteriormente en para el caso genérico **(6.1)** con condiciones de contorno simples puede simplificarse si se cumplen:

- i. $p(x)$, $q(x)$ y $r(x)$ son continuas en $[a, b]$, y
- ii. $q(x) > 0$ en $[a, b]$.

Si se cumplen dichas condiciones, el **PVF lineal (6.4)** tiene solución y ésta es única.

6.3.2. Nociones sobre la existencia y unicidad de la solución. Condiciones de contorno generales.

La existencia y unicidad de la solución del **PVF de 2º orden y condiciones de contorno generales expuesto en (6.2)** no resulta ser tan evidente y demostrable como para el caso **(6.1)**, el cual ha sido expuesto en la sección anterior **6.3.1**. El lector puede profundizar en esta cuestión en las referencias **[46]** y **[53]**. A continuación se incluyen una serie de condiciones necesarias pero no suficientes para garantizar la existencia y la unicidad de la solución del **PVF lineal** dado por **(6.5)**, como particularización de **(6.2)**.

$$\begin{cases} y''(x) = p(x)y'(x) + q(x)y(x) + r(x) \\ p_{11}y(a) + p_{12}y'(a) = p_{13} \\ p_{21}y(b) + p_{22}y'(b) = p_{23} \end{cases}, \quad x \in [a, b]$$

Ecuación **(6.5)**

Si se consideran los coeficientes $\{p_{1j}\}_{j=1}^3$ y $\{p_{2j}\}_{j=1}^3$ valores reales, éstos deben satisfacer **(6.6)**:

$$p_{11}^2 + p_{12}^2 \neq 0 \quad \text{a)}$$

$$p_{21}^2 + p_{22}^2 \neq 0 \quad \text{b)}$$

Ecuación **(6.6)**

La expresión anterior resulta ser muy importante porque impide la imposición, en términos matemáticos, de condiciones de contorno “insensibles”. Considerarlas éstas igualadas a cero implicaría una trivialidad intrascendente para el **PVF**, es decir, $p_{11} = p_{12} = p_{13} = p_{21} = p_{22} = p_{23} = 0$.



Con el objeto de ahondar en la cuestión de la existencia y unicidad de la solución de (6.5), se reescribe (únicamente en esta sección) dicho problema en (6.7):

$$\begin{cases} q(x)y(x) = y''(x) - p(x)y'(x) + r(x) \\ B_1(y) = p_{13} \\ B_2(y) = p_{23} \end{cases}, \quad x \in [a, b] \quad \text{a)}$$

$$B_1(y) = p_{11}y(a) + p_{12}y'(a), \quad B_2(y) = p_{22}y(b) + p_{23}y'(b) \quad \text{b)}$$

Ecuación (6.7)

Las agrupaciones B_1 y B_2 de (6.7.b) son operadores lineales que satisfacen (6.8):

$$B_i(e_1y_1 + e_2y_2) = e_1B_i(y_1) + e_2B_i(y_2), \quad i = 1, 2$$

Ecuación (6.8)

Los coeficientes e_1 y e_2 son reales. Se tienen los dos siguientes teoremas, los cuales surgen de considerar inicialmente la ecuación diferencial homogénea dada por (6.9):

$$q(x)y(x) = g(x, y'(x), y''(x)) = y''(x) - p(x)y'(x)$$

Ecuación (6.9)

6.3.2.1. Teorema 1.

Si se definen dos soluciones y_1 e y_2 de (6.9), las cuales satisfacen (6.10):

$$B_1(y_1) = B_1(y_2) = 0 \quad \text{ó} \quad B_2(y_1) = B_2(y_2) = 0$$

Ecuación (6.10)

Entonces el conjunto $\{y_1, y_2\}$ es linealmente independiente. Equivalentemente, si el conjunto $\{y_1, y_2\}$ es linealmente dependiente, surge (6.11):

$$B_1^2(y_1) + B_2^2(y_2) \neq 0 \quad \text{y} \quad B_2^2(y_1) + B_1^2(y_2) \neq 0$$

Ecuación (6.11)

6.3.2.1. Teorema 2.

Las siguientes premisas son equivalentes, esto es, o son todas ciertas o todas falsas:

- i. Existe un conjunto fundamental de soluciones $\{y_1, y_2\}$ para la ecuación diferencial homogénea dada por (6.9), verificándose (6.12):

$$B_1(y_1)B_2(y_2) - B_1(y_2)B_2(y_1) \neq 0$$

Ecuación (6.12)

- La definición de agrupaciones B_i es la dada por (7.b).
- ii. Para cada función g continua y un par de constantes reales (p_{13}, p_{23}) el **Problema de Valores en la Frontera de 2º orden (6.7)** tiene **solución única**.
 - iii. El **PVF** homogéneo dado por (6.13):

$$\begin{cases} q(x)y(x) = y''(x) - p(x)y'(x) \\ B_1(y) = p_{13} \\ B_2(y) = p_{23} \end{cases}, \quad x \in [a, b] \quad \text{a)}$$

$$B_1(y) = p_{11}y(a) + p_{12}y'(a), \quad B_2(y) = p_{22}y(b) + p_{23}y'(b) \quad \text{b)}$$

Ecuación (6.13)

- únicamente tiene solución trivial $y = 0$.
- iv. La ecuación diferencial homogénea (6.9) presenta dos soluciones linealmente independientes y_1 e y_2 , las cuales cumplen (6.14):

$$B_1(y_1) = 0 \quad y \quad B_2(y_2) = 0$$

Ecuación (6.14)

6.4. El método del disparo para PVFs de 2º orden lineales.

Una vez determinadas las condiciones en virtud de las cuales existe una solución única al **Problema de Valores en la Frontera** de 2º orden, se puede presentar la **metodología de resolución numérica** propuesta para el mismo. Es importante destacar que este trabajo se centra únicamente en la resolución de los **PVF** del tipo (6.4) y (6.5), considerando los distintos tipos de condiciones de contorno expuestas en 6.2.

El **método del disparo lineal** constituye la técnica más utilizada para abordar la resolución numérica de estos problemas. Este algoritmo permite **desdoblar un PVF en varios PVI** asociados, los cuales posteriormente se combinan linealmente para obtener la solución aproximada al primer problema. Existen multitud de trabajos de investigación relativos al empleo de este método, algunos de los cuales pueden ser consultados en [54-57].

6.4.1. Método del disparo lineal. Condiciones de contorno simples.

La solución discreta aproximada del **PVF de 2º orden lineal** del tipo (6.4) puede obtenerse considerando los siguientes **PVI** en los que se desdobla el problema original (6.15) y (6.16):

$$1 \rightarrow \begin{cases} y_1''(x) = p(x)y_1'(x) + q(x)y_1(x) + r(x) \\ y_1(a) = y_a, \quad y_1'(a) = 0 \\ x \in [a, b] \end{cases}$$

Ecuación (6.15)

$$2 \rightarrow \begin{cases} y_2''(x) = p(x)y_2'(x) + q(x)y_2(x) \\ y_2(a) = 0, \quad y_2'(a) = 1 \\ x \in [a, b] \end{cases}$$

Ecuación (6.16)

Habiendo garantizado en **2.2.1** la existencia y unicidad de la solución de ambos **PVIs**, se emplea algún método numérico expuesto en el **Capítulo 3** (o la solución exacta si existe para cada problema) para obtener la solución discreta aproximada $y_{i,k}$ de cada problema dada por (6.17), en los nodos de la malla (2.11):

$$\{x_k = t_k\}_{k=0}^n \rightarrow \begin{cases} t_0 = a \\ t_n = b \end{cases} \quad \left| \quad \{y_{i,k}\}_{k=0}^n \quad \right| \quad i = 1, 2$$

Ecuación (6.17)

La interpolación de la solución discreta de cada **PVI** asociado se realiza empleando las estrategias abordadas en los **Capítulos 4 y 5**, respectivamente. Una vez obtenidas $y_1(x)$ e $y_2(x)$, se construye la solución del **PVF** $y(x)$ con la combinación lineal dada por (6.18):

$$y(x) = y_1(x) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2(x), \quad y_2(t_n) \neq 0$$

Ecuación (6.18)

Es sencillo demostrar la razón de ser de la expresión anterior. Considérense las derivadas 1ª y 2ª dadas, respectivamente, por (6.19) y (6.20):

$$y'(x) = y_1'(x) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2'(x), \quad y_2(t_n) \neq 0$$

Ecuación (6.19)

$$y''(x) = y_1''(x) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2''(x), \quad y_2(t_n) \neq 0$$

Ecuación (6.20)

Al sustituir para $y_1''(x)$ e $y_2''(x)$ en la expresión de la 2ª derivada $y''(x)$ (6.20), se obtiene:

$$y''(x) = p(x)y_1'(x) + q(x)y_1(x) + r(x) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) (p(x)y_2'(x) + q(x)y_2(x)) =$$

$$\begin{aligned}
 &= p(x) \left(y_1'(x) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2'(x) \right) + q(x) \left(y_1(x) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2(x) \right) \\
 &\quad + r(x) = p(x)y'(x) + q(x)y(x) + r(x)
 \end{aligned}$$

Además, pueden verificarse también las condiciones de contorno utilizando:

$$y(a) = y(t_0) = y_1(t_0) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2(t_0) = y_0 + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) \cdot 0 = y_a$$

y

$$y(b) = y(t_n) = y_1(t_n) + \left(\frac{y_b - y_1(t_n)}{y_2(t_n)} \right) y_2(t_n) = y_1(t_n) + y_b - y_1(t_n) = y_b$$

Es muy importante recordar que $t_0 = a$ y que $t_n = b$. La ecuación (6.18) se reescribe en (6.21) utilizando las expresiones trazadores polinomiales empleados para interpolar las soluciones discretas $y_{i,k}$ de cada *PVI* en los que el *PVF* se desdobra 1S y 2S , y atendiendo a $y_1(t_n)$ e $y_2(t_n)$, aproximadas respectivamente por $y_{1,n}$ e $y_{2,n}$.

$$y(x) \cong {}^1S(x) + \left(\frac{y_b - y_{1,n}}{y_{2,n}} \right) {}^2S(x), \quad y_{2,n} \neq 0$$

Ecuación (6.21)

En otras palabras, la aproximación a la solución continua del *PVF* se obtiene calculando las soluciones aproximadas continuas de los *PVIs* asociados con técnicas conocidas para ello. Posteriormente, el desdoble realizado permite obtener una combinación lineal de aproximaciones (6.21) que aproxima la solución de interés $y(x)$. La *Figura 6.1* ilustra el concepto empleado. El “disparo” golpea el objetivo después de un disparo de prueba.

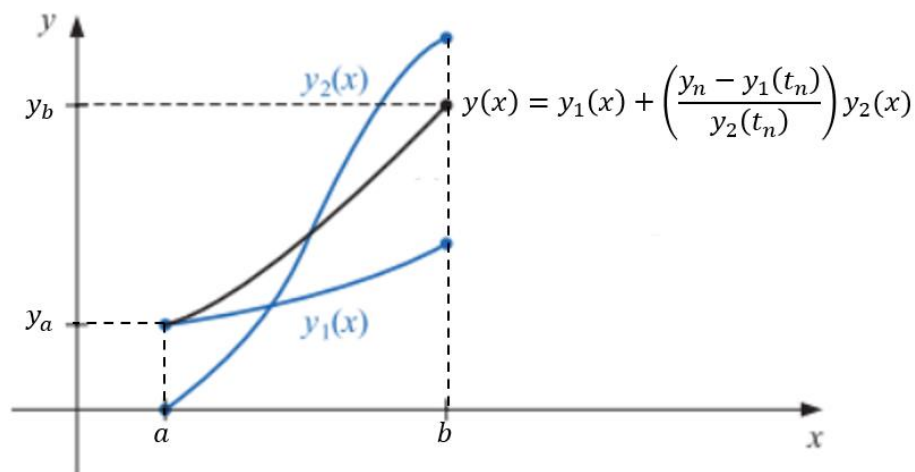


Figura 6.1. Método del disparo para problemas lineales. Condiciones de contorno simples.

6.4.1.1. Ejemplo. Algoritmo de resolución con nodos equiespaciados.

El objeto de esta sección concierne a la presentación de un algoritmo que implementa el método del disparo lineal para abordar la resolución aproximada del *PVF* (6.4), el cual está sujeto a condiciones de contorno simples. La metodología propuesta se basa en obtener aproximaciones a la función solución y (y_k) y a su primera derivada y' (z_k) en los nodos de la malla (2.11). La **Tabla 6.1** presenta las entradas, salidas y los pasos a realizar que componen el algoritmo ejemplo.

$$ENTRADAS \rightarrow \begin{cases} \text{Extremos del intervalo: } [a, b] \\ \text{Condiciones de contorno simples: } \begin{cases} y(a) = y_a \\ y(b) = y_b \end{cases} \\ \text{Número de subintervalos a considerar: } N \end{cases}$$

SALIDA	Solución aproximada y_k y 1ª derivada z_k $\forall k \in 0, 1, \dots, N$
PROCEDIMIENTO SISTEMÁTICO A APLICAR	
Paso 1	Calcular el tamaño de paso uniforme: $h = \frac{b-a}{N} = \frac{t_N - t_0}{N}$ A continuación, tomar condiciones iniciales de los <i>PVI</i> asociados a (6.4) ((6.15) y (6.16)): $y_{1,0} = y_a, \quad z_{1,0} = 0, \quad y_{2,0} = 0, \quad z_{2,0} = 1$
Paso 2	Para $k = 0, 1, \dots, N - 1$ hacer los pasos 3 y 4
Paso 3	Discretización de la variable independiente x según: $x_k = t_k = a + k \cdot h, \quad \{t_k\}_{k=0}^{N-1}$
Paso 4	Aplicación de un método numérico apropiado para calcular la solución discreta aproximada de cada <i>PVI</i> (6.15) y (6.16) por separado y sus primeras derivadas: $y_{1,k+1}, \quad z_{1,k+1}, \quad y_{2,k+1}, \quad z_{2,k+1}$
Paso 5	Cálculo de la salida en t_0 : $y_0 = y_a, \quad z_0 = \frac{y_b - y_{1,N}}{y_{2,N}}$ SALIDA \rightarrow (a, y_0, z_0)
Paso 6	Cálculo de la salida en $\{t_k\}_{k=1}^N$: $\vec{Y} = \{y_{1,k} + z_0 \cdot y_{2,k}\}_{k=1}^N, \quad \vec{Z} = \{z_{1,k} + z_0 \cdot z_{2,k}\}_{k=1}^N$ SALIDA \rightarrow $(\{t_k\}_{k=1}^N, \vec{Y}, \vec{Z})$
Paso 7	Mostrar: $(a, y_0, z_0) \cup (\{t_k\}_{k=1}^N, \vec{Y}, \vec{Z})$
Paso 8	PARAR. PROCEDIMIENTO COMPLETO.

Tabla 6.1. Algoritmo computacional del método del disparo lineal con condiciones de contorno simples.

Nótese que el **Paso 5** es de libre elección, siempre y cuando se seleccione un método numérico apropiado capaz de proporcionar aproximaciones a la función solución de (6.4), y , y a su primera derivada y' (ver **Capítulo 3**).

Si se interpolan las soluciones discretas aproximadas de cada **PVI** asociado en dicho **Paso 5** utilizando los trazadores polinomiales expuestos en los **Capítulos 4 y 5**, el procedimiento sigue siendo el mismo, aunque ahora la salida del **Paso 7** corresponde a S y S' , con dichos trazadores ya evaluados convenientemente.

Para finalizar la exposición del algoritmo de la **Tabla 6.1**, conviene resaltar la importancia de los posibles errores de redondeo que pueden presentarse. Si $y_1(x)$ (solución de (6.15)) aumenta rápidamente conforme x recorre el intervalo $[a, b]$ de forma progresiva, $y_{1,N} = y_1(b)$. Al mismo tiempo, si la magnitud de y_b es pequeña en comparación con $y_{1,N}$, el término z_0 es aproximadamente (6.22):

$$z_0 = \frac{y_b - y_{1,N}}{y_{2,N}} \cong -\frac{y_{1,N}}{y_{2,N}}$$

Ecuación (6.22)

Se reescriben los cálculos del **Paso 6** en (6.23):

$$\vec{Y} = \{y_{1,k} + z_0 \cdot y_{2,k}\}_{k=1}^N \cong \left\{ y_{1,k} - \frac{y_{1,N}}{y_{2,N}} \cdot y_{2,k} \right\}_{k=1}^N \quad a)$$

$$\vec{Z} = \{z_{1,k} + z_0 \cdot z_{2,k}\}_{k=1}^N \cong \left\{ z_{1,k} - \frac{y_{1,N}}{y_{2,N}} \cdot z_{2,k} \right\}_{k=1}^N \quad b)$$

Ecuación (6.23)

La consecuencia inmediata implica una posible pérdida de cifras significativas por cancelación. No obstante, puede monitorizarse de forma sencilla el comportamiento de y_1 observando si su aproximación discreta $y_{1,k}$ aumenta rápidamente al recorrerse el intervalo $[a, b]$.

Si lo anterior resulta problemático, entonces la alternativa es aplicar el método del disparo lineal en la resolución de **PVF** dado por (6.5) **hacia atrás**, también conocido como **método del disparo inverso**. Reescribiendo los **PVI** (6.15) y (6.16) asociados a (6.4), se tienen (6.24) y (6.25):

$$1 \rightarrow \begin{cases} y_1''(x) = p(x)y_1'(x) + q(x)y_1(x) + r(x) \\ y_1(b) = y_b, \quad y_1'(b) = 0 \\ x \in [a, b] \end{cases}$$

Ecuación (6.24)



$$2 \rightarrow \begin{cases} y_2''(x) = p(x)y_2'(x) + q(x)y_2(x) \\ y_2(b) = 0, \quad y_2'(b) = 1 \\ x \in [a, b] \end{cases}$$

Ecuación (6.25)

Por regla general, si $y_{1,k}$ e $y_{2,k}$ son $O(h^n)$ aproximaciones para $y_1(t_k)$ e $y_2(t_k)$, respectivamente, y_k será una aproximación $O(h^n)$ para $y(t_k)$ (6.26).

$$|y_{1,k} - y(t_k)| \leq Kh^n \left| 1 + \frac{y_{2,k}}{y_{1,N}} \right|$$

Ecuación (6.26)

6.4.1.2. Condiciones de contorno simples. Particularización a $y_2(b) = 0$.

El lector habrá podido percatarse de que el planteamiento del método del disparo lineal hecho anteriormente sólo es válido para el caso de $y_2(b) = y_2(t_n) \neq 0$, pues (6.18) tiene su razón de ser sólo y cuando el coeficiente de la combinación lineal que determina $y(x)$ y que acompaña a $y_2(x)$ existe, algo que sucede únicamente para valores no nulos del denominador $y_2(t_n)$. Si lo anterior no es posible, al ser $y_2(t_n) = 0$, puede desdoblarse el **PVF de 2º orden lineal** del tipo (6.4) en tres **PVIs** (6.27, 6.28 y 6.29). Dichos problemas asociados al original son los siguientes:

$$1 \rightarrow \begin{cases} y_1''(x) = p(x)y_1'(x) + q(x)y_1(x) + r(x) \\ y_1(a) = y_a, \quad y_1'(a) = 0 \\ x \in [a, b] \end{cases}$$

Ecuación (6.27)

$$2 \rightarrow \begin{cases} y_2''(x) = p(x)y_2'(x) + q(x)y_2(x) \\ y_2(a) = 0, \quad y_2'(a) = 1 \\ x \in [a, b] \end{cases}$$

Ecuación (6.28)

$$3 \rightarrow \begin{cases} y_3''(x) = p(x)y_3'(x) + q(x)y_3(x) \\ y_3(a) = 1, \quad y_3'(a) = 0 \\ x \in [a, b] \end{cases}$$

Ecuación (6.29)

En virtud de las mismas hipótesis y estrategias de resolución citadas en 6.4.1 y desarrolladas en este trabajo, se pueden obtener las tres soluciones de los **PVIs** asociados que dan lugar a la solución del **PVF** original $y(x)$. Antes de proceder, considérese $y(x)$ como superposición del problema **no homogéneo** 1 (6.27), cuya solución viene dada por $y_1(x)$, y una combinación lineal de

soluciones correspondientes a los problemas homogéneos 2 (6.28) y 3 (6.29), con soluciones respectivas $y_2(x)$ e $y_3(x)$. Se plantea la expresión (6.30):

$$y(x) = y_1(x) + c_2 y_2(x) + c_3 y_3(x)$$

Ecuación (6.30)

Los coeficientes c_2 y c_3 son reales. Razonando debidamente sobre la expresión anterior y las condiciones iniciales de los *PVI*s asociados al *PVF*, se tiene (6.31):

$$y(a) = y(t_0) = y_a = y_1(t_0) + c_2 y_2(t_0) + c_3 y_3(t_0) = y_0 + c_3 \rightarrow \boxed{c_3 = 0} \quad a)$$

$$y(b) = y(t_n) = y_b = y_1(t_n) + c_2 y_2(t_n) + c_3 y_3(t_n) = y_1(t_n) \rightarrow \boxed{c_2 = 0} \quad b)$$

Ecuación (6.31)

El coeficiente c_3 se anula despejando de (6.31.a). Al mismo tiempo, c_2 puede tomar cualquier valor tras plantear (6.31.b) y deducir que $y_1(x)$ es una solución que satisface ambas condiciones, siendo además única en relación a lo ya comentado en 6.3.1. Por comodidad se toma $c_2 = 0$. Finalmente, se deduce que, en el supuesto de $y_2(t_n) = 0$, la solución $y(x)$ del *PVF* lineal del tipo (6.4) mediante el método del disparo coincide con la solución del *PVI* asociado 1 no homogéneo (6.27).

El procedimiento de obtención de la **solución aproximada continua** sigue pasando por emplear las herramientas presentadas en el **Capítulo 3**, calculando la solución discreta $\{y_{1,k}\}_{k=0}^n$ mediante el método numérico seleccionado en la malla (2.11), para posteriormente construir el trazador polinomial que interpola a dicha solución discreta (**Capítulos 4 y 5**) y que es aproximación de la solución del *PVI* (1S) (6.32).

$$y(x) = y_1(x) \cong {}^1S(x)$$

Ecuación (6.32)

6.4.2. Método del disparo lineal. Condiciones de contorno generales.

La resolución del **Problema de Valores en la Frontera de 2º orden lineal** del tipo (6.4) abordada en 6.4.1 mediante la aplicación del **método del disparo lineal** no tiene en cuenta todos los **PVF de 2º orden lineales** que pueden surgir. De hecho, se trata de la particularización más sencilla que puede hacerse. A continuación, se desarrolla el método del disparo lineal que corresponde aplicar si el **PVF lineal** a resolver presenta condiciones de contorno del tipo (6.2), conocidas como **condiciones de contorno generales** [58]. Se consideran los siguientes tres *PVI*s (6.33, 6.34 y 6.35), asociados al *PVF* original, en los cuales se desdobra este último:

$$1 \rightarrow \begin{cases} y_1''(x) = p(x)y_1'(x) + q(x)y_1(x) + r(x) \\ y_1(a) = 0, \quad y_1'(a) = 0 \\ x \in [a, b] \end{cases}$$

Ecuación (6.33)

$$2 \rightarrow \begin{cases} y_2''(x) = p(x)y_2'(x) + q(x)y_2(x) \\ y_2(a) = 0, \quad y_2'(a) = 1 \\ x \in [a, b] \end{cases}$$

Ecuación (6.34)

$$3 \rightarrow \begin{cases} y_3''(x) = p(x)y_3'(x) + q(x)y_3(x) \\ y_3(a) = 1, \quad y_3'(a) = 0 \\ x \in [a, b] \end{cases}$$

Ecuación (6.35)

Puede considerarse la solución del **PVF** original como la superposición del problema **no homogéneo 1 (6.34)**, con solución $y_1(x)$, y una combinación lineal de soluciones correspondientes a los problemas **homogéneos 2 (6.34)** y **3 (6.35)**, con soluciones respectivas $y_2(x)$ e $y_3(x)$. Se plantea (6.36):

$$y(x) = y_1(x) + d_2y_2(x) + d_3y_3(x)$$

Ecuación (6.36)

Antes de proceder con la solución continua aproximada de cada **PVI** asociado por separado dada por el trazador polinomial correspondiente, es preciso obtener los **coeficientes d_2 y d_3 de la combinación lineal de soluciones discretas a los problemas homogéneos**. Para ello, se sustituye en las condiciones de contorno generales del tipo (6.2) los valores aproximados de y (y_k) e y' (z_k) en los nodos de (2.11), tanto en t_0 como en t_n , obtenidos éstos mediante el método numérico elegido del **Capítulo 3**. Se tiene (6.37):

$$y(a) = y(t_0) \cong y_0 = d_3 \quad \text{a)}$$

$$y'(a) = y'(t_0) \cong z_0 = d_2 \quad \text{b)}$$

$$y(b) = y(t_n) \cong y_n = y_{1,n} + d_2y_{2,n} + d_3y_{3,n} \quad \text{c)}$$

$$y'(b) = y'(t_n) \cong z_n = z_{1,n} + d_2z_{2,n} + d_3z_{3,n} \quad \text{d)}$$

Ecuación (6.37)

Determinadas de forma aproximada $y(t_0)$, $y'(t_0)$, $y(t_n)$ e $y'(t_n)$, se plantea un **Sistema 2x2 (6.38)** cuyas incógnitas son d_2 y d_3 . Manipulando las condiciones de contorno, se tiene:

$$\begin{cases} p_{11}d_3 + p_{12}d_2 = p_{13} \\ (p_{21}y_{2,n} + p_{22}z_{2,n})d_2 + (p_{21}y_{3,n} + p_{22}z_{3,n})d_3 = p_{23} - (p_{21}y_{1,n} + p_{22}z_{1,n}) \end{cases}$$

Ecuación (6.38)

Las expresiones explícitas de los coeficientes d_2 y d_3 tras resolver el sistema anterior se indican en (6.39):

$$d_2 = \frac{p_{13}(p_{21}y_{3,n} + p_{22}z_{3,n}) - p_{11}(p_{23} - p_{21}y_{1,n} - p_{22}z_{1,n})}{p_{12}(p_{21}y_{3,n} + p_{22}z_{3,n}) - p_{11}(p_{21}y_{2,n} + p_{22}z_{2,n})} \quad a)$$

$$d_3 = \frac{-p_{13}(p_{21}y_{2,n} + p_{22}z_{2,n}) + p_{12}(p_{23} - p_{21}y_{1,n} - p_{22}z_{1,n})}{p_{12}(p_{21}y_{3,n} + p_{22}z_{3,n}) - p_{11}(p_{21}y_{2,n} + p_{22}z_{2,n})} \quad b)$$

Ecuación (6.39)

Ahora ya se puede construir la solución aproximada continua del PVF lineal de 2º orden (6.5) $y(x)$ mediante las soluciones aproximadas con trazadores polinomiales interpoladores (Capítulos 4 y 5) de cada PVI asociado al PVF 1S , 2S y 3S . Particularizando (6.36) a estas aproximaciones conocidas, se tiene (6.40):

$$y(x) \cong {}^1S(x) + d_2 {}^2S(x) + d_3 {}^3S(x)$$

Ecuación (6.40)





CAPÍTULO 7.

Experimentos numéricos.



7.1. Introducción y estructura del capítulo.

El extenso marco teórico presentado en los **Capítulos 3, 4, 5 y 6**, concerniente a las estrategias de resolución de problemas formulados en términos de **EDOs** de 2º orden ha de ser convenientemente complementado mediante experimentos que permitan **ensayar las estrategias de interpolación y los métodos para la obtención de soluciones discretas en la malla (2.11) planteados**. Dado el contexto en el que se desarrolla este trabajo, **se plantean problemas (tanto PVI como PVF) con aplicaciones directas a la Ingeniería y extraídos de la literatura científica**, con los cuales se pretende determinar el comportamiento (idóneo o no) de todas las líneas de actuación expuestas. **También se experimenta con problemas de prueba surgidos en el desarrollo de este trabajo y de un carácter más puramente matemático.**

La estructura adoptada para este capítulo utiliza todos los problemas planteados y documentados para la realización de experimentos numéricos en **MATLAB**, presentando en cada uno de ellos una **demostración teórico-práctica del fenómeno de estudio acompañada de referencias bibliográficas**, con la cual se complementa la obtención de soluciones posterior, dándole a estas últimas la realidad física adecuada en base a su marco teórico y aportando conclusiones sobre las estrategias de resolución adoptadas.

En el **ANEXO IV** pueden consultarse todos los códigos de **MATLAB** generados para la realización de cada uno de los experimentos numéricos que aquí se desarrollan. La **Tabla 7.1** indica la correspondencia entre cada código con el experimento al que pertenece, junto con la sección de este capítulo en la que es objeto de estudio.

<i>Experimento</i>	<i>Sección</i>	<i>Breve descripción</i>	<i>Código MATLAB (ANEXO IV)</i>
1	7.2	Vibraciones lineales de 1gdl	"PVI_VibMec_Lineal_1gdl"
2	7.3	Problemas de prueba. Órdenes de convergencia empíricos	"PVI_Ordenes_Metodos"
3	7.4	Problema de prueba. Ensayos con <i>splines</i> cúbicos	"PVI_s3int"
4	7.5	Deflexión de una viga biapoyada	PVF_Viga_Biapoyada"
5	7.6	Perfil de temperaturas en una aleta anular hiperbólica	"PVF_Aleta_Anular_Hiperbolica"
6	7.7	Respuesta temporal de un péndulo no lineal	"PVI_Pendolo_No_Lineal"
7	7.8	PVI rígido	"PVI_EDO_Stiff"

Tabla 7.1. Resumen de los experimentos numéricos planteados. **Capítulo 7.**

Algunos de los problemas planteados en la tabla anterior tienen solución analítica. Esto no debe entenderse como algo negativo que inutilice las estrategias planteadas en el trabajo, sino como algo positivo a la hora de verificar la idoneidad de una u otra, pues en un caso de estudio más genérico e intrincado en el que no se disponga de soluciones exactas, el disponer de datos favorables acerca del comportamiento de los métodos a aplicar, constituye un factor de decisión importante a la hora de elegir uno u otro.

Expuestas todas las consideraciones anteriores, se procede con los experimentos numéricos en el mismo orden de la **Tabla 7.1**.

7.2. Experimento numérico 1. Vibraciones lineales de 1 grado de libertad.

Uno de los campos de la Física más interesantes en lo que a estudio de ecuaciones diferenciales se refiere es las **Vibraciones Mecánicas**. Dicho campo resulta ser determinante en Ingeniería, pues los fenómenos vibratorios han de ser conocidos minuciosamente si se pretende hacer un uso adecuado de los modelos matemáticos derivados de ellos. Por encima de todos los problemas (de carácter específico o más generalista) que pueden plantearse dentro de este área de conocimiento, destaca la respuesta temporal del **sistema Masa-Muelle-Amortiguador de 1 grado de libertad**, o coloquialmente “**sistema mkc**”, esquematizado en la **Figura 7.1**. Las referencias bibliográficas [59-64] muestran el fundamento y el enorme interés de este problema.

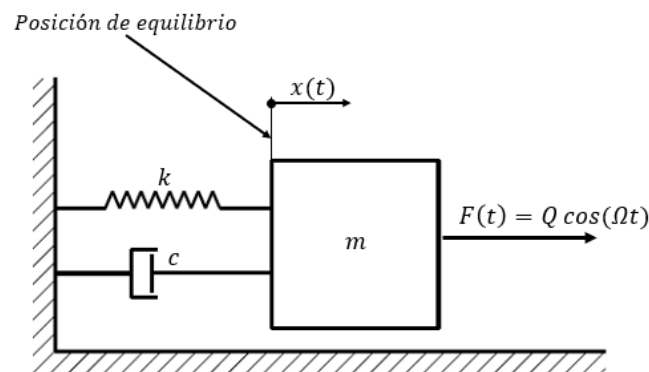


Figura 7.1. Sistema vibratorio “mkc” forzado sometido a una excitación armónica.

Se considera un **muelle elástico lineal** de rigidez constante k (que cumple la *Ley de Hooke*), un **amortiguador ideal** con coeficiente de amortiguamiento viscoso c y una **masa m** a la que se unen tanto el citado muelle como el amortiguador. Para obtener un problema lo más completo posible, también se considera una **excitación armónica $F(t)$** con frecuencia de excitación Ω . Se pretende obtener la evolución oscilatoria temporal de la variable vibratoria $x(t)$ correspondiente al problema forzado (al introducirse $F(t)$), mediante la resolución de la siguiente ecuación diferencial (7.1):

$$\ddot{x}(t) = -\frac{c}{m}\dot{x}(t) - \frac{k}{m}x(t) + \frac{Q}{m}\cos(\Omega t)$$

Ecuación (7.1)

Se trata de una **EDO lineal de 2º orden con coeficientes constantes**. Se puede introducir la siguiente notación (7.2):

$$\omega^2 = \frac{k}{m}, \quad n = \frac{c}{2m}, \quad q = \frac{Q}{m}$$

Ecuación (7.2)

La constante ω representa la **frecuencia natural del sistema** de la **Figura 7.1**. Sustituyendo en (7.1), se obtiene (7.3):

$$\ddot{x}(t) = -2n\dot{x}(t) - \omega^2 x(t) - q \cos(\Omega t)$$

Ecuación (7.3)

La solución analítica puede expresarse como **superposición de la solución general del problema homogéneo $x_h(t)$ y una solución particular del problema no homogéneo $x_p(t)$ (7.4)**, tal y como fue indicado en (2.4) para **EDOs lineales de 2º orden**.

$$x(t) = x_h(t) + x_p(t) \rightarrow \begin{cases} x_h(t) = e^{-nt}(C_1 \cos(\omega_d t) + C_2 \sin(\omega_d t)) \\ x_p(t) = M \cos(\Omega t) + N \sin(\Omega t) \end{cases} \quad \text{a)}$$

$$M = \frac{q(\omega^2 - \Omega^2)}{(\omega^2 - \Omega^2)^2 + 4n^2\Omega^2}, \quad N = \frac{2qn\Omega}{(\omega^2 - \Omega^2)^2 + 4n^2\Omega^2}, \quad \omega_d = \sqrt{\omega^2 - n^2} \quad \text{b)}$$

$$x(t) = e^{-nt}(C_1 \cos(\omega_d t) + C_2 \sin(\omega_d t)) + M \cos(\Omega t) + N \sin(\Omega t) \quad \text{c)}$$

Ecuación (7.4)

La función $x_h(t)$ representa las vibraciones libres amortiguadas a una frecuencia ω_d , mientras que $x_p(t)$ muestra los efectos de la excitación armónica $F(t)$ a una frecuencia Ω . La demostración de (7.4) puede consultarse con detalle en [59]. **C_1 y C_2 son constantes a determinar por las condiciones iniciales del problema**. La primera derivada de (7.4.c) viene dada por (7.5):

$$\dot{x}(t) = e^{-nt}((C_2\omega_d - nC_1)\cos(\omega_d t) - (C_2 + C_1\omega_d)\sin(\omega_d t)) + \Omega(N\cos(\Omega t) - M\sin(\Omega t))$$

Ecuación (7.5)

Pueden utilizarse los valores numéricos (en unidades del *Sistema Internacional*) propuestos en la **Tabla 7.2** para obtener un problema particularizado.

<i>Magnitud</i>	<i>Descripción</i>	<i>Valor</i>
m	Masa concentrada	1 kg
k	Rigidez del muelle elástico-lineal	5 N/m
c	Coefficiente de amortiguamiento viscoso	0.5 Ns/m
Q	Amplitud de la excitación armónica	10 N
Ω	Frecuencia de excitación	3 rad/s
$x(0)$	Condición inicial en desplazamiento	0 m
$\dot{x}(0)$	Condición inicial en velocidad	0 m/s

Tabla 7.2. Valores numéricos considerados para particularizar el **PVI** surgido de la **EDO (7.4)**.

También deben considerarse los valores de la **Tabla 7.3**, correspondientes a las expresiones agrupadas de **(7.2)** y **(7.4.b)**, junto con las constantes C_1 y C_2 .

<i>Magnitud</i>	<i>Valor</i>
ω	$\sqrt{5}$ rad/s
n	$1/4$ s ⁻¹
q	10 N/m
M	$-160/73$
N	$60/73$
ω_d	$\sqrt{79}/4$ rad/s
C_1	$160/73$
C_2	$-560\sqrt{79}/5767$

Tabla 7.3. Valores numéricos de las agrupaciones consideradas para particularizar el **PVI** surgido de la **EDO (7.4)**.

Con todo lo anterior, se obtiene un **Problema de Valores Iniciales del tipo (2.7)**, formulado en términos de una **EDO de 2º orden lineal de coeficientes constantes (7.6)**:

$$\begin{cases} \ddot{x}(t) = -\frac{1}{2}\dot{x}(t) - 5x(t) - 10 \cos(3t) \\ x(0) = 0 \text{ m}, \quad \dot{x}(0) = 0 \text{ m/s} \\ t \in [0,3] \text{ s} \end{cases}$$

Ecuación **(7.6)**

Cabe destacar que se ha empleado la notación de *Newton* para las derivadas, tal y como suele hacerse en Ingeniería cuando se está resolviendo un problema temporal (ver sección **2.1.1** del **Capítulo 2**). La solución analítica del **PVI** se

conoce, y es la dada por (7.7), particularización de (7.4.c) con los valores numéricos de las *Tablas 7.2* y *7.3*:

$$x(t) = \frac{60}{73} \left[e^{\left(-\frac{t}{4}\right)} \cdot \left(\frac{8}{3} \cos\left(\frac{\sqrt{79}}{4} t\right) - \frac{28\sqrt{79}}{237} \sin\left(\frac{\sqrt{79}}{4} t\right) \right) - \frac{8}{3} \cos(3t) + \sin(3t) \right]$$

Ecuación (7.7)

7.2.1. Solución exacta y numérica. Estudio del orden empírico de los métodos discretos.

Al disponer el problema planteado en esta sección de solución exacta y fácilmente calculable (7.6) con los datos particularizados de las *Tablas 7.2* y *7.3* se considera ésta una **oportunidad idónea para realizar un estudio empírico sobre los órdenes de convergencia que presentan algunos de los métodos numéricos para la obtención de soluciones discretas aproximadas** expuestos en el *Capítulo 3* e implementados en *MATLAB* en el *ANEXO I*.

Considérense las **mallas equiespaciadas representadas por sus respectivos pasos** uniformes dadas por la *Tabla 7.4* y construidas en base a (2.13), las cuales discretizan el dominio temporal continuo de (7.6).

<i>Identificador de malla (ID)</i>	<i>Paso uniforme (h)</i>	<i>Nº de subintervalos (n)</i>
1	0.1	30
2	0.05	60
3	0.025	120
4	0.0125	240
5	0.00625	480

Tabla 7.4. Mallas equiespaciadas progresivamente densificadas en el intervalo [0,3] (s).

Se observa que el densificado progresivo corresponde a una **disminución en potencias de base 2**. Se estima conveniente construir las mallas de esta manera en vez de con potencias de base 10 dados los órdenes de magnitud que presenta el *PVI* de estudio. Un densificado excesivo tan sólo conduce a un incremento del coste computacional asociado a las operaciones sin mejoras en los resultados (más tiempo de computación). Por tanto, se tiene (7.8):

$$h_{ID} = \frac{1}{2} h_{ID-1}, \quad n_{ID} = \frac{1}{2} n_{ID-1}, \quad ID = 2, \dots, 5$$

Ecuación (7.8)

7.2.1.1. Solución analítica $x(t)$ y primera derivada $\dot{x}(t)$.

La *Figura 7.2* muestra la solución analítica $x(t)$ del *PVI (7.5)* y la primera derivada $\dot{x}(t)$ de dicha solución en los nodos de la primera malla definida en la *Tabla 7.4* ($h = 0.1$). Se emplea la interpolación lineal por defecto de *MATLAB*.

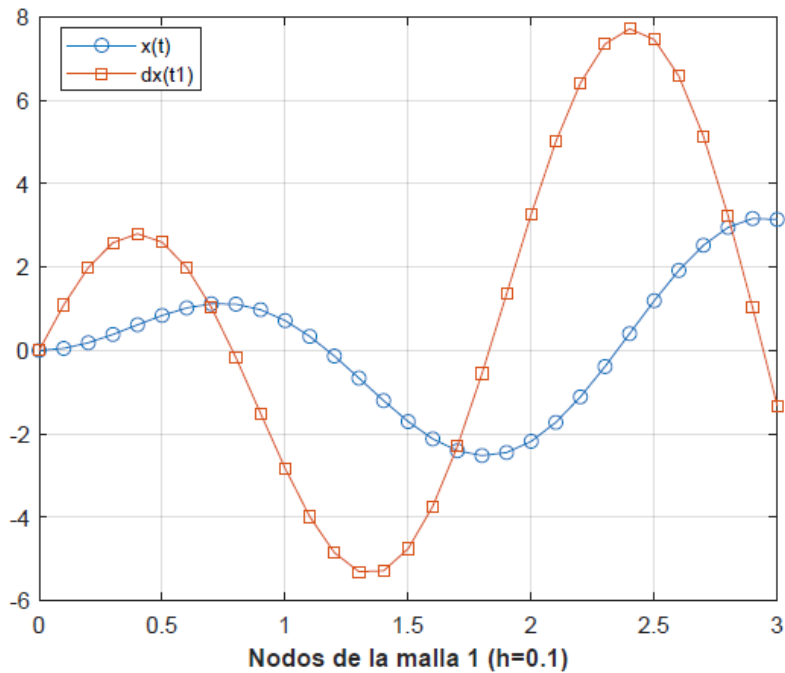


Figura 7.2. Gráfica de la solución analítica $x(t)$ del PVI (7.6) y primera derivada $\dot{x}(t)$. Interpolación lineal por defecto de *MATLAB*.

Asimismo, los valores numéricos correspondientes a ambas funciones pueden consultarse en la **Tabla 7.5**, obtenidos con aritmética de 4 cifras significativas.

<i>Nodos t</i>	$x(t)$	$\dot{x}(t)$	<i>Nodos t</i>	$x(t)$	$\dot{x}(t)$
0.00	0.000	0.000	1.60	-2.123	-3.741
0.10	0.049	1.092	1.70	-2.407	-2.303
0.20	0.185	1.991	1.80	-2.524	-0.555
0.30	0.384	2.587	1.90	-2.451	1.358
0.40	0.616	2.801	2.00	-2.182	3.270
0.50	0.841	2.599	2.10	-1.727	5.011
0.60	1.020	1.992	2.20	-1.115	6.418
0.70	1.120	1.036	2.30	-0.386	7.355
0.80	1.111	-0.173	2.40	0.404	7.721
0.90	0.978	-2.831	2.50	1.194	7.464
1.00	0.717	-1.885	2.60	1.922	6.586
1.10	0.338	-3.991	2.70	2.525	5.143
1.20	-0.135	-4.855	2.80	2.953	3.239
1.30	-0.663	-5.312	2.90	3.167	1.024
1.40	-1.204	-5.290	3.00	3.144	-1.324
1.50	-1.707	-4.760			

Tabla 7.5. Valores numéricos de la solución analítica $x(t)$ de (7.6) y su primera derivada $\dot{x}(t)$ con paso uniforme $h = 0.1$.

Los resultados son coherentes, pues las condiciones iniciales nulas prefijadas se cumplen, así como la tendencia de la derivada. Esto es importante, ya que en la descripción de un determinado fenómeno vibratorio, la correspondencia entre la variable vibratoria y la velocidad de vibración ha de ser certera.

7.2.1.2. Estudio del $|EGT|$ para cada método numérico propuesto.

Tras haber indicado la forma y los valores que toma la solución analítica del PVI (7.6), se puede proceder a estudiar el módulo del Error Global de Truncamiento (EGT (3.6)) respecto de dicha solución que cada método indicado en la *Tabla 7.6* (con su alias definido en el *Capítulo 3* y programados en el *ANEXO I*) produce, cuando todos ellos son aplicados al intervalo propuesto discretizado con la primera malla de la *Tabla 7.5* ($h = 0.1$). La razón de ser de este estudio tiene su origen en verificar intuitivamente, en base al orden de convergencia que posteriormente se demostrará empíricamente, el comportamiento de cada método de obtención de soluciones discretas aproximadas empleado.

Métodos <i>Runge-Kutta</i>	Métodos <i>Runge-Kutta-Nyström</i>
<i>RK Euler</i> explícito (" <i>ERKs1o</i> ")	<i>RKN</i> directo de orden 3 (" <i>RKN3</i> ")
<i>RK Heun</i> de orden 3 (" <i>HRK3s1o</i> ")	<i>RKN</i> transformado de orden 3 (" <i>RKN3t</i> ")
<i>RK</i> de orden 4 (" <i>RK4s1o</i> ")	<i>RKN</i> de orden 4 (" <i>RKN4</i> ")

Tabla 7.6. Métodos numéricos para la obtención de soluciones discretas aproximadas del PVI (7.6) empleados con su alias de *MATLAB*.

Cabe destacar que no se ensayan los métodos *Runge-Kutta-Nyström* correspondientes a problemas independientes de la primera derivada ya que no se está ante un problema de dichas características. Al mismo tiempo, tampoco se aplica el método *Runge-Kutta* de *Gauss* y orden 6 con 3 etapas, quedando reservado para el experimento numérico 2 (sección 7.2 de este capítulo).

7.2.1.2.1. $|EGT|$ para el método de Euler explícito.

El $|EGT|$ correspondiente al método de *Euler* explícito (sección 3.2.3 del *Capítulo 3*) se indica en la *Figura 7.3*, junto con sus valores numéricos en los nodos dados por la *Tabla 7.7* (notación científica y tres cifras significativas).

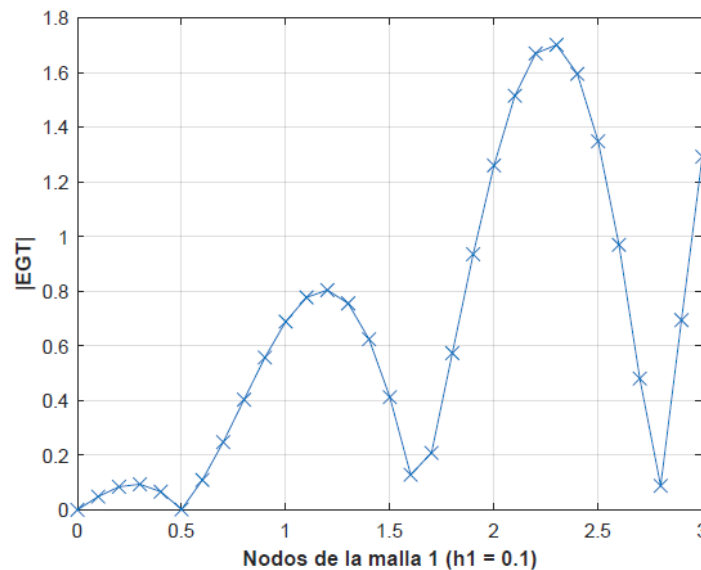


Figura 7.3. Gráfica del $|EGT|$ correspondiente al método de *Euler* explícito aplicado al PVI (7.6) en el intervalo considerado con paso $h = 0.1$.

<i>Nodos t</i>	$ x(t) - x_t $	<i>Nodos t</i>	$ x(t) - x_t $
0.00	0.00E+00	1.60	1.28E-01
0.10	4.86E-02	1.70	2.09E-01
0.20	8.45E-02	1.80	5.73E-01
0.30	9.39E-02	1.90	9.35E-01
0.40	6.66E-02	2.00	1.26E+00
0.50	1.77E-03	2.10	1.52E+00
0.60	1.09E-01	2.20	1.67E+00
0.70	2.48E-01	2.30	1.70E+00
0.80	4.03E-01	2.40	1.60E+00
0.90	5.57E-01	2.50	1.35E+00
1.00	6.89E-01	2.60	9.70E-01
1.10	7.77E-01	2.70	4.80E-01
1.20	8.04E-01	2.80	8.80E-02
1.30	7.56E-01	2.90	6.94E-01
1.40	6.25E-01	3.00	1.29E+00
1.50	4.12E-01		

Tabla 7.7. Valores numéricos del $|EGT|$ en los nodos de la malla 1 ($h = 0.1$) correspondiente al método de *Euler* explícito aplicado al *PVI* (7.6).

Se indica el valor máximo de dicho error, el cual se alcanza en el nodo $t = 2.3$ s y toma un valor desproporcionado respecto a los órdenes de magnitud de la solución analítica (Tabla 7.5). El método de *Euler* suele presentar este tipo de problemas, pues su precisión es muy limitada al emplear con mallas no muy densas aproximaciones basadas en líneas poligonales (tal y como se indica en la Figura 3.1 del Capítulo 3).

7.2.1.2.2. $|EGT|$ de métodos numéricos de orden 3.

Procediendo ahora con el módulo del *EGT* correspondiente a todos los métodos exclusivamente de orden 3 presentes en la Tabla 7.6, se obtiene la gráfica de la Figura 7.4. Los valores numéricos del error se adjuntan en la Tabla 7.8, con los máximos encuadrados.

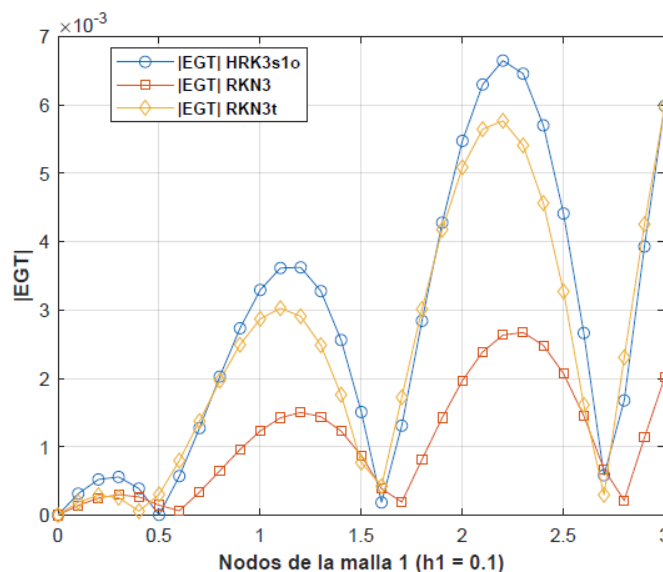


Figura 7.4. Gráfica del $|EGT|$ correspondiente a los métodos exclusivamente de orden 3 aplicados al *PVI* (7.6) en el intervalo considerado con paso uniforme $h = 0.1$.

<i>Nodos t</i>	<i> EGT </i> "HRK3"	<i> EGT </i> "RKN3"	<i> EGT </i> "RKN3t"	<i>Nodos t</i>	<i> EGT </i> "HRK3"	<i> EGT </i> "RKN3"	<i> EGT </i> "RKN3t"
0.00	0.00E+00	0.00E+00	0.00E+00	1.60	1.85E-04	3.86E-04	4.24E-04
0.10	3.13E-04	1.40E-04	1.89E-04	1.70	1.31E-03	1.91E-04	1.72E-03
0.20	5.21E-04	2.50E-04	2.90E-04	1.80	2.84E-03	8.13E-04	3.01E-03
0.30	5.58E-04	2.98E-04	2.55E-04	1.90	4.28E-03	1.43E-03	4.18E-03
0.40	3.87E-04	2.66E-04	5.71E-05	2.00	5.48E-03	1.97E-03	5.09E-03
0.50	2.47E-06	1.44E-04	3.03E-04	2.10	6.30E-03	2.39E-03	5.65E-03
0.60	5.70E-04	6.23E-05	7.97E-04	2.20	6.65E-03	2.63E-03	5.77E-03
0.70	1.27E-03	3.36E-04	1.38E-03	2.30	6.46E-03	2.67E-03	5.41E-03
0.80	2.03E-03	6.47E-04	1.97E-03	2.40	5.70E-03	2.48E-03	4.57E-03
0.90	2.73E-03	9.59E-04	2.49E-03	2.50	4.41E-03	2.07E-03	3.27E-03
1.00	3.29E-03	1.23E-03	2.87E-03	2.60	2.66E-03	1.45E-03	1.61E-03
1.10	3.62E-03	1.42E-03	3.03E-03	2.70	5.82E-04	6.72E-04	2.96E-04
1.20	3.62E-03	1.50E-03	2.91E-03	2.80	1.68E-03	2.14E-04	2.31E-03
1.30	3.28E-03	1.44E-03	2.49E-03	2.90	3.93E-03	1.13E-03	4.26E-03
1.40	2.56E-03	1.23E-03	1.76E-03	3.00	5.99E-03	2.02E-03	5.98E-03
1.50	1.51E-03	8.71E-04	7.68E-04				

Tabla 7.8. Valores numéricos del $|EGT|$ en los nodos de la malla 1 ($h = 0.1$) correspondientes a los métodos "HRK3s1o", "RKN3" y "RKN3t" aplicados al PVI (7.6).

Las aproximaciones a la solución exacta de la **Figura 7.2** y la **Tabla 7.5** mejoran sustancialmente respecto a la obtenida mediante el método de *Euler*, ya que los errores máximos, alcanzados en el entorno del mismo nodo, disminuyen hasta situarse en una cota del orden de 10^{-3} . Como se verá posteriormente, esto tiene su explicación directa en el orden de convergencia que presentan los métodos numéricos ensayados. Cabe destacar además que los errores se propagan siguiendo la solución de la **Figura 7.2**, es decir, de forma periódica.

7.2.1.2.3. $|EGT|$ de métodos numéricos de orden 3.

Finalmente, graficando los errores correspondientes a los métodos restantes de orden 4, se obtiene la **Figura 7.5**, con sus correspondiente tabla de valores numéricos y máximos (**Tabla 7.9**):

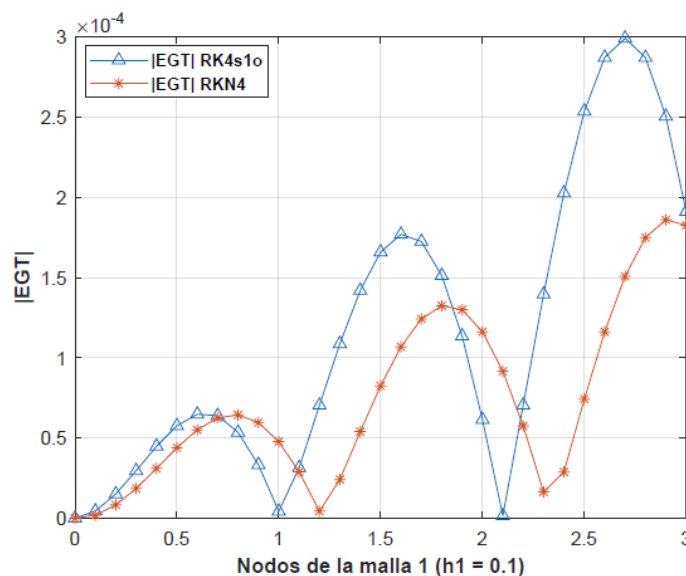


Figura 7.5. Gráfica del $|EGT|$ correspondiente a los métodos exclusivamente de orden 4 aplicados al PVI (7.6) en el intervalo considerado con paso uniforme $h = 0.1$.



<i>Nodos t</i>	$ EGT $ "RK4"	$ EGT $ "RKN4"	<i>Nodos t</i>	$ EGT $ "RK4"	$ EGT $ "RKN4"
0.00	0.00E+00	0.00E+00	1.60	1.77E-04	1.07E-04
0.10	4.42E-06	1.81E-06	1.70	1.72E-04	1.24E-04
0.20	1.51E-05	8.27E-06	1.80	1.51E-04	1.32E-04
0.30	2.96E-05	1.85E-05	1.90	1.14E-04	1.30E-04
0.40	4.48E-05	3.10E-05	2.00	6.14E-05	1.16E-04
0.50	5.76E-05	4.39E-05	2.10	1.51E-06	9.16E-05
0.60	6.47E-05	5.51E-05	2.20	7.04E-05	5.75E-05
0.70	6.39E-05	6.25E-05	2.30	1.40E-04	1.63E-05
0.80	5.35E-05	6.44E-05	2.40	2.03E-04	2.88E-05
0.90	3.33E-05	5.96E-05	2.50	2.54E-04	7.43E-05
1.00	4.25E-06	4.77E-05	2.60	2.87E-04	1.16E-04
1.10	3.14E-05	2.89E-05	2.70	2.99E-04	1.51E-04
1.20	7.04E-05	4.36E-06	2.80	2.87E-04	1.75E-04
1.30	1.09E-04	2.41E-05	2.90	2.51E-04	1.86E-04
1.40	1.42E-04	5.40E-05	3.00	1.91E-04	1.83E-04
1.50	1.66E-04	8.25E-05			

Tabla 7.9. Valores numéricos del $|EGT|$ en los nodos de la malla 1 ($h = 0.1$) correspondientes a los métodos "RK4s1o" y "RKN4" aplicados al PVI (7.6).

El comportamiento de estos métodos numéricos resulta ser mejor que el de todos los anteriormente ensayados, pues los errores están acotados en un máximo del orden de 10^{-4} , es decir, 10 veces menos que en el caso de los métodos de tercer orden y hasta 10^4 veces menos que en el caso del método de Euler, el cual ya se ha indicado que es de primer orden. Además, los errores máximos se desplazan al entorno del nodo extremo del intervalo ($t = 3 s$), algo coherente teniendo en cuenta que los métodos son de un paso, acumulando los errores en la aproximación del paso anterior para la determinación de la aproximación discreta correspondiente al siguiente nodo del siguiente paso.

7.2.1.3. Determinación del orden empírico de los métodos numéricos. Comparación con el orden teórico.

Si el lector ha prestado atención a las conclusiones que se aportan en el apartado anterior, apoyadas todas ellas en los resultados gráficos y numéricos del $|EGT|$ correspondiente a cada método ensayado, no le resultará complicado predecir los resultados que se aportan a continuación.

Una práctica común a la hora de determinar empíricamente el orden de convergencia de un método numérico de obtención de soluciones discretas aproximadas (caso de todos los métodos propuestos en la **Tabla 7.6**) consiste en graficar el error en el último nodo del intervalo empleando mallas progresivamente densificadas (las definidas en la **Tabla 7.4**) para su posterior regresión lineal. Esto permite, si los valores en las ordenadas y abscisas se dan como logaritmos en la base del densificado de malla efectuado manteniendo las escalas lineales, obtener una línea de tendencia cuya pendiente ha de ser el orden empírico.

Para poder realizar el estudio con cada uno de los métodos numéricos propuestos en la **Tabla 7.6**, se necesitan computar los $|EGT|$ de cada método para cada malla de la **Tabla 7.4** en el nodo extremo del PVI (7.6) discretizado

con (2.11), el cual es $t = 3$ s. Procediendo en **MATLAB**, se obtiene la **Tabla 7.10**:

ID malla	ERKs1o	HRK3s1o	RK4s1o	RKN3	RKN3t	RKN4
1 ($h = 0.1$)	1.29E+00	5.99E-03	1.91E-04	2.02E-03	5.98E-03	1.83E-04
2 ($h = 0.05$)	6.96E-01	8.18E-04	1.01E-05	2.70E-04	8.04E-04	1.12E-05
3 ($h = 0.025$)	3.49E-01	1.06E-04	5.69E-07	3.48E-05	1.04E-04	6.92E-07
4 ($h = 0.0125$)	1.74E-01	1.35E-05	3.36E-08	4.41E-06	1.32E-05	4.30E-08
5 ($h = 0.00625$)	8.67E-02	1.70E-06	2.04E-09	5.55E-07	1.66E-06	2.68E-09

Tabla 7.10. Valores numéricos del **|EGT|** para cada malla de ensayo y correspondientes a cada método ensayado en el nodo extremo $t = 3$ (s) del **PVI (7.6)**.

Al analizar la tabla, se llega a la conclusión evidente de que **disminuir el paso provoca una disminución del |EGT| en el nodo extremo**. Para poder graficar y realizar las regresiones pertinentes, se toman los datos en la forma de (7.9):

$$|EGT| \rightarrow |\log_2 |EGT||, \quad h \rightarrow |\log_2(h)|$$

Ecuación (7.9)

En la **Tabla 7.11** se obtiene el orden empírico del método de *Euler* mediante las expresiones logarítmicas de (7.9) (partiendo de la definición de la sección 3.2.2 correspondiente al **Capítulo 3**). Además, puede observarse la **convergencia hacia el orden teórico conforme la malla se densifica y el método proporciona una aproximación mejor**.

ID malla	EGT "ERK1s1o"	Orden empírico: $p = \left \log_2 \left(\frac{ EGT _{ID-1}}{ EGT _{ID}} \right) \right $
1 ($h = 0.1$)	1.29E+00	
2 ($h = 0.05$)	6.96E-01	0.893
3 ($h = 0.025$)	3.49E-01	0.994
4 ($h = 0.0125$)	1.74E-01	1.005
5 ($h = 0.00625$)	8.67E-02	1.004
PROMEDIO		$\bar{p} = 0.974$

Tabla 7.11. Determinación del orden empírico del método de *Euler* explícito. **|EGT|** en nodo extremo $t = 3$ (s).

El mismo procedimiento puede aplicarse a todos los métodos ensayados de orden 3. Procediendo, se tiene la **Tabla 7.12**:

ID malla	EGT "HRK3s1o"	Orden empírico p:	EGT "RKN3"	Orden empírico p:	EGT "RKN3t"	Orden empírico p:
1 ($h = 0.1$)	5.99E-03	-	2.02E-03	-	5.98E-03	-
2 ($h = 0.05$)	8.18E-04	2.872	2.70E-04	2.901	8.04E-04	2.894
3 ($h = 0.025$)	1.06E-04	2.946	3.48E-05	2.956	1.04E-04	2.955

4 ($h = 0.0125$)	1.35E-05	2.974	4.41E-06	2.979	1.32E-05	2.979
5 ($h = 0.00625$)	1.70E-06	2.988	5.55E-07	2.989	1.66E-06	2.990
PROMEDIO	$\bar{p} = 2.945$		$\bar{p} = 2.956$		$\bar{p} = 2.954$	

Tabla 7.12. Determinación del orden empírico de los métodos ensayados exclusivamente de orden 3 aplicados al **PVI (7.6)**. **|EGT|** en nodo extremo $t = 3$ (s).

Finalmente, el orden empírico de los métodos de orden 4 se obtiene en la **Tabla 7.13**:

ID malla	EGT "RK4s1o"	Orden empírico p:	EGT "RKN4"	Orden empírico p:
1 ($h = 0.1$)	1.91E-04	-	1.83E-04	-
2 ($h = 0.05$)	1.01E-05	4.246	1.12E-05	4.027
3 ($h = 0.025$)	5.69E-07	4.146	6.92E-07	4.016
4 ($h = 0.0125$)	3.36E-08	4.080	4.30E-08	4.008
5 ($h = 0.00625$)	2.04E-09	4.041	2.68E-09	4.004
PROMEDIO	$\bar{p} = 4.128$		$\bar{p} = 4.014$	

Tabla 7.13. Determinación del orden empírico de los métodos ensayados exclusivamente de orden 4 aplicados al **PVI (7.6)**. **|EGT|** en nodo extremo $t = 3$ (s).

Se pueden graficar los resultados de las tablas anteriores y observar las tendencias con pendiente coincidente al orden empírico calculado. Se tienen, para cada uno de los métodos ensayados, las tendencias dadas por la **Figura 7.6**:

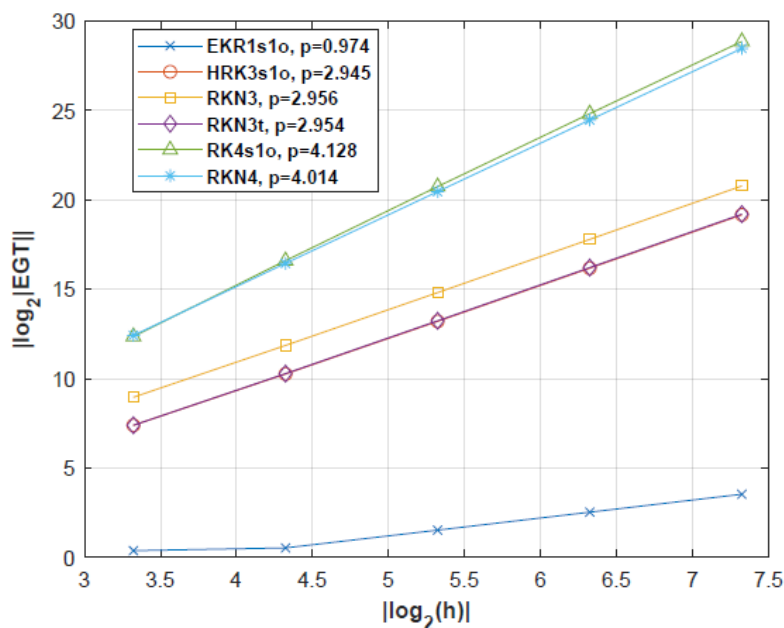


Figura 7.6. Determinación del orden empírico de los métodos numéricos ensayados aplicados al **PVI (7.6)**.

Para observar con mayor nitidez las tendencias correspondientes a los métodos “*HRK3s1o*” y “*RKN3t*”, considérese el zoom ilustrado en la **Figura 7.7**:

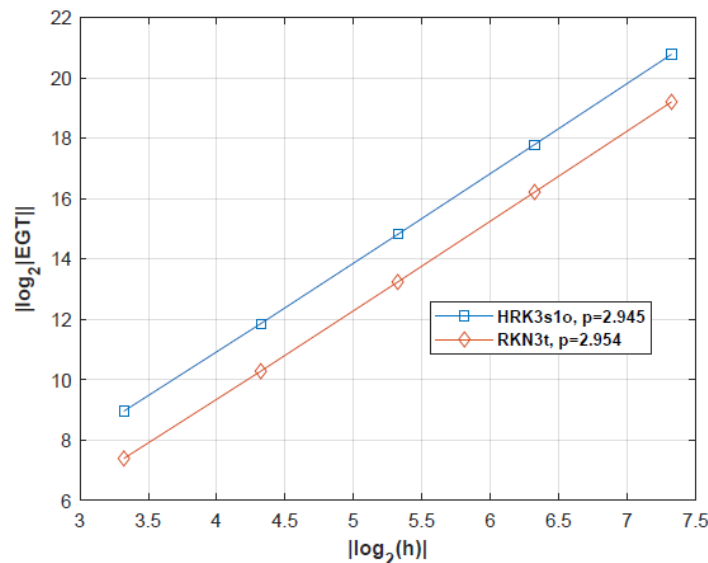


Figura 7.7. Determinación del orden empírico de los métodos “*HRK3s1o*” y “*RKN3t*” aplicados al **PVI (7.6)**.

Finalmente, en la **Tabla 7.14** se indica el error relativo cometido respecto del orden teórico de todos los métodos ensayados.

<i>Método ensayado</i>	<i>Orden teórico</i>	<i>Orden empírico</i>	<i>Error cometido (%)</i>
“ <i>ERKs1o</i> ”	1	0.974	2.6 %
“ <i>HRK3s1o</i> ”	3	2.945	1.83 %
“ <i>RKN3</i> ”	3	2.956	1.47 %
“ <i>RKN3t</i> ”	3	2.954	1.53 %
“ <i>RK4s1o</i> ”	4	4.128	3.2 %
“ <i>RKN4</i> ”	4	4.014	0.35 %

Tabla 7.14. Errores de los órdenes empíricos respecto de los teóricos correspondientes a los métodos numéricos ensayados y aplicados al **PVI (7.6)**.

7.2.2. Conclusiones del experimento.

Las conclusiones de este experimento numérico, teniendo en cuenta todos los resultados y estrategias para conseguirlas abordadas previamente, son las indicadas a continuación:

- i. El módulo del **EGT** de cada método numérico ensayado, empleando para su determinación la solución exacta (7.7) que en este problema sí esta disponible, no es necesariamente creciente conforme avanzan los pasos del método numérico. **Se determina que su comportamiento, para este caso particular, resulta ser periódico.**
- ii. Es posible computar los órdenes empíricos de los métodos numéricos dispuestos en este experimento aplicando la definición de **EGT** hecha en la ecuación (3.6), correspondiente a la sección 3.2.2 del **Capítulo 3**, y aplicada mediante una disminución sucesiva del paso (**Tabla 7.4**).



- iii. Se obtiene una **convergencia progresiva del orden empírico hacia el valor teórico conforme se disminuye el paso**, deduciendo que **las mallas de ensayo empleadas son lo suficientemente sensibles para los métodos numéricos aplicados**.

Finalmente, se recuerda que el *script* de **MATLAB "PVI_VibMec_Lineal_1gdl"**, empleado para la realización de este experimento numérico, puede ser consultado en el **ANEXO IV** de esta memoria.

7.3. Experimento numérico 2. Problemas de prueba. Órdenes de convergencia empíricos.

El experimento numérico 1 (7.1) resulta ser de gran interés a la hora de evaluar el comportamiento del **|EGT|** correspondiente a los métodos numéricos aplicados que obtienen soluciones discretas aproximadas en la malla (2.11) del **PVI (7.6)**, planteado éste como problema idóneo para la realización del estudio citado. Además, también es posible verificar el orden empírico de los métodos ensayados, **faceta muy importante para un ingeniero cuando éste pretende resolver un determinado problema aproximando su solución y con el mínimo error posible (o al menos, siendo consciente de su comportamiento)**.

No obstante, no es posible realizar el estudio para todos los métodos del **Capítulo 3**, ya que el problema no es independiente de la primera derivada y' , por lo que aquellos métodos *Runge-Kutta-Nyström* explícitos expuestos en 3.3.5 no son objeto de análisis. Además, el método *Runge-Kutta* de Gauss (3 etapas y orden 6, sección 3.2.6) tampoco se ha ensayado.

Con todas las consideraciones anteriores, **el experimento numérico desarrollado a continuación concierne a la aplicación de la misma metodología que en el anterior pero empleando dos PVIs de 2º orden especialmente diseñados para poder ensayar los métodos numéricos restantes citados, ambos con solución analítica para facilitar el tratamiento del EGT y su módulo**.

Considérese el **PVI** lineal completo dado por (7.10), el cual es del tipo (2.7).

$$1 \rightarrow \begin{cases} y''(x) = xy'(x) - 5y(x) + x^4 + 22x^3 + 15x^2 + 10x + 7 \\ y(0) = 1, \quad y'(0) = 1 \\ x \in [0,2] \end{cases}$$

Ecuación (7.10)

A su vez, considérese su solución analítica dada por el siguiente polinomio de grado 5 (7.11):

$$y(x) = x^5 + x^4 + x^3 + x^2 + x + 1$$

Ecuación (7.11)

Ahora, puede considerarse también el **PVI** lineal independiente de la primera derivada y' y del tipo (2.8) indicado en (7.12):

$$2 \rightarrow \begin{cases} y''(x) = y(x) - x^4 - x^3 + 11x^2 + 5x + 1 \\ y(0) = 1, \quad y'(0) = 1 \\ x \in [0,2] \end{cases}$$

Ecuación (7.12)

Su solución analítica es el siguiente polinomio de grado 4 (7.13):

$$y(x) = x^4 + x^3 + x^2 + x + 1$$

Ecuación (7.13)

El **PVI** (7.10) es útil para ensayar el comportamiento del método *Runge-Kutta de Gauss*, mientras que el (7.12) lo es para los métodos *Runge-Kutta-Nyström* especialmente diseñados para problemas independientes de y' . Resumiendo, todos los métodos que son objeto de ensayo en este experimento se indican en la **Tabla 7.15**, con su alias en **MATLAB** expuesto también en las **Tablas 3.2** y **3.6** (códigos disponibles en el **ANEXO I**).

<i>Método</i>	<i>Alias en MATLAB</i>
RK de Gauss	"GRK6s1o"
RKN simple de orden 4 (no y')	"RKN4_no_dy"
RKN simple de orden 5 (no y')	"RKN51_no_dy"
RKN especial de orden 5 (no y')	"RKN52_no_dy"

Tabla 7.15. Métodos numéricos para la obtención de soluciones discretas aproximadas de los **PVIs** (7.10) y (7.12) empleados con su alias de **MATLAB**.

Considérense ahora las mallas de nodos equiespaciadas según (2.13) indicadas en la **Tabla 7.16**, con las cuales se aplica la misma metodología para obtener el orden empírico que en el experimento numérico 1.

<i>Identificador de malla (ID)</i>	<i>Paso uniforme (h)</i>	<i>Nº de subintervalos (n)</i>
1	0.4	5
2	0.2	10
3	0.1	20
4	0.05	40
5	0.025	80

Tabla 7.16. Mallas equiespaciadas progresivamente densificadas en el intervalo [0,2].

El densificado progresivo realizado también corresponde (al igual que las mallas de la **Tabla 7.4**) a una disminución en potencias de base 2. La regla aplicada es la dada por (7.8).

Antes de comenzar con los estudios relativos al orden empírico de cada uno de los métodos propuestos en la **Tabla 7.15**, conviene observar cuál es la forma que toman las soluciones de (7.10) y (7.12). Graficando (7.11) y (7.13) en los

nodos de la malla x_3 ($h = 0.1$) propuesta en la **Tabla 7.16** y empleando interpolación lineal por defecto de **MATLAB**, se tiene la **Figura 7.8**:

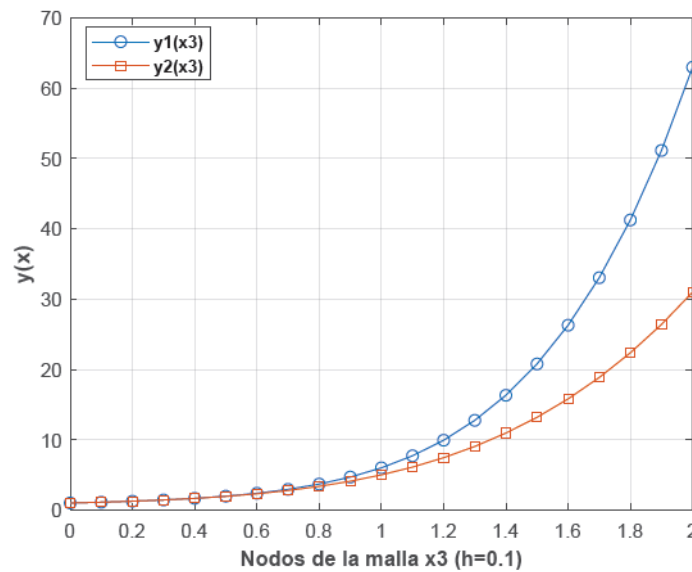


Figura 7.8. Soluciones analíticas de los **PVI(7.10)** y **(7.12)** en los nodos de la malla x_3 .

Se observa claramente cómo **(7.11)** incrementa más sus valores que **(7.13)**, al ser un polinomio quíntico frente a un polinomio de cuarto grado evaluado en los mismos nodos del mismo intervalo $[0,2]$. Los valores numéricos de cada una de las soluciones vienen dados por la **Tabla 7.17**:

Nodos x	<i>PVI(7.10): $y_1(x_3)$</i>	<i>PVI(7.12): $y_2(x_3)$</i>
0.00	1.000	1.000
0.10	1.111	1.111
0.20	1.250	1.250
0.30	1.428	1.425
0.40	1.660	1.650
0.50	1.969	1.938
0.60	2.383	2.306
0.70	2.941	2.773
0.80	3.689	3.362
0.90	4.686	4.095
1.00	6.000	5.000
1.10	7.716	6.105
1.20	9.930	7.442
1.30	12.756	9.043
1.40	16.324	10.946
1.50	20.781	13.188
1.60	26.295	15.810
1.70	33.054	18.855
1.80	41.265	22.370
1.90	51.162	26.401
2.00	63.000	31.000

Tabla 7.17. Valores numéricos de las soluciones de los **PVI(7.10)** y **(7.12)** en los nodos de la malla x_3 .

En la tabla anterior se ha utilizado aritmética de 3 cifras decimales significativas. Expuestas las soluciones de los **PVI(7.10)** y **(7.12)** planteados, ya se puede

comenzar el estudio del módulo del EGT y el orden empírico para cada uno de los métodos numéricos de la *Tabla 7.15*.

7.3.1. $|EGT|$ y orden empírico para el método *Runge-Kutta de Gauss*.

Para poder plantear la determinación empírica del orden correspondiente al método *Runge-Kutta de Gauss* (3 etapas y orden teórico 6), ha de considerarse el $|EGT|$ de este método para cada una de las mallas de la *Tabla 7.16*, con el objeto de **enjuiciar gráficamente el comportamiento** del mismo y cuando es aplicado a la resolución del *PVI (7.10)*. Procediendo con las mallas x_1 y x_2 , se tienen las *Figuras 7.9* y *7.10*, respectivamente.

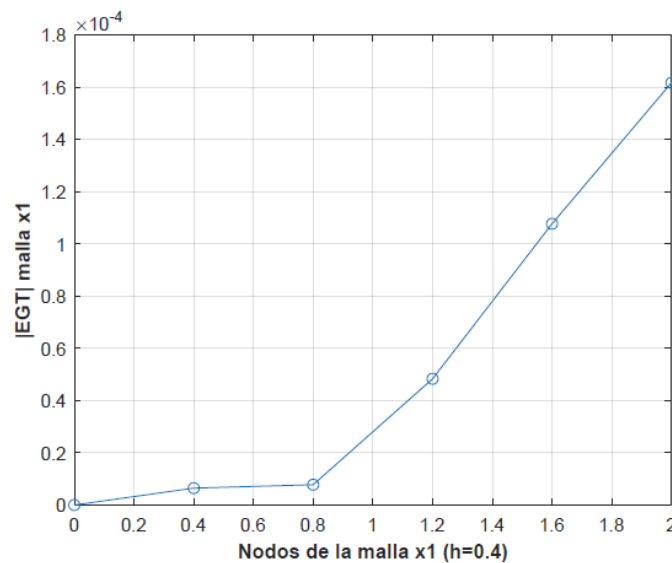


Figura 7.9. $|EGT|$ correspondiente al método “GRK6s1o” para la malla x_1 tras resolver el *PVI (7.10)*.

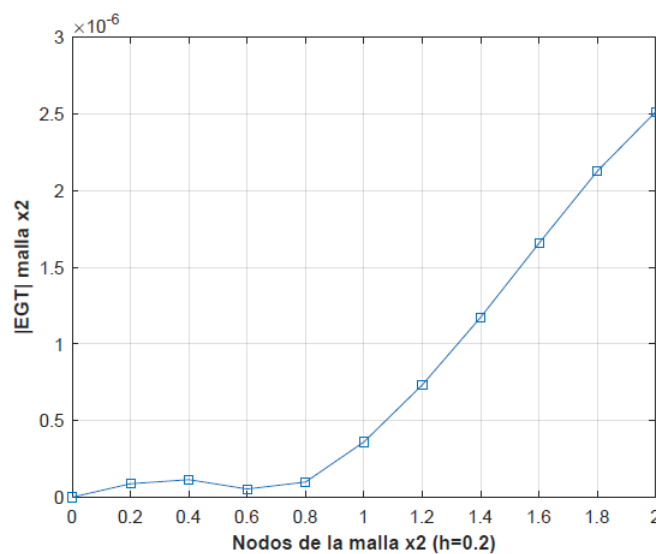


Figura 7.10. $|EGT|$ correspondiente al método “GRK6s1o” para la malla x_2 tras resolver el *PVI (7.10)*.

Los resultados son los esperados, pues al densificar la malla el $|EGT|$ decrece (y su comportamiento se mantiene y concuerda con el de la solución analítica de la *Figura 7.8*). Sin todavía poder decir nada acerca del orden empírico, pero anticipando que se han graficado ambos errores de forma independiente a los demás (con las otras mallas de la *Tabla 7.16*) por ser de órdenes de magnitud dispares (mallas no demasiado densas), se grafica en la *Figura 7.11* el $|EGT|$ correspondiente al resto de mallas.

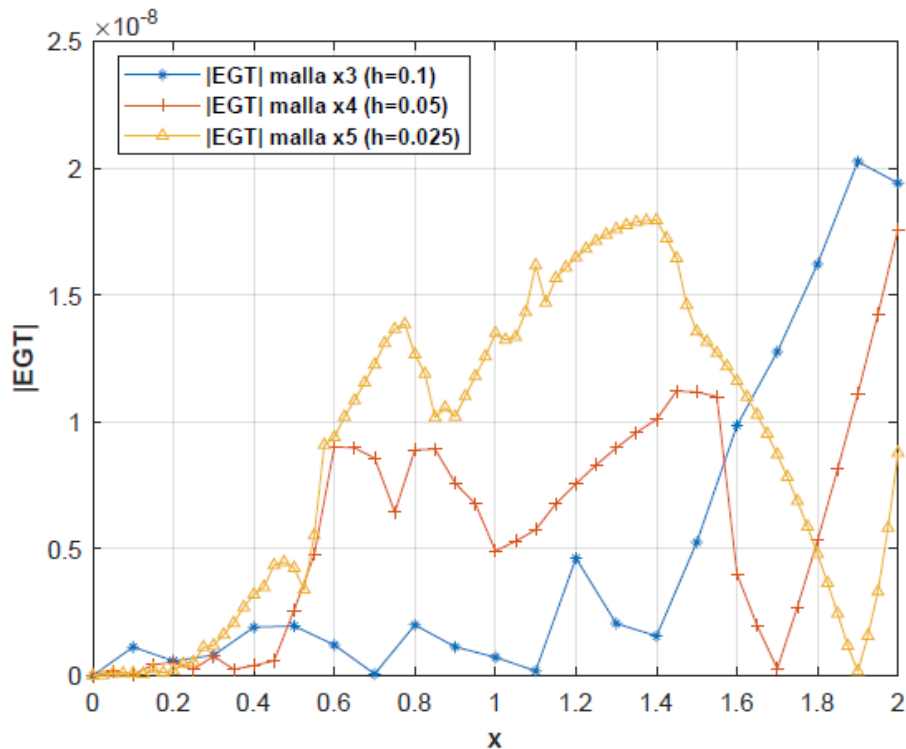


Figura 7.11. Gráfica del $|EGT|$ para el método “GRK6s1o” aplicado a la resolución del PVI (7.10) empleando diversas mallas de ensayo.

Ahora, se tiene un **comportamiento mucho más irregular del error independientemente de la malla empleada**. El significado que tiene la *Figura 7.11* es evidente, pues el método empleado, al ser de un orden teórico elevado (6), provoca que la disminución progresiva del paso efectuada en la *Tabla 7.16* sea insensible al comportamiento una vez sobrepasado un determinado umbral. Ese umbral, a la vista de las *Figuras 7.9 y 7.10*, corresponde a la malla x_2 ($h = 0.2$), pues sí se observa una disminución en el orden de magnitud del error con respecto a la malla x_1 ($h = 0.4$).

Para evaluar numéricamente la deducción anterior y poder obtener la recta cuya pendiente es el orden empírico del método numérico ensayado (empleando las mallas exclusivamente sensibles), considérese la *Tabla 7.18*, en la cual se ha obtenido el valor máximo de EGT para cada malla (utilizando la función “*id_elemento_vector*” de *MATLAB*, ANEXO V), así como el orden empírico correspondiente, mediante el mismo razonamiento expuesto en la sección 7.2.1.3 del experimento numérico 1.

ID malla	$ELT_{m\acute{a}x}$	Orden empírico: $p = \left \log_2 \left(\frac{EGT_{ID-1}}{EGT_{ID}} \right) \right $
1 ($h = 0.4$)	1.62E-04	
2 ($h = 0.2$)	2.51E-06	6.001
3 ($h = 0.1$)	2.03E-08	6.954
4 ($h = 0.05$)	1.76E-08	0.204
5 $h = (0.025)$	1.80E-08	-0.003

Tabla 7.18. Determinación del orden empírico del método “GRK6s1o”. $|EGT|$ máximo para cada malla de ensayo.

La conclusión es clara, y verifica las deducciones hechas sobre la gráfica de la **Figura 7.11**. El método, al ser de alto orden, es insensible a la disminución del paso a partir de la malla x_2 . No obstante, sí se consigue un orden empírico bastante coherente realizando el promedio de los dos primeros valores, es decir, $\bar{p} = 6.482$. El error relativo cometido respecto del orden teórico (6) es del 8.03 %.

Finalmente, trazando la recta correspondiente a las mallas x_1 y x_2 con las magnitudes logarítmicas de (7.9), se consigue en la **Figura 7.12** la evidencia gráfica que pretendía obtenerse al comienzo de esta sección.

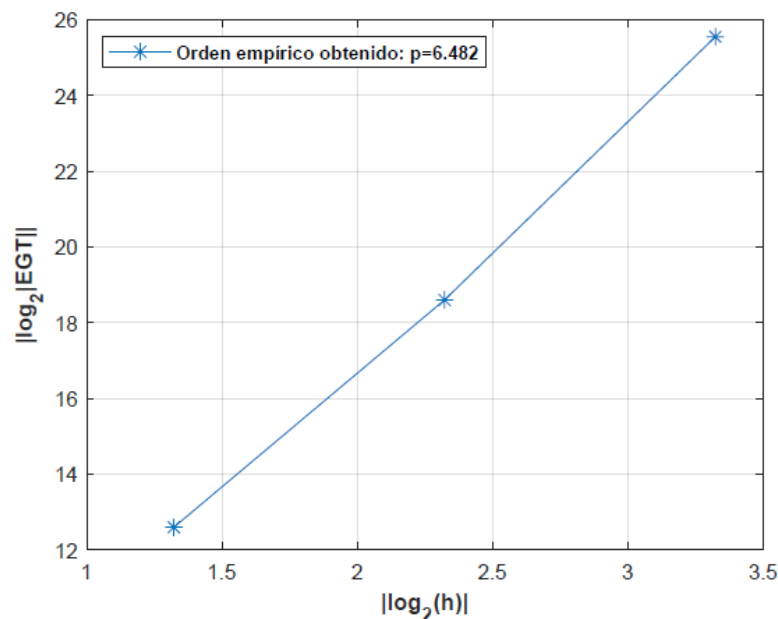


Figura 7.12. Determinación del orden empírico del método “GRK6s1o” aplicado a la resolución del PVI (7.10).

7.3.2. $|EGT|$ y orden empírico para los métodos Runge-Kutta-Nyström independientes de y' .

El mismo procedimiento se aplica en esta sección para los métodos RKN de la **Tabla 7.15**, pero empleando ahora el PVI (7.12), ya que es un problema independiente de la primera derivada y' . El $|EGT|$ del método “RKN4_no_dy”,

cuando se emplea la malla x_3 para aproximar soluciones discretas de (7.12) es el dado por la **Figura 7.13**:

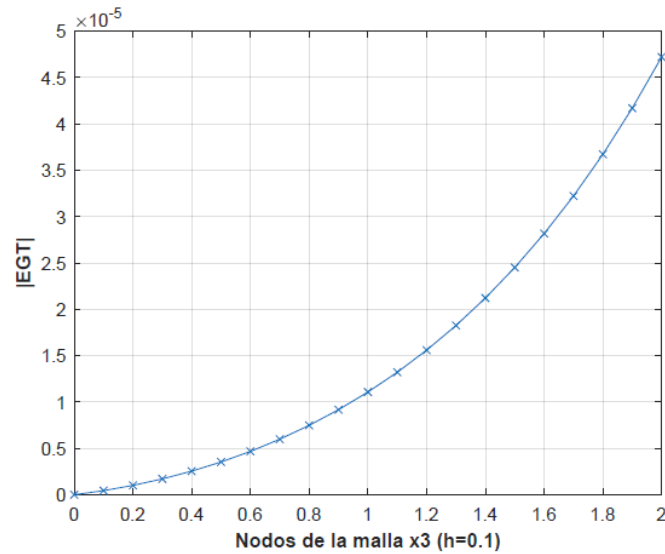


Figura 7.13. Gráfica del $|EGT|$ correspondiente al método “RKN4_no_dy” aplicado al PVI (7.12) en la malla x_3 .

Para esta malla en concreto, el $|EGT|$ del método empleado se comporta prácticamente igual que la solución analítica, lo cual indica sensibilidad al densificado progresivo (equivalentemente disminución del tamaño de paso h) al menos hasta la malla x_3 . Procediendo ahora con la gráfica del error de los métodos *RKN* de orden 5 propuestos para su ensayo, se tiene la **Figura 7.14**:

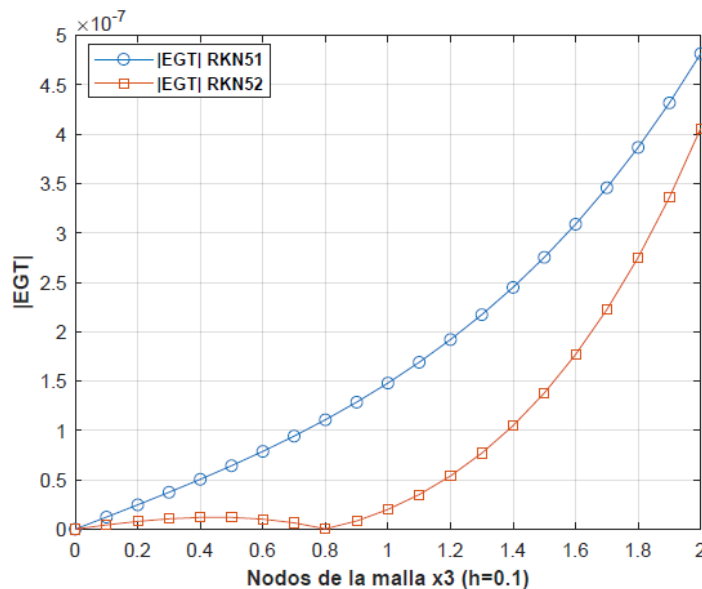


Figura 7.14. Gráfica del $|EGT|$ correspondiente a los métodos *RKN* de orden 5 para problemas independientes de y' aplicados al PVI (7.12) empleando la malla x_3 .

El $|EGT|$ decrece respecto al del método “RKN4_no_dy” de la **Figura 7.13**. Esto hace indicar que, efectivamente, el orden de convergencia para los métodos

de orden teórico 5 es mayor. Para obtener empíricamente dicho orden y dilucidar si el densificado de malla progresivo de la **Tabla 7.16** es sensible a todos los métodos, se tiene la **Tabla 7.19**, la cual considera ahora el $|EGT|$ en el último nodo de la discretización del **PVI (7.12)**, es decir, en $x = 2$, ya que el comportamiento del error tiende a ser de la misma naturaleza que la solución analítica (monótonamente creciente).

ID malla	$ EGT $ "RKN4"	Orden empíricop:	$ EGT $ "RKN51"	Orden empíricop:	$ EGT $ "RKN52"	Orden empíricop:
1 ($h = 0.4$)	1.04E-02	-	4.53E-04	-	2.80E-04	-
2 ($h = 0.2$)	7.18E-04	3.852	1.50E-05	4.916	1.15E-05	4.611
3 ($h = 0.1$)	4.72E-05	3.927	4.82E-07	4.960	4.06E-07	4.821
4 ($h = 0.05$)	3.03E-06	3.967	1.53E-08	4.980	1.35E-08	4.914
5 ($h = 0.025$)	1.92E-07	3.982	4.80E-10	4.991	4.33E-10	4.958
PROMEDIO		$\bar{p} = 3.931$		$\bar{p} = 4.962$		$\bar{p} = 4.826$

Tabla 7.19. Determinación del orden empírico de los métodos **RKN** ensayados para problemas independientes de y' aplicados al **PVI (7.12)**. $|EGT|$ en nodo extremo $x = 2$.

Al obtener convergencia progresiva hacia el orden teórico, mejorando el promedio de orden empírico en cada densificado, se verifica la sensibilidad de los métodos **RKN** ensayados para cada una de las mallas propuestas en la **Tabla 7.16**, validando el procedimiento y el empleo del $|EGT|$ exclusivamente en el nodo extremo $x = 2$. La gráfica definitiva con las líneas de tendencia cuya pendiente es el orden calculado es la dada por la **Figura 7.15**. Se vuelven a emplear magnitudes logarítmicas en escala lineal tanto para las ordenadas como para las abscisas según (7.9).

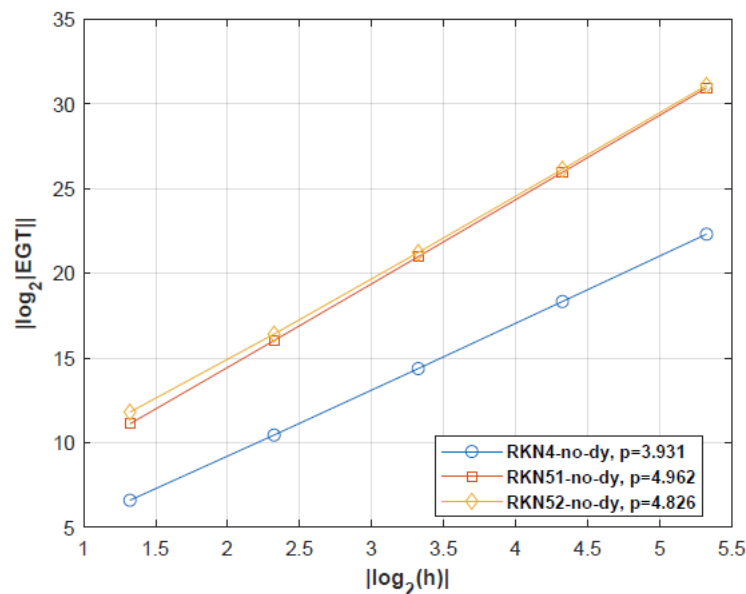


Figura 7.15. Determinación del orden empírico de los métodos **RKN** para problemas independientes de y' aplicados al **PVI (7.12)**.



El error relativo cometido respecto del orden teórico se indica en la **Tabla 7.20**:

<i>Método ensayado</i>	<i>Orden teórico</i>	<i>Orden empírico</i>	<i>Error cometido (%)</i>
<i>"RKN4_no_dy"</i>	4	3.931	1.73 %
<i>"RKN51_no_dy"</i>	5	4.962	0.76 %
<i>"RKN52_no_dy"</i>	5	4.826	3.48 %

Tabla 7.20. Errores de los órdenes empíricos respecto de los teóricos correspondientes a los métodos *RKN* ensayados para problemas independientes de y' y aplicados al **PVI (7.12)**.

7.3.3. Conclusiones del experimento.

Las conclusiones que pueden extraerse de este experimento numérico, en concordancia con todos los resultados presentados, se indican a continuación:

- i. En la determinación del orden empírico de un método numérico, **no necesariamente se obtiene una convergencia progresiva hacia el valor del orden teórico al disminuir el paso**, pues si las **mallas no son lo suficientemente sensibles** al comportamiento de dicho método, el $|EGT|$ tiende a estabilizarse y a no mostrar la tendencia necesaria (aunque el método se comporte correctamente). Esto ocurre en métodos de orden elevado, como el *Runge-Kutta* implícito de *Gauss* (3 etapas y orden teórico 6, sección **3.2.6** de la memoria).
- ii. Métodos numéricos aplicados a problemas monótonamente crecientes (como los **PVIs (7.10) y (7.12)**) suelen presentar un comportamiento del $|EGT|$ asociado a su aproximación a la solución exacta de la misma naturaleza que la tendencia de la solución, siempre y cuando las mallas sean lo suficientemente sensibles a las aproximaciones.
- iii. **Los métodos numéricos implícitos presentan ventajas respecto a los explícitos**, en cuanto a la obtención de un orden más elevado y convergencia más rápida con una complejidad algorítmica no necesariamente más grande, salvo la resolución del sistema no lineal que calcula sus etapas (normalmente menos que las de un explícito) en cada paso ("*GRK6s1o*" del **ANEXO I**).

Se recuerda al lector la posibilidad de consultar el código de **MATLAB** desarrollado ("**PVI_Ordenes_Metodos**") para la realización de este experimento numérico en el **ANEXO IV** de la memoria.

7.4. Experimento numérico 3. Problema de prueba. Ensayos con *splines* cúbicos.

Habiendo experimentado ya con los métodos numéricos *Runge-Kutta* y *Runge-Kutta-Nyström* del **Capítulo 3** (y **ANEXO I**), verificando su orden empírico mediante densificados de mallas equiespaciadas apropiados y estudiando la forma que puede tomar el **|EGT|** para cada uno de ellos, llega el momento de observar el comportamiento de las estrategias de interpolación de soluciones basadas en trazadores polinomiales (*splines* y *pseudo-splines*), las cuales se fundamentan en el marco teórico desarrollado en los **Capítulos 4** y **5**.

Este experimento numérico concierne a la experimentación exclusiva con *splines* cúbicos, y su objetivo principal es determinar si son lo suficientemente fiables para ser utilizados en la interpolación de las soluciones discretas obtenidas en los nodos de malla (2.11) del **PVI** al que se apliquen.

El **PVI** de referencia en este experimento viene dado por (7.14), el cual es lineal de 2º orden y del tipo (2.7). Tiene solución analítica polinómica cúbica (7.15):

$$\begin{cases} y''(x) = xy'(x) - 3y(x) + x^2 + 8x + 5 \\ y(0) = 1, \quad y'(0) = 1 \\ x \in [0,1] \end{cases}$$

Ecuación (7.14)

$$y(x) = x^3 + x^2 + x + 1$$

Ecuación (7.15)

Considérense las siguientes mallas equiespaciadas que discretizan el dominio de (7.14), indicadas en la **Tabla 7.21** y construidas según (2.13).

<i>Identificador de malla (ID)</i>	<i>Paso uniforme (h)</i>	<i>Nº de subintervalos (n)</i>
1	0.25	4
2	0.125	8
3	0.08 $\hat{3}$	12
4	0.0625	16
5	0.0 $\hat{3}$	30
6	0.03125	32

Tabla 7.21. Mallas equiespaciadas en el intervalo [0,1].

Las mallas x_1 , x_2 , x_4 y x_6 se densifican según potencias de base 2, de tal modo que el número de subintervalos n responde a 2^m , con $m = 2, 3, 4, 5$. El resto corresponden a un número de subintervalos que no sigue ninguna ley respecto al densificado anteriormente citado. La forma que toma (7.15) se indica en la **Figura 7.16**, en la cual se emplea interpolación lineal por defecto de **MATLAB** y la malla x_4 de la tabla anterior.

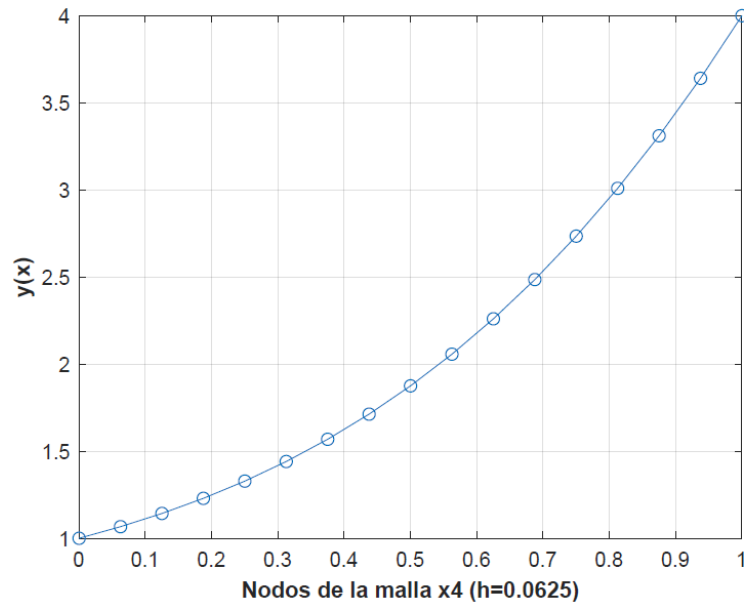


Figura 7.16. Gráfica de la solución analítica $y(x)$ del PVI (7.14) en los nodos de la malla x_4 . Interpolación lineal por defecto de MATLAB.

7.4.1. Experimento con *splines* cúbicos. Error de redondeo del esquema recursivo. Aritmética de coma flotante.

Al Interpolarse con un *spline* cúbico la solución discreta exacta de un PVI cuya solución es polinómica también cúbica (caso de (7.14) y (7.15)), el resultado debería de ser, en ausencia de errores de redondeo, la propia solución exacta. Es por ello que se conoce a priori el valor que han de tomar los parámetros (coeficientes) del trazador S en cada intervalo delimitado por dos nodos consecutivos. De hecho, en el intervalo $[t_{k-1}, t_k]$, se tendrá (7.16):

$$\begin{aligned}
 y(x) &= x^3 + x^2 + x + 1 = ((x - t_{k-1}) + t_{k-1})^3 + ((x - t_{k-1}) + t_{k-1})^2 + \\
 &+ ((x - t_{k-1}) + t_{k-1}) + 1 = (x - t_{k-1})^3 + (3t_{k-1} + 1)(x - t_{k-1})^2 + \\
 &+ (3t_{k-1}^2 + 2t_{k-1} + 1)(x - t_{k-1}) + (t_{k-1}^3 + t_{k-1}^2 + t_{k-1} + 1) = \\
 &\boxed{a_{k0} + a_{k1}(x - t_{k-1}) + a_{k2}(x - t_{k-1})^2 + a_{k3}(x - t_{k-1})^3}
 \end{aligned}$$

Ecuación (7.16)

Se deduce que la expresión anterior es equivalente a (7.15) pero en la base de polinomios de (4.15), con la cual se construyen los *splines* citados en MATLAB. Por tanto, se tienen los siguientes valores conocidos a_{ki} del trazador (7.17):

$$\begin{aligned}
 a_{k0} &= t_{k-1}^3 + t_{k-1}^2 + t_{k-1} + 1 && \text{a)} \\
 a_{k1} &= 3t_{k-1}^2 + 2t_{k-1} + 1 && \text{b)} \\
 a_{k2} &= 3t_{k-1} + 1 && \text{c)}
 \end{aligned}$$

$$a_{k3} = 1$$

d)

Ecuación (7.17)

Si se utiliza la función “*spline3int*” disponible en el **ANEXO II**, dichos coeficientes pueden obtenerse fácilmente en cada intervalo (con cada una de las mallas de la **Tabla 7.21**) y ser comparados con los valores de (7.17), con el objetivo de observar si el esquema recursivo planteado en (4.14) es sensible a posibles errores de redondeo acumulados o, en cambio, es adecuado y perfectamente aplicable a multitud de problemas posteriores. Esos errores, en caso de existir, serán más susceptibles de poder causar el fallo del método en los últimos intervalos surgidos de cada malla de ensayo (acumulación progresiva), por lo que se fija el interés por el valor de los coeficientes y la forma de la solución aproximada en el entorno del nodo extremo $x = 1$ por la izquierda, tomando el último subintervalo de cada malla, es decir, aquel que cumple $k = n$.

Procediendo con todas las mallas se indica el valor del coeficiente a_{n3} (el más susceptible de fallo), junto con el valor de la solución exacta en los nodos del último subintervalo $[t_{n-1}, t_n]$ empleando para ello aritmética de 16 cifras significativas, la cual corresponde a la aritmética de coma flotante por defecto que implementa **MATLAB** (doble precisión) en todos sus cálculos numéricos. Se tiene la **Tabla 7.22**:

	<i>Nodos</i> $[t_{n-1}, t_n]$	<i>Sol. Exacta</i> $y(x)$	<i>Coficiente</i> a_{n3}
Malla x_1 ($h = 0.25$)	0.750	2.7343750000000000	1.0000000000000000
	1.000	4.0000000000000000	
Malla x_2 ($h = 0.125$)	0.875	3.3105468750000000	1.0000000000000000
	1.000	4.0000000000000000	
Malla x_3 ($h = 0.08\hat{3}$)	0.916	3.527199074074070	1.000000343804162
	1.000	4.0000000000000000	
Malla x_4 ($h = 0.0625$)	0.9375	3.640380859375000	1.0000000000000000
	1.0000	4.0000000000000000	
Malla x_5 ($h = 0.0\hat{3}$)	0.96551724137931	3.797818688753127	30066.25949573336
	1.0000000000000000	4.0000000000000000	
Malla x_6 ($h = 0.03125$)	0.96875	3.816375732000000	1.0000000000000000
	1.00000	4.0000000000000000	

Tabla 7.22. Valores del coeficiente a_{n3} para cada una de las mallas de ensayo y solución exacta en nodos del último intervalo generado $[t_{n-1}, t_n]$.

Los valores del coeficiente a_{n3} difieren notablemente respecto del valor teórico para las mallas x_3 y, sobre todo, x_5 , dando lugar a aproximaciones inaceptables a las solución (7.15) del **PVI** (7.14). En cambio, para aquellas mallas generadas mediante potencias de base 2 el valor de a_{n3} calculado coincide exactamente con el teórico. Se detecta que pequeños errores, en general inevitables (por el empleo de aritmética de coma flotante) se transmiten y amplifican muy rápidamente deteriorando la calidad de la aproximación hasta hacerla inaceptable. Este es un comportamiento típico de un algoritmo inestable.

Para las mallas x_1 , x_2 , x_4 y x_6 , la aritmética de coma flotante con 16 cifras significativas proporciona los mismos valores que la aritmética exacta a la hora de obtener las soluciones nodales sin redondeos, al precisar todas ellas de menos dígitos significativos que los totales disponibles.

Si en cada subintervalo se pierden dígitos, progresivamente el esquema evoluciona al fallo al depender la aproximación k de su inmediatamente anterior $k - 1$ (o la $k + 1$ de la k). **A pesar de tener la certeza teórica de que un trazador cúbico ha de reproducir exactamente un polinomio de a lo sumo el mismo grado, el esquema de cálculo no lo lleva a la práctica debido a su carácter inestable.**

Para tener evidencia gráfica de lo anterior, se aportan las **Figuras 7.17 y 7.18**, las cuales corresponden a la solución aproximada interpolada con los *splines* cúbicos generados para las mallas x_3 (“**s3int3**”) y x_5 (“**s3int5**”), y en los últimos intervalos correspondientes a cada una de ellas $[t_{n-1}, t_n = 1]$. **En dichas gráficas también se indica la solución exacta que el *spline* debería de seguir correctamente sin la presencia de inestabilidades.**

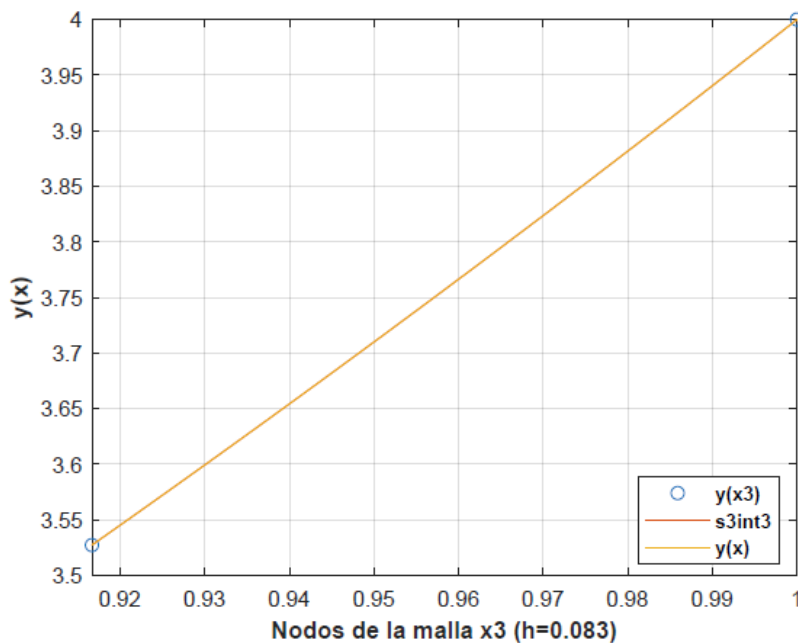


Figura 7.17. Gráfica de la solución interpolada en los nodos de la malla x_3 mediante el empleo de *splines* cúbicos. Solución aproximada del **PVI (7.14)**.

La figura anterior no es capaz de mostrar gráficamente la distorsión, teniendo que recurrir a la **Tabla 7.22** para observar la discrepancia respecto del coeficiente de referencia teórico $a_{n3} = 1$. No obstante, el resultado obtenido es coherente, pues la inestabilidad de la estrategia interpoladora todavía no es lo suficientemente grande como para provocar fallos a gran escala (préstese atención al orden de magnitud del valor experimental de a_{n3}), **aunque es**

preciso dejar constancia que el problema sigue existiendo estando ligado explícitamente a la naturaleza de la estrategia empleada.

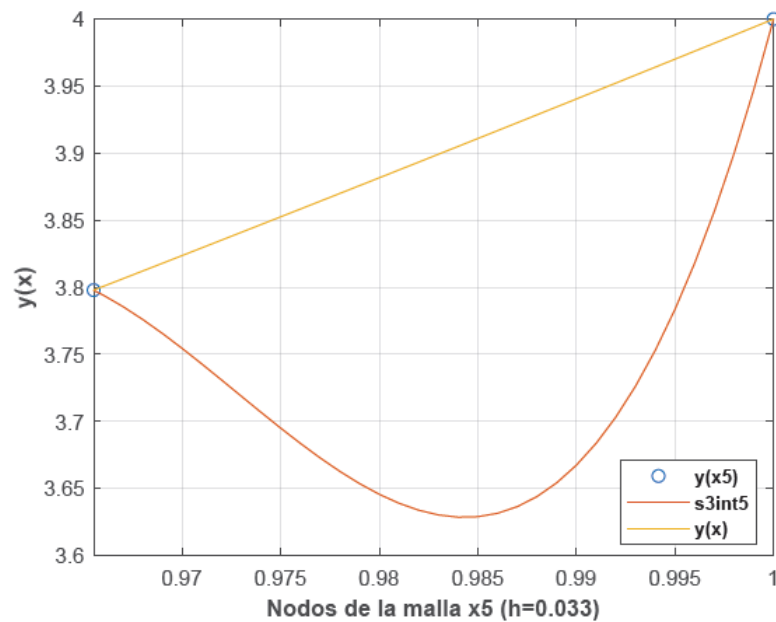


Figura 7.18. Gráfica de la solución interpolada en los nodos de la malla x_5 mediante el empleo de *splines* cúbicos. Solución aproximada del *PVI* (7.14).

Finalmente, esta última gráfica sí muestra claramente lo que la *Figura 7.17* no hacía, pues ahora la distorsión es clara (orden de magnitud del a_{n3} experimental completamente desproporcionado) y demuestra la inestabilidad del esquema recursivo planteado en la sección 4.4.3 del *Capítulo 4*, por acumulación y amplificación progresiva del error de redondeo fruto de la pérdida de dígitos en aritmética de coma flotante.

7.4.2. Conclusiones del experimento.

Este breve pero importante experimento numérico arroja una serie de conclusiones determinantes sobre los *splines* cúbicos, pero también tiene otra serie de implicaciones que a continuación se detallan:

- i. El esquema recursivo para la determinación de parámetros (coeficientes) que determinan unívocamente el *spline* cúbico interpolador de la solución de un *PVI* de 2º orden (sea lineal o no) **falla completamente por ser inestable**.
- ii. Las consideraciones expuestas son perfectamente extrapolables a todas las estrategias de interpolación empleando *pseudo-splines* cuárticos (sección 5.2 del *Capítulo 5*), puesto que el principio en el que se basa su determinación unívoca es el mismo que provoca el fallo de los *splines*.
- iii. Los *splines* cuadráticos (expuestos al comienzo del *Capítulo 2*) no son susceptibles de fallo e inestabilidad atendiendo a las consideraciones de este experimento, pues carecen a priori de un coeficiente

problemático que perturbe el esquema recursivo para la determinación de parámetros en cada intervalo (sección 4.2 del **Capítulo 4**).

- iv. Las estrategias de interpolación restantes y basadas en polinomios osciladores (sección 5.3 del **Capítulo 5**) no se fundamentan en ningún esquema recursivo demostrado inestable, por lo que pueden ser aplicadas sin preocupaciones relativas al truncado progresivo de dígitos.

Algunas consideraciones adicionales desprendidas de estas conclusiones se comentan en el **Capítulo 8** como **líneas futuras de investigación basadas en el material disponible y aprovechable de este trabajo**, relativas al empleo de *splines* cúbicos y *pseudo-splines* cuárticos interpoladores. El código de **MATLAB** (“*PVI_s3int*”) que ha sido necesario utilizar para realizar este experimento numérico puede ser consultado en el **ANEXO IV**.

7.5. Experimento numérico 4. Deflexión de una viga biapoyada.

Un problema muy común en *Ingeniería Civil* e *Ingeniería Mecánica* concierne a la **deflexión de una viga biapoyada** en sus extremos y sometida a una combinación de cargas longitudinal y transversal **coplanares** [12]. Mediante las hipótesis establecidas por la *Mecánica del Sólido Deformable* [65], se alcanza un equilibrio en configuración deformada que da lugar al **análisis estático de la viga**, fundamentado en la obtención de su deflexión como función de las cargas propuestas. Como problema particularizado, considérese una viga de longitud L sometida a una **carga transversal simétrica senoidal** $q(x)$ y a una **carga longitudinal uniforme** P (*Figura 7.19*).

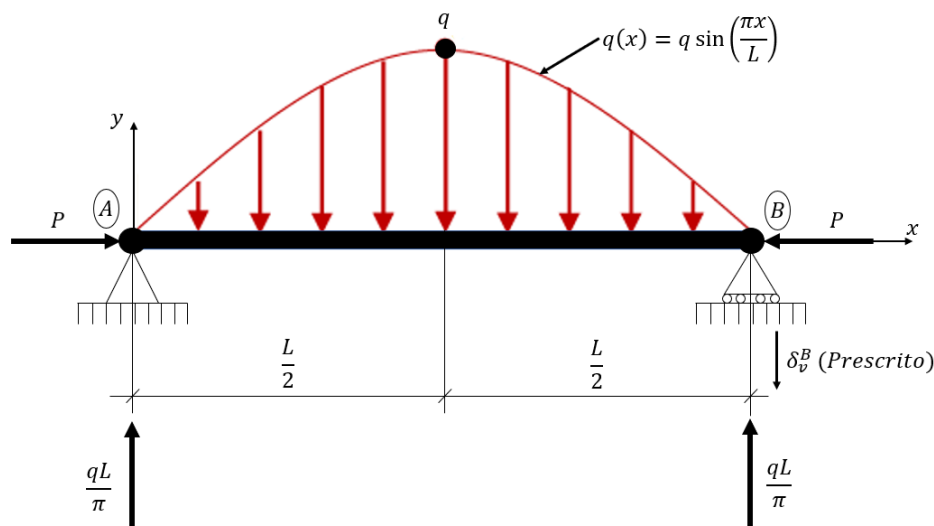


Figura 7.19. Viga biapoyada sometida a carga transversal senoidal simétrica $q(x)$ y longitudinal uniforme P .

La configuración de apoyos es *fijo(A)-móvil(B)* para que la viga sufra los efectos de la carga longitudinal. Haber considerado desplazamientos horizontales truncados en ambos apoyos A y B implicaría una flexión simple libre de solicitaciones longitudinales en dirección x , ya que no se aplican cargas intraelementales entre apoyos. Hay que destacar la existencia de un desplazamiento vertical δ_v^B prescrito en B respecto de la posición nominal. Esto suele realizarse a la hora de modelar asentamientos. La **Figura 7.19** también muestra las reacciones verticales surgidas en los apoyos, para garantizar el equilibrio de la viga en dicha dirección.

Antes de obtener el problema matemático cuya resolución conduce a la deflexión de la viga, es preciso matizar que se va a emplear la **Teoría de 2º orden** (puede profundizarse su conocimiento en [66] y [67]). Esta teoría se basa en **grandes desplazamientos**, o lo que es lo mismo, utiliza ecuaciones de equilibrio surgidas de la configuración deformada de la viga para detectar los efectos de la deformación debido a las cargas aplicadas. Es de especial interés cuando no es posible afirmar que el modelo de pequeños desplazamientos o de equilibrio en configuración indeformada (simplificado) presenta resultados físicos coherentes. El *Diagrama de Sólido Libre* de una porción de viga se plantea en la **Figura 7.20** ($v(x)$ es la deflexión de la viga incógnita que pretende calcularse).

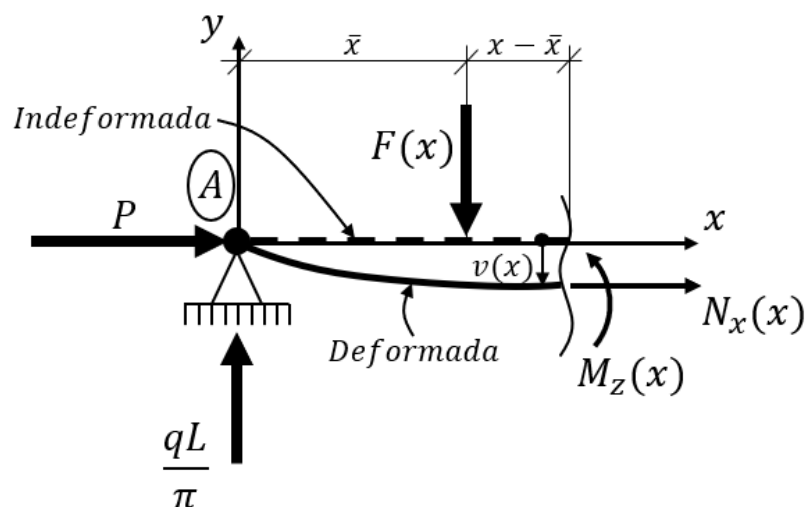


Figura 7.20. Diagrama de Sólido Libre de una porción arbitraria de viga.

La carga $F(x)$ es la resultante de la distribución senoidal $q(x)$ en el intervalo $[0, x]$. Su valor es el siguiente (7.18):

$$F(x) = \int_0^x q(x) dx = q \int_0^x \sin\left(\frac{\pi x}{L}\right) dx = \frac{qL}{\pi} \left(1 - \cos\left(\frac{\pi x}{L}\right)\right)$$

Ecuación (7.18)



Para poder situar dicha carga, se precisa obtener su punto de aplicación. Este punto coincide con el centroide \bar{x} de la distribución senoidal $q(x)$ en la dirección x . Su valor es el dado por (7.19):

$$\bar{x} = \frac{1}{F(x)} \int_0^x x dF(x) = \frac{\pi \int_0^x x \sin\left(\frac{\pi x}{L}\right) dx}{L \left(1 - \cos\left(\frac{\pi x}{L}\right)\right)} = \frac{L \sin\left(\frac{\pi x}{L}\right) - \pi x \cos\left(\frac{\pi x}{L}\right)}{\pi \left(1 - \cos\left(\frac{\pi x}{L}\right)\right)}$$

Ecuación (7.19)

Si se quiere situar el centroide \bar{x} con respecto a la sección de corte y no el origen de coordenadas A , puede utilizarse (7.20):

$$x - \bar{x} = \frac{\pi x - L \sin\left(\frac{\pi x}{L}\right)}{\pi \left(1 - \cos\left(\frac{\pi x}{L}\right)\right)}$$

Ecuación (7.20)

Conocidas las expresiones anteriores, se plantea el **equilibrio de momentos en configuración deformada** respecto a la sección de corte arbitraria x (7.21).

$$M_z(x) = -F(x)(x - \bar{x}) + Pv(x) + \frac{qL}{\pi}x$$

Ecuación (7.21)

Ahora, aproxímese la curvatura de la deformada con (7.22):

$$v''(x) \cong \frac{M_z(x)}{EI_z(x)}$$

Ecuación (7.22)

E es el **Módulo de Young del material** con el que se ha fabricado la viga, mientras que $I_z(x)$ representa el **momento de inercia respecto de un eje perpendicular al papel** variable con la coordenada x , pues la viga no tiene porqué ser de sección constante. Sustituyendo (7.20) y (7.22) en (7.21), surge (7.23):

$$EI_z(x)v''(x) = Pv(x) + \frac{qL^2}{\pi^2} \sin\left(\frac{\pi x}{L}\right)$$

Ecuación (7.23)

La expresión anterior es una **EDO de 2º orden lineal no homogénea de coeficientes variables e independiente de la primera derivada v'** (del tipo (2.3)). Expresándola en forma normal, y añadiendo las condiciones de contorno en desplazamientos verticales v concordantes con los apoyos que aportan el

sentido físico completo al problema de la **Figura 7.19** (sin olvidar el desplazamiento prescrito δ_v^B en B con su signo), se obtiene **(7.24)**:

$$\begin{cases} v''(x) = \left(\frac{P}{EI_z(x)}\right)v(x) + \frac{qL^2}{\pi^2 EI_z(x)} \sin\left(\frac{\pi x}{L}\right) \\ v(0) = 0, \quad v(L) = -\delta_v^B \\ x \in [0, L] \end{cases}$$

Ecuación **(7.24)**

Es evidente que se ha obtenido un **Problema de Valores en la Frontera**, formulado en términos de una **EDO** cuya naturaleza es la descrita anteriormente en el intervalo $[0, L]$ (**PVF** del tipo **(6.4)**). Ahora se precisa obtener una particularización numérica del mismo. Si el momento de inercia $I_z(x)$ sigue una distribución lineal desde A hasta B , se tiene la expresión **(7.25)**:

$$I_z(x) = \left(\frac{I_B - I_A}{L}\right)x + I_A$$

Ecuación **(7.25)**

Se han considerado I_A e I_B como los valores de I_z en $x = 0$ y $x = L$, respectivamente. Además, $I_B = 10I_A$. La **Tabla 7.23** recoge todos los valores numéricos necesarios para particularizar **(7.24)**, expresados en unidades del *Sistema Internacional*.

<i>Magnitud</i>	<i>Descripción</i>	<i>Valor (Unidades SI)</i>
L	Longitud de la viga	8 m
δ_v^B	Desplazamiento prescrito en B	-0.02 m
E	Módulo de Young (acero)	$2.1 \cdot 10^{11}$ Pa
P	Carga longitudinal	$5 \cdot 10^4$ N
q	Máxima carga transversal	$5 \cdot 10^4$ N/m
I_A	Inercia en $x = 0$	$5 \cdot 10^{-5}$ m ⁴
I_B	Inercia en $x = L$	$5 \cdot 10^{-4}$ m ⁴

Tabla 7.23. Valores numéricos para particularizar la viga biapoyada.

Finalmente, sustituyendo todos los valores en **(7.24)**, se obtiene el **PVF** a resolver **(7.26)**:

$$\begin{cases} v''(x) = \left(\frac{8}{210(9x + 8)}\right)v(x) + \frac{256\sin\left(\frac{\pi x}{8}\right)}{105\pi^2(9x + 8)} \\ v(0) = 0, \quad v(L) = -0.02 \text{ m} \\ x \in [0, 8] \text{ m} \end{cases}$$

Ecuación **(7.26)**

7.5.1. Solución numérica. Aplicación del método del disparo lineal.

Este problema, tal y como ha sido formulado, carece de solución analítica. No obstante, se dispone de los medios necesarios para computar su solución numérica de forma eficiente y rápida. Considérense, en primer lugar, las dos mallas de nodos discretas dadas por la **Tabla 7.24**, las cuales van a ser empleadas en la construcción de las soluciones posteriormente empleando estrategias de interpolación diversas.

<i>Malla</i>	<i>Alias</i>	<i>Subintervalos</i>	<i>Paso uniforme h</i>
1	x_1	1 \rightarrow [0,2)	$h_{11} = 0.5$
		2 \rightarrow [2,6)	$h_{12} = 1$
		3 \rightarrow [6,8]	$h_{13} = 0.5$
2	x_2	1 \rightarrow [0,2)	$h_{21} = 1$
		2 \rightarrow [2,6)	$h_{22} = 0.25$
		3 \rightarrow [6,8]	$h_{23} = 1$

Tabla 7.24. Mallas no equiespaciadas en el intervalo [0,8] (m).

Con cada malla se adopta una estrategia distinta. La primera se construye teniendo en cuenta que lo más probable es que la solución de (7.26) varíe rápidamente en el entorno de los apoyos de la viga, ya que al no existir restricción de giro pero sí de desplazamiento vertical la predisposición a ello es evidente. En cambio, se adopta un paso menos denso en las secciones centrales. Esto contrasta con la estrategia de la segunda malla, pues ésta considera inicialmente que el mayor densificado ha de efectuarse en el centro para poder así aproximar mejor los cambios de curvatura debidos a la **cuasi-simetría** del problema (nótese que existe un desplazamiento prescrito en B de valor δ_v^B como condición de contorno y la inercia I_z es variable).

Al tratarse de un **PVF**, la estrategia de resolución propuesta en este trabajo es el **método del disparo lineal**. Teniendo en cuenta las condiciones de contorno simples de (7.26), se plantean los siguientes **PVs** dados por (7.27) y (7.28), en concordancia con lo expuesto en la sección 6.4.1 del **Capítulo 6**.

$$\text{No homogéneo} \rightarrow \begin{cases} v_1''(x) = \left(\frac{8}{210(9x+8)} \right) v_1(x) + \frac{256 \sin\left(\frac{\pi x}{8}\right)}{105\pi^2(9x+8)} \\ v_1(0) = 0, \quad v_1'(0) = 0 \\ x \in [0,8] \text{ m} \end{cases}$$

Ecuación (7.27)

$$\text{Homogéneo} \rightarrow \begin{cases} v_2''(x) = \left(\frac{8}{210(9x+8)} \right) v_2(x) \\ v_2(0) = 0, \quad v_2'(0) = 1 \\ x \in [0,8] \text{ m} \end{cases}$$

Ecuación (7.28)

La solución se obtiene aplicando la expresión (6.18) particularizada a los datos actuales, al no anularse la solución de (7.28) en el último nodo de las mallas de la **Tabla 7.24**.

7.5.1.1. Obtención de la solución discreta aproximada del PVF para cada malla de ensayo.

Al tratarse de un problema independiente de v' , sus **PVIs** asociados también lo son, por lo que se recurre a un método numérico específicamente diseñado para este tipo de problemas. El método elegido es el *Runge-Kutta-Nyström* de orden 4 independiente de la primera derivada (sección 3.3.5.1 del **Capítulo 3**), cuyo alias en **MATLAB** es "*RKN4_no_dy*", disponible en el **ANEXO I** y con orden empírico demostrado previamente en el experimento numérico 2 de este capítulo (7.3).

Las soluciones discretas del **PVF**, para cada una de las mallas de la **Tabla 7.24**, se indican en las **Tablas 7.25** y **7.26**, en donde también se muestran las soluciones obtenidas para los **PVIs (7.27)** y **(7.28)**, denotadas con un segundo subíndice que corresponde a la malla de evaluación. Se emplea notación científica con 3 cifras significativas.

Malla x_1	PVI(7.27): v_{11}	PVI(7.28): v_{12}	PVF(7.26): v_{x_1}
0.00	0.00E+00	0.00E+00	0.00E+00
0.50	1.97E-04	5.00E-01	-1.11E-02
1.00	1.32E-03	1.00E+00	-2.12E-02
1.50	3.81E-03	1.50E+00	-3.00E-02
2.00	7.89E-03	2.00E+00	-3.72E-02
3.00	2.11E-02	3.01E+00	-4.67E-02
4.00	4.08E-02	4.02E+00	-4.97E-02
5.00	6.61E-02	5.03E+00	-4.72E-02
6.00	9.57E-02	6.05E+00	-4.04E-02
6.50	1.12E-01	6.56E+00	-3.59E-02
7.00	1.28E-01	7.07E+00	-3.09E-02
7.50	1.45E-01	7.58E+00	-2.55E-02
8.00	1.62E-01	8.09E+00	-2.00E-02

Tabla 7.25. Valores numéricos de las soluciones discretas aproximadas en la malla x_1 de los **PVIs (7.27)** y **(7.28)** posteriormente combinadas por disparo lineal para obtener la solución del **PVF (7.26)**.

Malla x_2	PVI(7.27): v_{21}	PVI(7.28): v_{22}	PVF(7.26): v_{x_2}
0.00	0.00E+00	0.00E+00	0.00E+00
1.00	1.29E-03	1.00E+00	-2.12E-02
2.00	7.84E-03	2.00E+00	-3.72E-02
2.25	1.05E-02	2.25E+00	-4.02E-02
2.50	1.36E-02	2.51E+00	-4.28E-02
2.75	1.71E-02	2.76E+00	-4.49E-02
3.00	2.10E-02	3.01E+00	-4.67E-02

3.25	2.54E-02	3.26E+00	-4.80E-02
3.50	3.01E-02	3.51E+00	-4.90E-02
3.75	3.52E-02	3.77E+00	-4.95E-02
4.00	4.07E-02	4.02E+00	-4.97E-02
4.25	4.65E-02	4.27E+00	-4.96E-02
4.50	5.27E-02	4.52E+00	-4.91E-02
4.75	5.92E-02	4.78E+00	-4.83E-02
5.00	6.60E-02	5.03E+00	-4.72E-02
5.25	7.30E-02	5.28E+00	-4.59E-02
5.50	8.03E-02	5.54E+00	-4.43E-02
5.75	8.79E-02	5.79E+00	-4.25E-02
6.00	9.56E-02	6.05E+00	-4.04E-02
7.00	1.28E-01	7.07E+00	-3.09E-02
8.00	1.62E-01	8.09E+00	-2.00E-02

Tabla 7.26. Valores numéricos de las soluciones discretas aproximadas en la malla x_2 de los *PVIs* (7.27) y (7.28) posteriormente combinadas por disparo lineal para obtener la solución del *PVF* (7.26).

A la vista de los resultados de cada tabla, lo primero que debe de apreciarse es que la combinación de soluciones ha sido realizada correctamente según (6.18), pues efectivamente (7.28) no se anula en el nodo final $x = 8$ (m). En cuanto a las condiciones de contorno, éstas se cumplen perfectamente. Hay que denotar también que el signo obtenido de la solución del *PVF* (7.26) es negativo, en concordancia con el sistema de referencia de las Figuras 7.19 y 7.20.

Ahora, para poder dar un sentido más físico e ingenieril a las soluciones obtenidas, se proponen las **estrategias de interpolación** de la **Tabla 7.27**, entendiéndose este experimento como un **procedimiento dual** que ensaya el comportamiento de los métodos planteados en este trabajo, aprovechándose del interés que un caso real como el de la **Figura 7.19** tiene desde el punto de vista de un ingeniero.

Estrategia	Sección	Alias trazadores malla x_1		Alias trazadores malla x_2		
		<i>PVI</i>	<i>PVF</i> (7.26)	<i>PVI</i>	<i>PVF</i> (7.26)	
<i>Splines cuadráticos</i>	2.2	(7.27)	"s2int11"	"s2int1"	(7.27)	"s2int21"
		(7.28)	"s2int12"		(7.28)	"s2int22"
<i>Pseudo-splines cúbicos</i>	2.3	(7.27)	"ps3int11"	"ps3int1"	(7.27)	"ps3int21"
		(7.28)	"ps3int12"		(7.28)	"ps3int22"
<i>Pseudo-splines quinticos</i>	2.4	(7.27)	"ps5int11"	"ps5int1"	(7.27)	"ps5int21"
		(7.28)	"ps5int12"		(7.28)	"ps5int22"

Tabla 7.27. Cuadro sinóptico resumen de estrategias de interpolación adoptadas para el *PVF* (7.26).

Nótese que se ha asignado un alias a los *splines* o *pseudo-splines* generados para cada estrategia. **Estos trazadores se combinan compartiendo mallas con el mismo criterio que el empleado para las soluciones discretas aproximadas aplicando el método del disparo lineal, pero ahora empleando la expresión (6.21) una vez evaluados.** El marco teórico de las estrategias de interpolación puede consultarse en los **Capítulos 4 y 5**.

7.5.1.2. Estrategia de interpolación con *splines* cuadráticos.

La **Figura 7.21** muestra la deflexión de la viga interpolada en los nodos de la malla x_1 , empleando el trazador resultante “*s2int1*”, obtenido por combinación según (6.21) de los trazadores “*s2int11*” y “*s2int12*”.

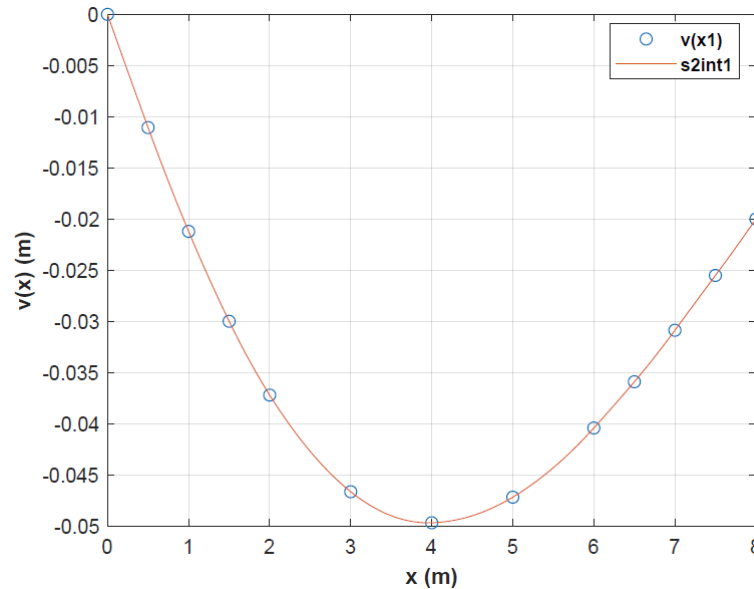


Figura 7.21. Deflexión de la viga biapoyada interpolada en los nodos de la malla x_1 mediante aplicación del método del disparo lineal. Solución del *PVF* (7.26).

Del mismo modo, se puede obtener la deflexión de la viga interpolando ahora las soluciones discretas de la malla x_2 (**Figura 7.22**). En concordancia con la **Tabla 7.27**, se obtiene el *spline* “*s2int2*”, habiendo combinado previamente los *splines* “*s2int21*” y “*s2int22*”.

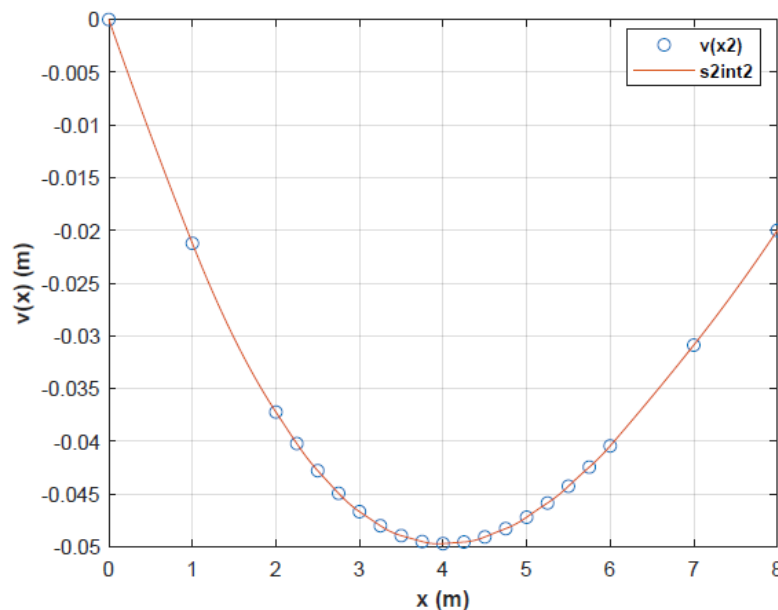


Figura 7.22. Deflexión de la viga biapoyada interpolada en los nodos de la malla x_2 mediante aplicación del método del disparo lineal. Solución del *PVF* (7.26).

A la vista de los resultados gráficos obtenidos para cada una de las soluciones discretas en mallas no equiespaciadas, cabe destacar que, aparentemente, no se consiguen mejoras de interpolación lo suficientemente apreciables como para ser tenidas en cuenta. Por tanto, se realizan dos zooms a ambas gráficas en el intervalo $[3,5](m)$ (*Figura 7.23*) y $[7,8](m)$ (*Figura 7.24*), las cuales muestran conjuntamente los trazadores “s2int1” y “s2int2” para compararlos.

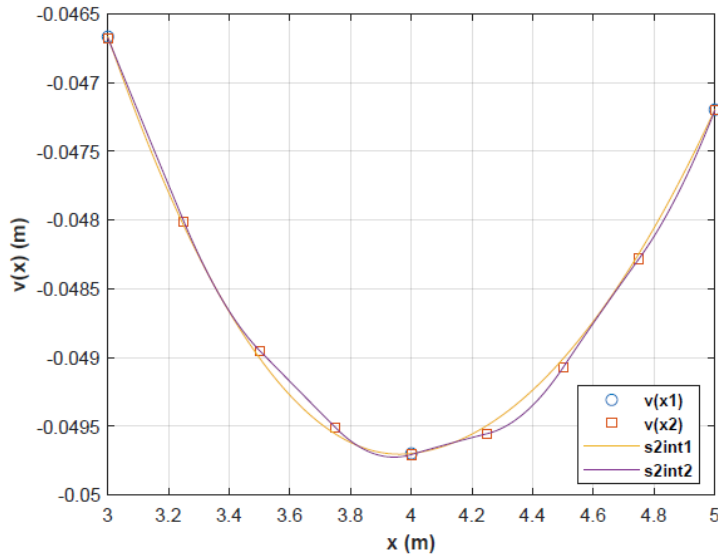


Figura 7.23. Zoom de las *Figuras 7.21* y *7.22* en el intervalo $[3,5](m)$.

Los resultados son sorprendentes. La *Figura 7.23* indica que un **densificado local en zonas de cambio de curvatura no implica una mejora en la regularidad**, puesto de manifiesto esto último al existir fluctuaciones impropias de la realidad física del problema en el trazador “s2int2”. Se recuerda al lector que en la sección **4.3.1** del **Capítulo 4** tan sólo se impuso continuidad C^1 del *spline*, por lo que todos los esfuerzos para controlar la continuidad de la segunda derivada son intrascendentes. En otras palabras, **densificar no implica mejorar**.

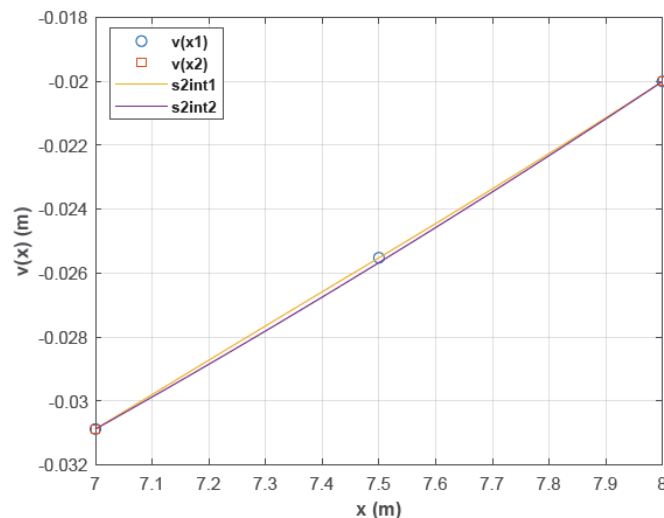


Figura 7.24. Zoom de las *Figuras 7.21* y *7.22* en el intervalo $[7,8](m)$.

En cuanto al intervalo $[7,8](m)$, la **Figura 7.24** indica una correspondencia entre ambos trazadores bastante coherente. Al ser la deflexión de la viga ahora monótonamente creciente (puesto de manifiesto en los valores numéricos de las **Tablas 7.25** y **7.26**), no existen los problemas de la **Figura 7.23** con “**s2int2**”, por lo que se puede concluir que, **a falta de un criterio de tolerancia que informe de la idoneidad de una interpolación en particular**, se pueden dar por válidas ambas alternativas en el entorno de los apoyos *A* y *B* (recuérdese que el problema es **cuasi-simétrico**, ver **Figura 7.19**).

7.5.1.3. Estrategia de interpolación con *pseudo-splines* cúbicos.

Una alternativa que puede subsanar los problemas en la regularidad de la solución puesta de manifiesto en la **Figura 7.23** es el empleo de un trazador que imponga continuidad C^2 . Por tanto, se repite el experimento de **7.5.1.1** con *pseudo-splines* cúbicos obtenidos con interpolación de *Hermite* (sección **5.3.2**, **Capítulo 5**), los cuales además interpolan a la derivada discreta de la solución aproximada, por lo que aprovechan el esfuerzo computacional del método numérico “**RKN4_no_dy**” utilizado previamente.

Estudiando únicamente el comportamiento de los trazadores en la zona central del dominio $[3,5](m)$, se tiene la **Figura 7.25**, comparativa entre “**ps3int1**” y “**ps3int2**”.

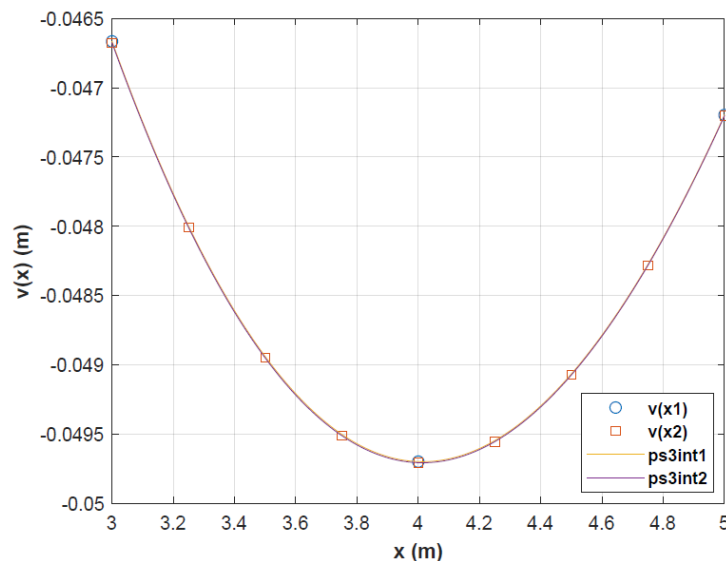


Figura 7.25. Deflexión de la viga biapoyada (**7.26**) en el intervalo $[3,5](m)$ mediante método del disparo lineal. Comparativa entre mallas e interpolaciones con *pseudo-splines* cúbicos.

Los *pseudo-splines* “**ps3int1**” y “**ps3int2**” están prácticamente superpuestos el uno con el otro. Además, ahora ya no se tiene el problema de regularidad surgido del empleo de *splines* cuadráticos, pues la monotonía de las soluciones es mucho más suave y acorde a la realidad física del problema de la **Figura 7.19**. Se elude, de momento, el empleo de valores numéricos en nodos no pertenecientes a las mallas de la **Tablas 7.24**, **7.25** y **7.26**, reservando el estudio a posteriores apartados.

7.5.1.4. Estrategia de interpolación con *pseudo-splines* quínticos.

Finalmente, se opta por seguir el mismo flujo de trabajo que en 7.5.1.1 y 7.5.1.2 y analizar el comportamiento de una nueva estrategia de interpolación de las soluciones discretas obtenidas en las mismas mallas que antes, empleando *pseudo-splines* quínticos, cuyo marco teórico puede ser consultado en la sección 5.3.3 del **Capítulo 5**. En concordancia con los apartados citados, se tiene la **Figura 7.26**, la cual se centra una vez más en el intervalo $[3,5](m)$ del dominio del *PVF* (7.26).

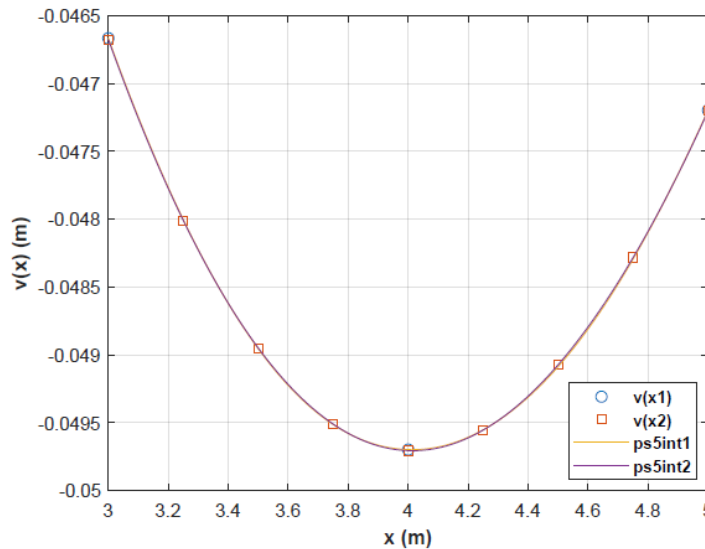


Figura 7.26. Deflexión de la viga biapoyada (7.26) en el intervalo $[3,5](m)$ mediante método del disparo lineal. Comparativa entre mallas e interpolaciones con *pseudo-splines* quínticos.

A la vista de la gráfica anterior, la cual indica una superposición entre trazadores de la misma naturaleza que la comentada en la **Figura 7.25**, se concluye que **el empleo de *pseudo-splines* quínticos interpoladores no resulta en una mejora sustancial de la regularidad del problema en términos exclusivamente cualitativos**, puesto que para conseguirla es suficiente con el empleo de *pseudo-splines* cúbicos. Nótese que los trazadores quínticos obtienen por colocación el valor de la segunda derivada del *PVF* (7.26) para posterior interpolación (en realidad, de sus *PVIs* asociados (7.27) y (7.28) y combinación por disparo lineal).

7.5.1.5. Comparativa entre estrategias de interpolación.

Finalmente, tras haber obtenido con las mallas de la **Tabla 7.24** las soluciones discretas aproximadas del *PVF* (7.26) (**Tablas 7.25 y 7.26**) y sus posteriores soluciones interpoladas con múltiples estrategias (**Tabla 7.27**), se puede realizar una comparativa entre todas ellas (**tanto cualitativa como cuantitativa**), en aquellos intervalos en los que proceda prestar una especial atención. Se considera, por tanto, el entorno de la sección central de la viga, tal y como se ha ido considerando en anteriores apartados. La **Figura 7.27** es una comparativa entre estrategias de interpolación con malla x_1 , y en el intervalo del dominio de (7.26) correspondiente a $[3.6,4.4](m)$.

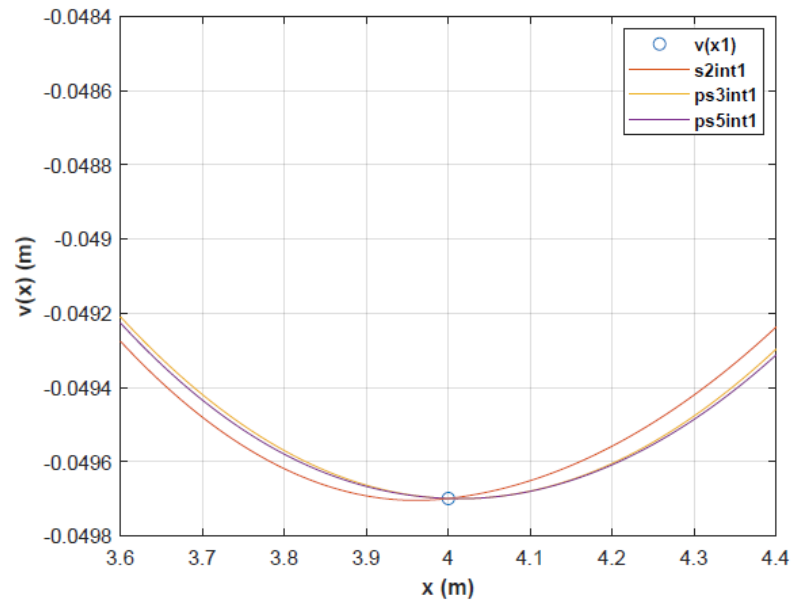


Figura 7.27. Deflexión de la viga biapoyada (7.26) en el intervalo $[3.6,4.4]$ (m) mediante método del disparo lineal. Comparativa entre estrategias de interpolación en malla x_1 .

Cualitativamente, se puede afirmar que se cumple todo lo comentado hasta ahora en los apartados en lo que se estudia cada estrategia individualmente. No obstante, y asumiendo la tendencia discordante de “s2int1” respecto de “ps3int1” y “ps5int1” por lo establecido en 7.5.1.2 (continuidad C^1), es preciso matizar que **también existe diferencia cuantitativa apreciable en las aproximaciones con pseudo-splines**. Para valorar numéricamente este hecho, se consideran los valores de los tres trazadores en dos nodos distintos $t_1 = 3.65$ y $t_2 = 4.35$ (no pertenecientes a la malla inicial x_1). Asumiendo ahora “ps5int1” como la estrategia más exacta, en concordancia con las condiciones impuestas en su marco teórico, se puede evaluar el error cometido por las estrategias restantes. Con todo lo anterior, se tiene la **Tabla 7.28**:

Nodo t	$ps5int1(t)$	$s2int1(t)$	Error “s2int1”	$ps3int1(t)$	Error “ps3int1”
$t_1 = 3.65$	-4.93384E-02	-4.93865E-02	4.81E-05	-4.93222E-02	1.62E-05
$t_2 = 4.35$	-4.94057E-02	-4.93343E-02	7.14E-05	-4.93934E-02	1.23E-05

Tabla 7.28. Valores numéricos nodales de la deflexión de la viga biapoyada (7.26) y error cometido respecto a la estrategia de interpolación con pseudo-splines quinticos en malla x_1 .

La tabla muestra una aproximación mejor en cada uno de los nodos seleccionados y respecto del valor dado por “ps5int1” para “ps3int1”. No obstante, el orden de magnitud del error con respecto al calculado para “s2int1” no disminuye y permanece en 10^{-5} , por lo que el **factor de decisión a la hora de elegir una estrategia u otra es más la regularidad alcanzada por cada trazador que la disminución del error** capaz de producir por una estrategia con condiciones más restrictivas a la solución del PVF (7.26).

Ahora, se puede repetir el mismo razonamiento para la malla x_2 . Por tanto, se tienen la **Figura 7.28** y la **Tabla 7.29**.

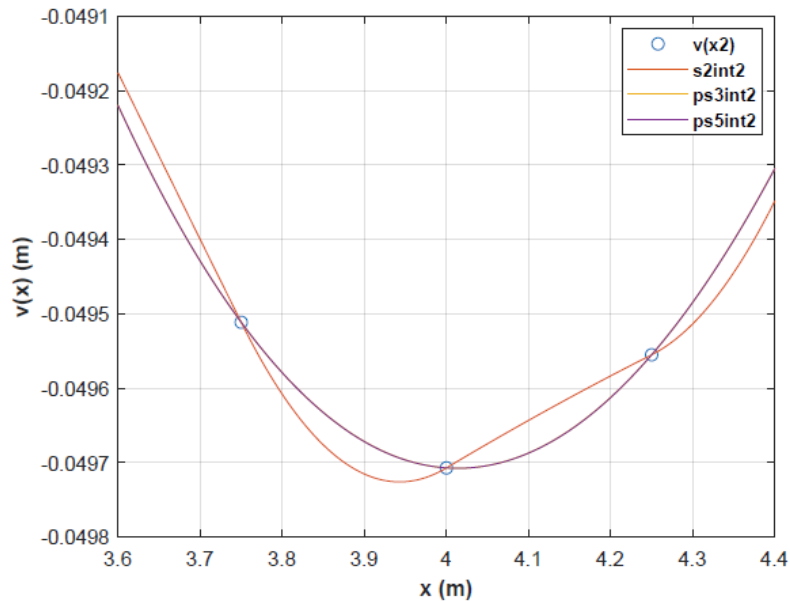


Figura 7.28. Deflexión de la viga biapoyada (7.26) en el intervalo $[3.6,4.4]$ (m) mediante método del disparo lineal. Comparativa entre estrategias de interpolación en malla x_2 .

Nodo t	$ps5int2(t)$	$s2int2(t)$	Error " $s2int2$ "	$ps3int2(t)$	Error " $ps3int2$ "
$t_1 = 3.65$	-4.93320E-02	-4.92881E-02	4.39E-05	-4.93318E-02	2E-07
$t_2 = 4.35$	-4.94012E-02	-4.94446E-02	4.34E-05	-4.94010E-02	2E-07

Tabla 7.29. Valores numéricos nodales de la deflexión de la viga biapoyada (7.26) y error cometido respecto a la estrategia de interpolación con *pseudo-splines* quinticos en malla x_2 .

Se pone de manifiesto una vez más la falta de continuidad en curvatura (ligada a la no imposición de condiciones de continuidad con la segunda derivada del *spline* cuadrático S'') y peor aproximación con malla densificada empleando *splines* cuadráticos respecto a la obtenida con la malla x_1 . En cuanto al comportamiento de la estrategia de interpolación con *pseudo-splines* cúbicos, destaca la **disminución de error en los nodos de ensayo del trazador " $ps3int2$ " respecto al obtenido para la primera malla en la Tabla 7.28** (con un orden de 10^{-7} y empleando 6 cifras significativas), todo ello asumiendo " $ps5int2$ " como solución aproximada más cercana a la exacta desconocida del PVF (7.26).

7.5.2. Conclusiones del experimento.

Las conclusiones finales de este experimento numérico, teniendo en cuenta las observaciones realizadas de todo el material disponible, son las indicadas a continuación:

- La interpolación con trazadores polinomiales (cúbicos (7.5.1.3) o quinticos (7.5.1.4)) de la solución discreta aproximada dada por la combinación de soluciones correspondiente al método del disparo lineal del PVF (7.26) (7.5.1.1), así como de sus derivadas arroja mejores resultados en regularidad de la interpolación y ajuste a la realidad física del problema que un densificado local o global del mallado discreto de partida.

- ii. El empleo de *splines* cuadráticos (7.5.1.2) con continuidad C^1 no resulta ser suficiente para interpolar correctamente la solución discreta de (7.26), debido a las **fluctuaciones y cambios indeseables de curvatura**. No obstante, aún estando su marco teórico fundamentado en el esquema recursivo señalado (sección 4.3 del **Capítulo 4**) como inestable para trazadores cúbicos en el experimento numérico 3 (7.3), es posible utilizarlo satisfactoriamente sin fallos ligados más allá de sus condiciones de continuidad ya que no presenta ningún coeficiente problemático en cada intervalo (a_{k3} no está definido).
- iii. **Nunca ha de optarse por un densificado creciente para obtener mejor precisión tras haber identificado ésta última insuficiente si se dispone de estrategias de interpolación de soluciones**. Este procedimiento ha de entenderse como una **estrategia dual**, la cual finalmente puede conllevar a reducir el coste computacional de la obtención de la solución.

El script de **MATLAB** “*PVF_Viga_Biapoyada*” empleado para la realización del experimento, puede consultarse en el **ANEXO III** de esta memoria.

7.6. Experimento numérico 5. Perfil de temperaturas en una aleta anular hiperbólica.

Las **aletas**, también conocidas como **superficies adicionales**, son prolongaciones utilizadas para **aumentar el flujo de calor** desde la superficie de un sólido hasta un fluido. Si se incrementa la superficie sometida a convección aumenta el flujo de calor. Algunas de las aplicaciones clásicas de las aletas se encuentran en los *radiadores, motores eléctricos, refrigeradores, placas base de ordenadores...* [68]. Su función puede ser de refrigeración para evitar problemas de elevada temperatura (ordenadores, refrigeradores...) o de calentamiento de un determinado entorno (radiadores). Uno de los puntos importantes a la hora de optimizar una aleta es la **distribución de temperaturas existente, para nada uniforme y altamente dependiente de la geometría de diseño**. La ecuación general de las aletas toma la forma de (7.29), y su demostración puede consultarse en [68].

$$A'(r) \cdot T'(r) + A(r) \cdot T''(r) = \frac{h}{k} \cdot S'(r) \cdot (T - T_{\infty})$$

Ecuación (7.29)

Se trata de una **EDO de 2º orden lineal de coeficientes variables** para la distribución de temperaturas $T(r)$ basada en las **Hipótesis de Gardner** [69], donde destaca el asumir un campo de temperaturas unidireccional estacionario. Se ha adoptado la notación de *Lagrange* al tratarse de un problema dependiente de la posición (2.1.1). El significado físico de los coeficientes y parámetros de (7.29) es el mostrado en la **Tabla 7.30**.

<i>Magnitud</i>	<i>Descripción</i>
$A(r)$	Superficie de tránsito del flujo de calor por conducción
$S'(r)$	Variación de la superficie lateral de disipación convectiva al medio fluido
h	Coefficiente de película convectivo
k	Conductividad térmica del material sólido
T_∞	Temperatura del fluido que baña la aleta

Tabla 7.30. Significado físico de los términos de la ecuación general de las aletas.

Haciendo uso de las hipótesis presentadas y teniendo en cuenta el significado físico e ingenieril del fenómeno se aborda la **distribución radial de temperaturas adimensionalizada para una aleta anular hiperbólica**. Este caso corresponde a aletas colocadas alrededor de una tubería por la que circula un fluido que disipa calor.

La aleta anular que se pretende estudiar mediante la resolución de (7.30) presenta la geometría dada por la **Figura 7.29**, en donde se adopta un **sistema de referencia en coordenadas cilíndricas**, con el que la aleta presenta simetría respecto a la coordenada radial r (vista isométrica en corte).

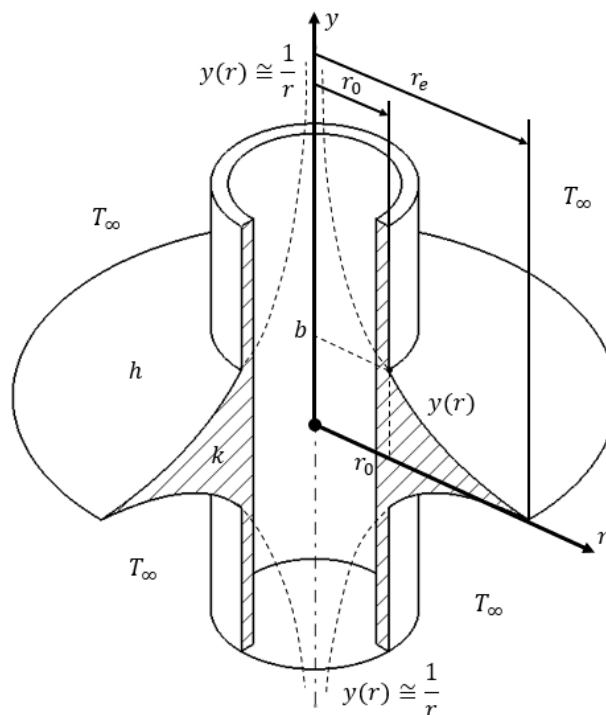


Figura 7.29. Detalle en corte del tubo rodeado por la aleta anular hiperbólica.

El contorno $y(r)$ puede aproximarse a la **hipérbola unidad, despreciando también el espesor de la aleta en su extremo**. En este apartado, la obtención de la ecuación (7.29) particularizada al caso de estudio se omite, pues existen referencias que la proponen directamente ([70] y [71]). La expresión resultante

es la dada por (7.30), en donde se adoptan variables adimensionales de radio R y temperatura θ :

$$\theta''(R) = M^2 \cdot R \cdot \theta(R)$$

Ecuación (7.30)

Se definen θ , R , c y M (número adimensional de Biot) en (7.31):

$$\theta(R) = \frac{T(R) - T_\infty}{T_0 - T_\infty} \quad \text{a)}$$

$$R = \frac{r}{r_e}, \quad c = \frac{r_0}{r_e} \quad \text{b)}$$

$$M = \sqrt{\frac{hr_e^3}{kbr_0}} \quad \text{c)}$$

Ecuación (7.31)

El valor T_0 corresponde a la temperatura en la base de la aleta ($r = r_0$). Si se utiliza (7.30) en combinación con condiciones de contorno debidamente impuestas, surge (7.32):

$$\begin{cases} \theta''(R) = M^2 \cdot R \cdot \theta(R) \\ \theta\left(\frac{r_0}{r_e}\right) = 1, \quad \theta'(1) = 0 \\ R \in \left[\frac{r_0}{r_e}, 1\right] \end{cases}$$

Ecuación (7.32)

La segunda condición de tiene su razón de ser en considerar el **flujo de calor despreciable en el extremo de la aleta**. Se ha obtenido un **Problema de Valores en la Frontera del tipo (6.5)**, con condiciones de contorno generales y formulado en términos de una EDO de 2º orden lineal con coeficientes variables en el intervalo $[0, 1]$ e independiente de la primera derivada. Para este caso particular, existe solución analítica (7.33) (también propuesta en [70] y [71]), la cual se construye mediante funciones de Bessel modificadas de primera especie I_α .

$$\theta(R) = \sqrt{\frac{R}{c}} \cdot \left[\frac{I_{1/3}\left(\frac{2}{3}MR^{3/2}\right)I_{2/3}\left(\frac{2}{3}M\right) - I_{-1/3}\left(\frac{2}{3}MR^{3/2}\right)I_{-2/3}\left(\frac{2}{3}M\right)}{I_{1/3}\left(\frac{2}{3}Mc^{3/2}\right)I_{2/3}\left(\frac{2}{3}M\right) - I_{-1/3}\left(\frac{2}{3}Mc^{3/2}\right)I_{-2/3}\left(\frac{2}{3}M\right)} \right]$$

Ecuación (7.33)

En la obtención de un problema con valores numéricos, pueden considerarse los datos por la **Tabla 7.31**, en unidades del *Sistema Internacional*.

<i>Magnitud</i>	<i>Descripción</i>	<i>Valor</i>
r_0	Radio base de la aleta	0.015 m
r_e	Radio extremo de la aleta	0.05 m
b	Altura mitad base de la aleta	$7 \cdot 10^{-3}$ m
k	Conductividad térmica (acero)	$50.2 \frac{W}{m \cdot K}$
h	Coefficiente de película convectivo (convección forzada)	$200 \frac{W}{m^2 \cdot K}$

Tabla 7.31. Valores numéricos propuestos para la aleta anular.

Sustituyendo en (7.30), (7.31) y (7.32), se obtiene finalmente (7.34), con solución analítica (7.35).

$$\begin{cases} \theta''(R) = \frac{25000}{5271} \cdot R \cdot \theta(R) \\ \theta(0.3) = 1, \quad \theta'(R) = 0 \\ R \in [0.3, 1] \end{cases}$$

Ecuación (7.34)

$$\theta(R) = \sqrt{\frac{10R}{3}} \cdot \left[\frac{I_{1/3} \left(\frac{100}{3} \sqrt{\frac{10}{5271}} R^{3/2} \right) I_{2/3} \left(\frac{100}{3} \sqrt{\frac{10}{5271}} \right) - I_{-1/3} \left(\frac{100}{3} \sqrt{\frac{10}{5271}} R^{3/2} \right) I_{-2/3} \left(\frac{100}{3} \sqrt{\frac{10}{5271}} \right)}{I_{1/3} \left(\sqrt{\frac{300}{5271}} \right) I_{2/3} \left(\frac{100}{3} \sqrt{\frac{10}{5271}} \right) - I_{-1/3} \left(\sqrt{\frac{300}{5271}} \right) I_{-2/3} \left(\frac{100}{3} \sqrt{\frac{10}{5271}} \right)} \right]$$

Ecuación (7.35)

7.6.1. Solución numérica. Aplicación del método del disparo lineal y análisis del error.

Este problema, al igual que el planteado en el experimento numérico 4 (7.5), debe ser resuelto mediante alguna técnica de resolución de *PVFs*. Dado que este trabajo utiliza el método del disparo lineal (**Capítulo 6**), se fija dicha técnica como la empleada para obtener su solución, junto con **estrategias de interpolación centradas exclusivamente en pseudo-splines cúbicos basados en polinomios interpoladores de Hermite** (marco teórico en 5.3.2 del **Capítulo 5**). Además, también interesa conocer el **comportamiento cualitativo y cuantitativo del error de interpolación**, para lo cual se dispone de la solución exacta expuesta en (7.35), permitiendo ésta última obtener conclusiones.

Para dar comienzo al estudio, al igual que se ha hecho en anteriores experimentos, considérense las siguientes mallas de nodos que discretizan el dominio del *PVF* (7.34) según lo estipulado en (2.11) y la **Tabla 7.32**:

<i>Malla</i>	<i>Alias</i>	<i>Subintervalos</i>	<i>Paso uniforme h</i>
1	R_1	$n_1 = 5$	$h_1 = 0.14$
2	R_2	$n_2 = 2$	$h_2 = 0.35$
3	R_3	$1 \rightarrow [0.3, 0.8)$	$h_{31} = 0.25$
		$2 \rightarrow [0.8, 1]$	$h_{32} = 0.05$

Tabla 7.32. Mallas para abordar la resolución del PVF (7.34).

Las dos primeras mallas son equiespaciadas, pretendiendo determinar la solución sin ningún densificado local de nodos para posterior interpolación. En cuanto a la tercera, el densificado en el entorno del nodo extremo $R = 1$ por la izquierda responde a la intención de observar con mayor nitidez el cumplimiento de la segunda condición de contorno de (7.34), la cual se formula en términos de la primera derivada θ' , recordando que su significado físico corresponde a un flujo de calor despreciable en el extremo de la aleta de perfil hiperbólico.

Los tres PVI's en los que el PVF de estudio ha de desdoblarse para poder aplicar el método de disparo lineal posteriormente corresponden a (7.36), (7.37) y (7.38) (todos ellos homogéneos para este problema particular), en concordancia con lo expuesto en la sección 6.4.2 del Capítulo 6.

$$\begin{cases} \theta_1''(R) = \frac{25000}{5271} \cdot R \cdot \theta_1(R) \\ \theta_1(0.3) = 0, \quad \theta_1'(0.3) = 0 \\ R \in [0.3, 1] \end{cases}$$

Ecuación (7.36)

$$\begin{cases} \theta_2''(R) = \frac{25000}{5271} \cdot R \cdot \theta_2(R) \\ \theta_2(0.3) = 0, \quad \theta_2'(0.3) = 1 \\ R \in [0.3, 1] \end{cases}$$

Ecuación (7.37)

$$\begin{cases} \theta_3''(R) = \frac{25000}{5271} \cdot R \cdot \theta_3(R) \\ \theta_3(0.3) = 1, \quad \theta_3'(0.3) = 0 \\ R \in [0.3, 1] \end{cases}$$

Ecuación (7.38)

Es muy fácil apreciar que el primer PVI tiene solución nula, pues es el único caso físicamente posible al considerar temperatura adimensional y flujo de calor nulo en la base de la aleta de la Figura 7.29, lo que corresponde a una temperatura igual que la del fluido ambiente T_∞ sin que exista flujo por gradiente nulo. Se centra por tanto el estudio en conocer las soluciones de los



PVIs (7.37) y (7.38), para posteriormente combinarlas y obtener la del *PVF* (7.34). Los coeficientes de la combinación lineal de soluciones (6.36) vienen dados por (7.39), siendo estos últimos particularizaciones de (6.39):

$$d_2 = \boxed{-1.3471}, \quad d_3 = \boxed{1}$$

Ecuación (7.39)

7.6.1.1. Obtención de la solución discreta aproximada del *PVF* para cada malla de ensayo.

Ahora, los valores numéricos aproximados de las soluciones de cada *PVI*, así como de la solución del *PVF* de partida obtenidos por disparo lineal se indican, para cada malla de la *Tabla 7.32*, en las *Tablas 7.33*, *7.34* y *7.35*. Para obtener las soluciones de (7.37) y (7.38), se emplea el método *Runge-Kutta-Nyström* simple de orden 5 para problemas independientes de la primera derivada, con alias en *MATLAB* “*RKN51_no_dy*” y expuesto en la sección 3.3.5.3 del *Capítulo 3* (disponible también para su uso en el *ANEXO I*). Se omite el *PVI* (7.36), por las consideraciones ya expuestas anteriormente sobre sus soluciones nulas.

<i>Malla R₁</i>	<i>PVI(7.37):θ₂₁</i>	<i>PVI(7.38):θ₃₁</i>	<i>PVF(7.34):θ_{R₁}</i>
0.30	0.000000	1.000000	1.000000
0.44	0.140804	1.016159	0.826483
0.58	0.287697	1.074101	0.686546
0.72	0.450507	1.190401	0.583529
0.86	0.644021	1.387158	0.519604
1.00	0.889839	1.696096	0.497403

Tabla 7.33. Valores numéricos de las soluciones discretas aproximadas en la malla *R₁* de los *PVIs* (7.37) y (7.38) posteriormente combinadas por disparo lineal para obtener la solución del *PVF* (7.34).

<i>Malla R₂</i>	<i>PVI(7.37):θ₂₂</i>	<i>PVI(7.38):θ₃₂</i>	<i>PVF(7.34):θ_{R₂}</i>
0.30	0.000000	1.000000	1.000000
0.65	0.366293	1.12373	0.630295
1.00	0.889740	1.695973	0.497399

Tabla 7.34. Valores numéricos de las soluciones discretas aproximadas en la malla *R₂* de los *PVIs* (7.37) y (7.38) posteriormente combinadas por disparo lineal para obtener la solución del *PVF* (7.34).

<i>Malla R₃</i>	<i>PVI(7.37):θ₂₃</i>	<i>PVI(7.38):θ₃₃</i>	<i>PVF(7.34):θ_{R₃}</i>
0.30	0.000000	1.000000	1.000000
0.55	0.255279	1.057397	0.713513
0.80	0.556119	1.291221	0.542078
0.85	0.628761	1.369825	0.522828
0.90	0.707752	1.462256	0.508851
0.95	0.794310	1.570316	0.500308
1.00	0.889834	1.696092	0.497406

Tabla 7.35. Valores numéricos de las soluciones discretas aproximadas en la malla *R₃* de los *PVIs* (7.37) y (7.38) posteriormente combinadas por disparo lineal para obtener la solución del *PVF* (7.34).

Se ha considerado idóneo emplear aritmética de 6 cifras decimales significativas para presentar los resultados, ya que así es posible observar la diferencia entre unas soluciones y otras al aplicar el mismo método numérico para obtenerlas. Los valores de las tablas anteriores son la base para interpolar las soluciones de (7.34), ya que posteriormente a su determinación unívoca los trazadores polinomiales que interpolan soluciones de los *PVIs* (7.37) y (7.38) se combinan utilizando la ecuación (6.40).

7.6.1.2. |EGT| cometido en la determinación de la solución discreta del PVF y su primera derivada.

Disponer de la solución del *PVF* dada por (7.35) es una importante ventaja, pues permite observar el **módulo del EGT** cometido por el método “*RKN51_no_dy*”, una vez que han sido combinadas las soluciones de los *PVIs* por disparo lineal. Utilizando los datos de las últimas columnas de las *Tablas* 7.33, 7.34 y 7.35, se puede obtener dicho error según la ecuación (3.32). Dado que el problema a resolver presenta una condición de contorno en términos de la primera derivada en su nodo extremo final $R = 1$, **se fija el interés por conocer cómo de precisa es la aproximación que el método del disparo es capaz de aportar al valor nulo de la derivada en dicho nodo**, función lógicamente de las mallas de la *Tabla* 7.32 y el método numérico aplicado para cada *PVI*.

Para poder computar el |EGT| correspondiente a la primera derivada (definido en (3.33)), ha de obtenerse $\theta'(R)$. Su expresión se indica en (7.40).

$$\begin{aligned} \theta'(R) = & \frac{\sqrt{cR}}{2AcR} \cdot \left[I_{1/3} \left(\frac{2}{3} MR^{3/2} \right) I_{2/3} \left(\frac{2}{3} M \right) - I_{-1/3} \left(\frac{2}{3} MR^{3/2} \right) I_{-2/3} \left(\frac{2}{3} M \right) \right] + \\ & + \frac{MR}{2A\sqrt{c}} \cdot \left[I_{2/3} \left(\frac{2}{3} M \right) \left(I_{-2/3} \left(\frac{2}{3} MR^{3/2} \right) + I_{4/3} \left(\frac{2}{3} MR^{3/2} \right) \right) \right. \\ & \quad \left. - I_{-2/3} \left(\frac{2}{3} M \right) \left(I_{-4/3} \left(\frac{2}{3} MR^{3/2} \right) + I_{2/3} \left(\frac{2}{3} MR^{3/2} \right) \right) \right] \\ A = & I_{1/3} \left(\frac{2}{3} Mc^{3/2} \right) I_{2/3} \left(\frac{2}{3} M \right) - I_{-1/3} \left(\frac{2}{3} Mc^{3/2} \right) I_{-2/3} \left(\frac{2}{3} M \right) \end{aligned}$$

Ecuación (7.40)

Se ha empleado la derivación de funciones de *Bessel* modificadas de primera especie con orden α I_α definida por la fórmula de recurrencia (7.41), con posterior aplicación de la regla de la cadena.

$$I'_\alpha(z) = \frac{1}{2} (I_{\alpha-1}(z) - I_{\alpha+1}(z))$$

Ecuación (7.41)

Derivando (6.40) se obtiene (7.42), expresión empleada para combinar las derivadas discretas aproximadas obtenidas por el método *RKN* para los *PVIs* (7.37) y (7.38), en donde también se ha anulado ya la solución de (7.36).

$$\theta'(R) = d_2\theta'_2(R) + d_3\theta'_3(R)$$

Ecuación (7.42)

Utilizando las expresiones anteriores y los datos disponibles calculados en la sección 7.6.1.1, se calculan los $|EGT|$ para cada malla de la *Tabla 7.32*. Estos errores vienen dados en las *Tablas 7.36*, *7.37* y *7.38*, junto con los valores de la primera derivada de la solución discreta del *PVF* (7.34) combinados por disparo lineal, empleando notación científica con 3 cifras significativas.

<i>Malla</i> R_1	θ'_{R_1}	$ EGT _{\theta}$	$ EGT _{\theta'}$
0.30	-1.35E+00	0.00E+00	1.72E-07
0.44	-1.13E+00	5.54E-08	3.34E-07
0.58	-8.71E-01	7.51E-08	3.90E-07
0.72	-5.99E-01	4.91E-08	3.42E-07
0.86	-3.12E-01	3.52E-08	2.03E-07
1.00	-4.78E-16	2.01E-07	4.78E-16

Tabla 7.36. $|EGTs|$ proporcionados por el método “*RKN51_no_dy*” aplicado a la resolución del *PVF* (7.34) por disparo lineal en la malla R_1 .

<i>Malla</i> R_2	θ'_{R_2}	$ EGT _{\theta}$	$ EGT _{\theta'}$
0.30	-1.35E+00	0.00E+00	1.75E-05
0.65	-7.36E-01	1.66E-05	4.40E-05
1.00	-4.78E-16	4.43E-06	4.78E-16

Tabla 7.37. $|EGTs|$ proporcionados por el método “*RKN51_no_dy*” aplicado a la resolución del *PVF* (7.34) por disparo lineal en la malla R_2 .

<i>Malla</i> R_3	θ'_{R_3}	$ EGT _{\theta}$	$ EGT _{\theta'}$
0.30	-1.35E+00	0.00E+00	7.87E-07
0.55	-9.27E-01	2.68E-06	4.99E-06
0.80	-4.37E-01	2.60E-06	2.10E-06
0.85	-3.33E-01	2.51E-06	1.60E-06
0.90	-2.26E-01	2.44E-06	1.08E-06
0.95	-1.15E-01	2.40E-06	5.53E-07
1.00	-4.78E-16	2.38E-06	4.78E-16

Tabla 7.38. $|EGTs|$ proporcionados por el método “*RKN51_no_dy*” aplicado a la resolución del *PVF* (7.34) por disparo lineal en la malla R_3 .

Se concluye que, a la vista de los resultados obtenidos para el $|EGT|$ de la derivada en el último nodo, la condición de contorno se satisface perfectamente tras haber aplicado el método del disparo independientemente de cuál sea la malla empleada, pues se obtienen errores ridículos del orden de 10^{-16} . *MATLAB* opera en aritmética de coma flotante con 16 cifras significativas, por lo que a todos los efectos se tiene el valor nulo precisado por (7.34).

En relación al módulo del *EGT* de la solución discreta, a continuación se llevan a cabo estudios que lo involucran, junto con el error de interpolación asociado a los *pseudo-splines* cúbicos fijados como trazadores a emplear en este experimento.

7.6.1.3. Interpolación de la solución discreta mediante el empleo de *pseudo-splines* cúbicos. Estudio del error.

Si ahora se aprovecha toda la información de la solución aproximada del *PVF* (7.34) disponible en las tablas de las secciones 7.6.1.1 y 7.6.1.2, se puede llevar a cabo la interpolación de las soluciones mediante *pseudo-splines* cúbicos. Interesa en este apartado conocer el error cometido a la hora de interpolar las soluciones de (7.34) mediante la combinación de trazadores definida en (6.40).

Antes de proceder, se aporta la *Figura 7.30*, en la cual se indica la forma de la solución exacta (7.35), evaluada en un número suficiente de puntos y con los nodos de las mallas de la *Tabla 7.32* señalados.

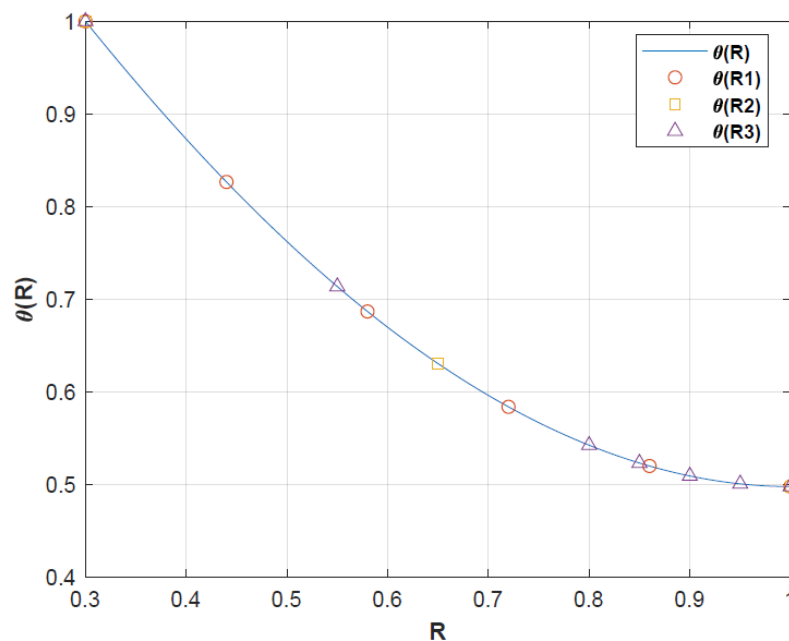


Figura 7.30. Forma de la solución exacta del *PVF* (7.34) dada por (7.35).

Efectivamente, la distribución adimensional de temperaturas $\theta(R)$ cae desde la base de la aleta hasta su extremo atenuándose progresivamente el flujo de calor (representado por el valor decreciente de $\theta'(R)$, con valor final nulo).

Ahora, se procede a interpolar la solución discreta aproximada en la malla R_1 dada por la *Tabla 7.33*, interpolando previamente las soluciones de los *PVIs* (7.37) y (7.38). Se obtiene por aplicación del método del disparo lineal y la ecuación (6.40) el *pseudo-spline* cúbico "*ps3int1*", cuya forma viene dada por la *Figura 7.31*:

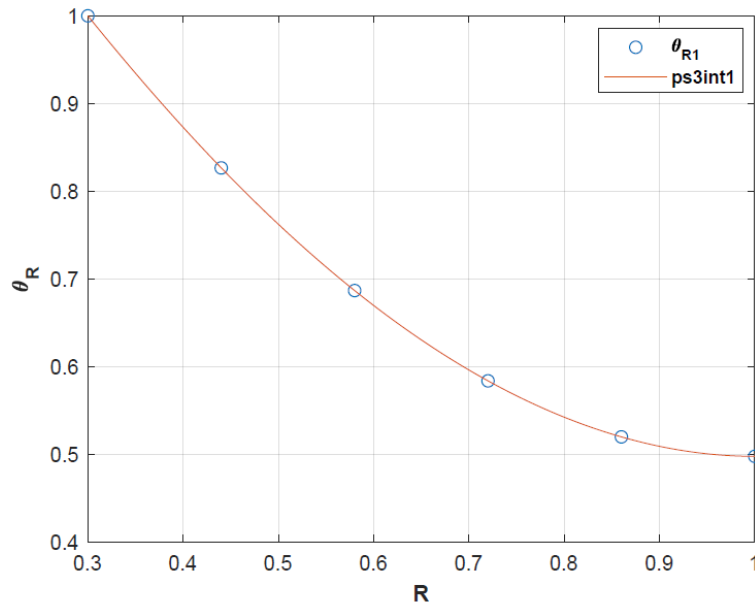


Figura 7.31. Solución aproximada del *PVF* (7.34) interpolada con un *pseudo-spline* cúbico en los nodos de la malla R_1 .

Empleando los datos de la *Tabla 7.36* y la solución exacta de la *Figura 7.30*, se puede graficar el error cometido en la interpolación, teniendo en cuenta también el $|EGT|$ que proporciona el método *RKN* elegido para obtener en la malla R_1 las soluciones de los *PVs* en los que el *PVF* se desdobra. Por tanto, se tiene la *Figura 7.32*:

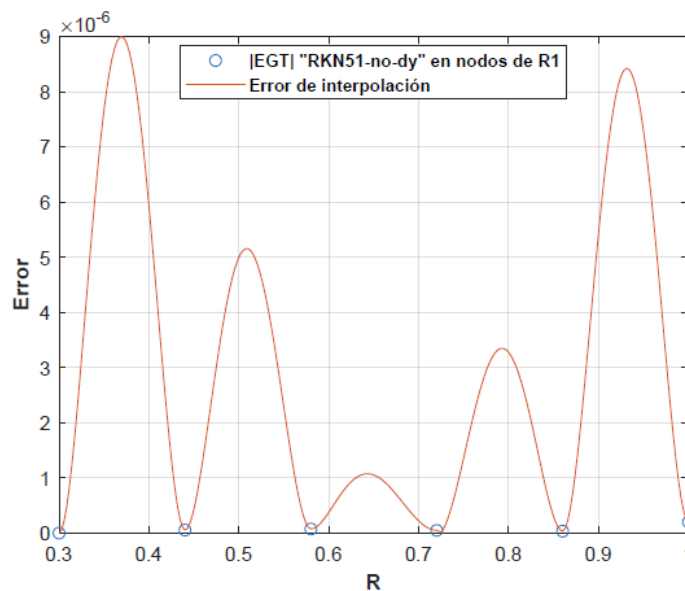


Figura 7.32. Gráfica del error cometido respecto de (7.35) al interpolar la solución discreta aproximada en la malla R_1 con “*ps3int1*”.

Se observa un crecimiento del error de interpolación significativo en el entorno de los nodos intermedios de los intervalos generados tras discretizar el dominio de (7.34) mediante R_1 . **No obstante, la estrategia de interpolación es estable,**

pues dicho error no se propaga, manteniéndose acotado superiormente en 10^{-6} con incluso disminución en intervalos lejanos a los extremos del problema.

Se puede repetir el procedimiento anterior empleando la malla equiespaciada y menos densa R_2 . El *pseudo-spline* cúbico interpolador ahora es “*ps3int2*” y los resultados del procedimiento se indican en las **Figuras 7.33** (solución interpolada) y **7.34** (comportamiento del error).

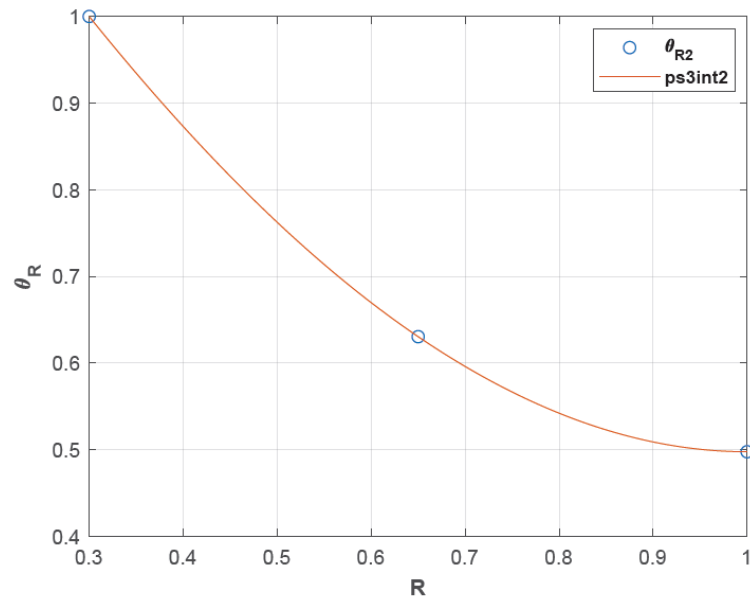


Figura 7.33. Solución aproximada del PVF (7.34) interpolada con un *pseudo-spline* cúbico en los nodos de la malla R_2 .

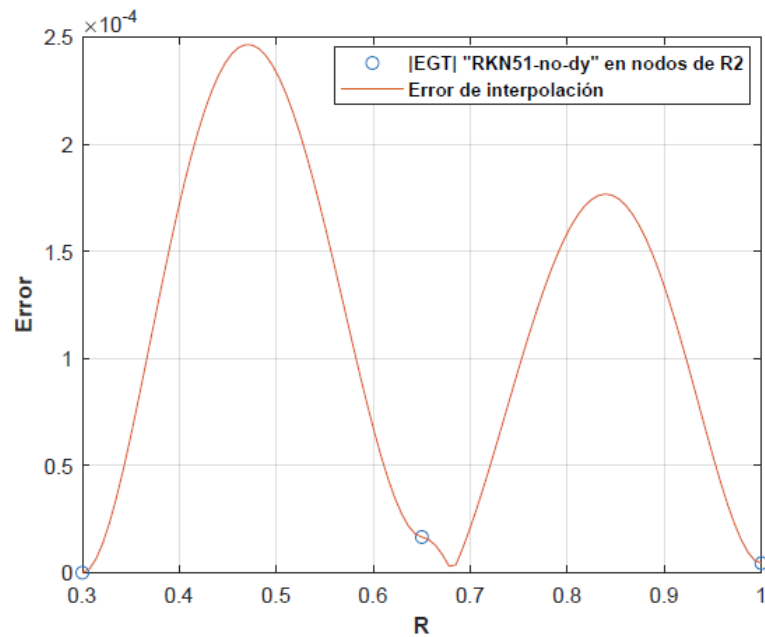


Figura 7.34. Gráfica del error cometido respecto de (7.35) al interpolar la solución discreta aproximada en la malla R_2 con “*ps3int2*”.

Los resultados obtenidos son los esperados, pues el error ahora aumenta (máximos del orden de 10^{-4}) respecto al obtenido en la **Figura 7.32** para la malla R_1 , ya que se está empleando un paso mayor ($h_2 = 0.35$) dentro de lo razonable para el problema. No obstante, **el comportamiento cualitativo del error es similar al anterior**, ya que se alcanzan máximos locales en los entornos de los nodos intermedios correspondientes a cada subintervalo de la discretización del **PVF (7.34)**. Una vez más, se demuestra que la estrategia de interpolación con **pseudo-splines** cúbicos es estable, independizando su comportamiento en este sentido de los nodos empleados para interpolar.

Finalmente, los resultados para la malla R_3 se indican en las **Figuras 7.35** y **7.36**:

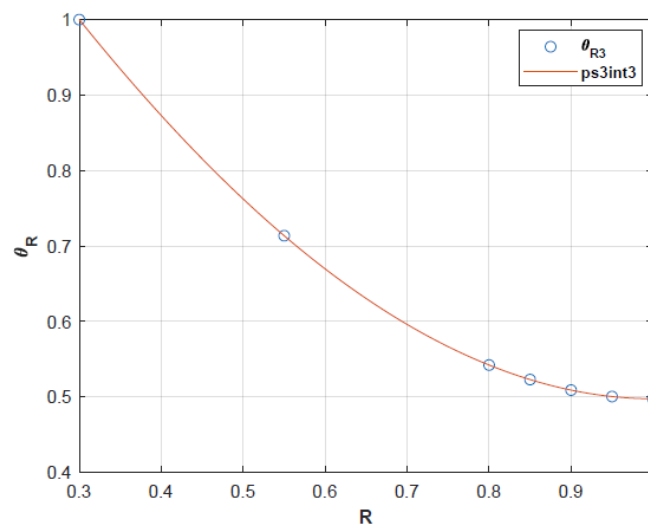


Figura 7.35. Solución aproximada del **PVF (7.34)** interpolada con un **pseudo-spline** cúbico en los nodos de la malla R_3 .

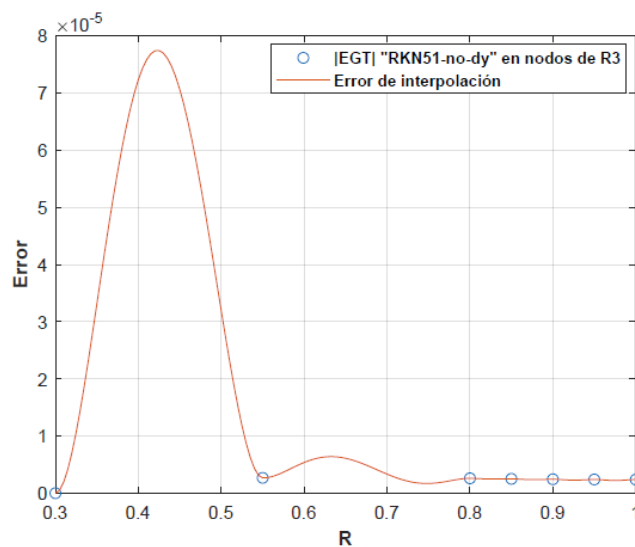


Figura 7.36. Gráfica del error cometido respecto de **(7.35)** al interpolar la solución discreta aproximada en la malla R_3 con "ps3int3".

El densificado en el entorno del nodo extremo del problema $R = 1$ provoca una mejor aproximación (valores discretos de la solución y de su primera derivada disponibles en las **Tablas 7.35** y **7.38**, junto con los **|EGT|** correspondientes). El error de interpolación decae en orden de magnitud cuando $h = h_{32} = 0.05$ (intervalo $[0.8,1]$) respecto del obtenido para $h = h_{31} = 0.25$ (intervalo $[0.3,0.8]$) y valores máximos del orden de 10^{-5}). **Para observar mejor el comportamiento de dicho error al disminuir el paso, considérese el zoom de la Figura 7.36 dado por la Figura 7.37, en el intervalo $[0.7, 1]$.**

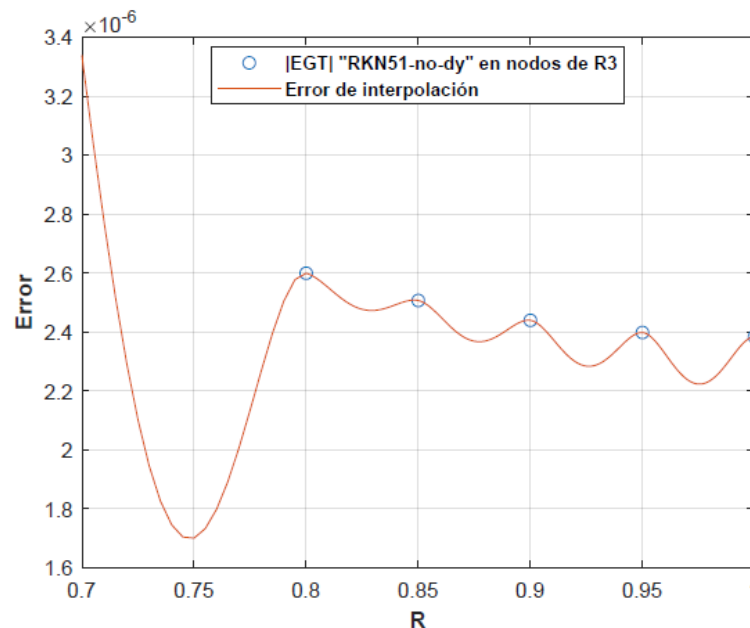


Figura 7.37. Zoom de la **Figura 7.34** en el intervalo $[0.7,1]$.

Efectivamente, la estabilidad de la estrategia provoca una disminución del error de interpolación ligada al densificado local, además de verificar una vez más que **los errores en intervalos anteriores y adyacentes no repercuten en aquellos de intervalos posteriores al no emplear esquemas recursivos problemáticos por su inestabilidad**, en contraposición con las estrategias de interpolación basadas en *splines* cúbicos y *pseudo-splines* cuárticos ya abordadas en **7.4** (marco teórico en **Capítulos 4** y **5**, respectivamente).

7.6.2. Conclusiones del experimento.

Las conclusiones generales de este experimento, en concordancia con las ya aportadas conforme se han ido indicando los resultados numéricos y gráficos se resumen a continuación:

- i. El abordaje de un **PVF** lineal de 2º orden con condiciones de contorno generales resulta ser más complejo en términos algorítmicos que el problema con condiciones de contorno simples. No obstante, y aprovechando la realidad física que presenta el caso particular (**7.34**), se puede simplificar la resolución conociendo que uno de los **PVIs** en los que hay que desdoblar el **PVF** tiene solución nula (**7.36**), evitando por



tanto tener que resolver hasta tres problemas simultáneamente y combinar sus soluciones al final.

- ii. Las estrategias de interpolación basadas en *pseudo-splines* cúbicos y quínticos (secciones 5.3.2 y 5.3.3 del **Capítulo 5**, respectivamente) **son estables y convergen hacia la solución del problema al que son aplicadas cuando disminuye progresivamente el paso empleado para calcular la solución discreta en los nodos de la malla (2.11)**. Esto es debido a que **el error de interpolación no se propaga intervalo a intervalo al no emplear un esquema recursivo para el cálculo unívoco del trazador polinomial**. A pesar de haber empleado tan sólo *pseudo-splines* cúbicos en este problema, se puede afirmar lo citado para los quínticos por construirse de forma similar.
- iii. La única propagación del error entre intervalos en la resolución de (7.34) tiene lugar al aplicar el método numérico “*RKN51_no_dy*” (o cualquiera que fuese adecuado del **Capítulo 3**), ya que al ser de un paso obtiene las aproximaciones mediante la información disponible y calculada previamente en el nodo inmediatamente anterior. No obstante, esto no ha de repercutir en el comportamiento del error exclusivamente de interpolación, pues ya se ha comentado que es intrínseco a cada intervalo de la discretización.

El código de **MATLAB** “*PVF_Aleta_Anular_Hiperbolica*” empleado para la realización de este experimento numérico puede consultarse en el **ANEXO IV**.

7.7. Experimento numérico 6. Respuesta temporal forzada de un péndulo no lineal.

Habiendo ya realizado y concluido un buen número de experimentos numéricos, **se propone llevar a cabo en esta sección la determinación de la respuesta temporal correspondiente a un péndulo no lineal**.

Como ya se indicó en 7.1, las **Vibraciones Mecánicas** son un campo de estudio de enorme interés en Ingeniería del que nacen multitud de modelos físicos basados mayoritariamente en ecuaciones diferenciales. En la introducción de dicho primer experimento se presentaba el problema clásico basado en un sistema “*mkc*” (**Figura 7.1**), formulado en términos de una **EDO** de 2^º lineal con coeficientes constantes, la cual al imponer condiciones iniciales desembocaba en el **PVI (7.6)**, con solución analítica (7.7).

No obstante, **la restricción principal de los modelos vibratorios lineales viene impuesta por la hipótesis de pequeños desplazamientos**, algo que no ha sido comentado con anterioridad. Si no se adopta esta hipótesis, bien sea por trabajar de forma genérica o porque se demuestra que el problema a abordar no la satisface al existir amplitudes y desplazamientos grandes, surgen **modelos vibratorios mayoritariamente no lineales [59]**. Las ecuaciones suelen

ser más complejas y en su mayoría carecen de solución analítica sin posibilidad de aplicar el principio de superposición, por lo que se debe recurrir a métodos numéricos para resolverlas.

En concreto, las oscilaciones que tienen lugar en un péndulo simple sometido a una excitación armónica son un ejemplo muy clásico e ilustrativo, razón por la cual se ha decidido llevar a cabo este experimento. Considérese la **Figura 7.38** como esquema del péndulo a estudiar.

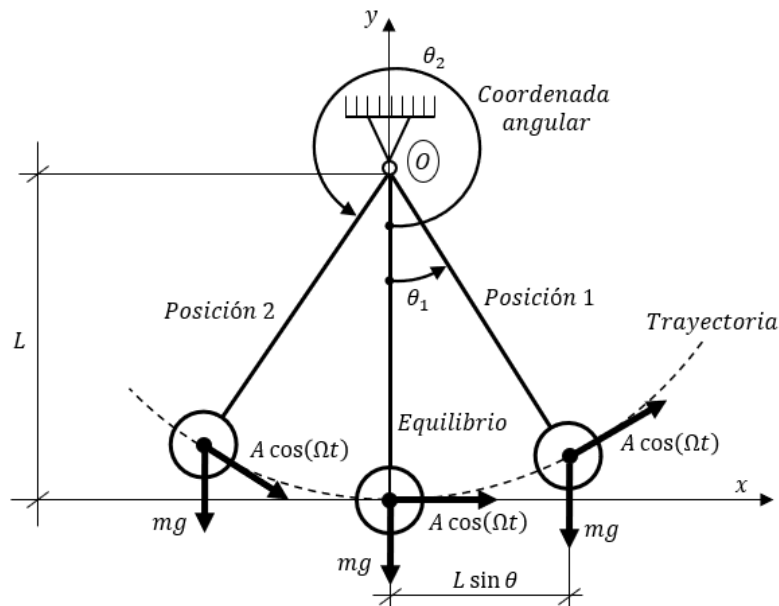


Figura 7.38. Esquema del péndulo no lineal del que se desea obtener su respuesta temporal.

Se representa el oscilador en distintas posiciones angulares θ , teniendo en cuenta que está unido a la bancada o suelo de forma articulada en O , por lo que la posición de equilibrio en el eje y del sistema coordenado corresponde al caso de no existir ninguna sollicitación salvo el peso propio de la masa concentrada m en su extremo dado por mg (varilla rígida de longitud L y masa despreciable). Si se introduce una excitación externa armónica $A \cos(\Omega t)$ a frecuencia Ω a lo largo de la trayectoria circular que la masa concentrada en el extremo de la varilla recorre al perder el equilibrio vertical, se obtiene un sistema que oscila permanentemente (en ausencia de amortiguamiento) y que toma sucesivas posiciones como las marcadas en la **Figura 7.38**, definido el sentido positivo del movimiento oscilatorio como antihorario.

El tratamiento matemático del péndulo se realiza mediante la ecuación que toma la forma dada por (7.43):

$$\ddot{\theta}(t) = -\frac{g}{L} \sin(\theta(t)) + A \cos(\Omega t)$$

Ecuación (7.43)

Es importante denotar que, a la hora de estudiar la respuesta del sistema, ésta no ha de ser necesariamente armónica al existir no linealidades, por lo que la frecuencia no corresponde a la de excitación Ω .

Se adopta el tiempo como variable independiente para estudiar la respuesta oscilatoria regida por la evolución de la coordenada angular θ . Además, y en concordancia con lo expuesto en 2.1.1, se utiliza notación de Newton para las derivadas. Si las amplitudes de las oscilaciones fuesen pequeñas, se podría aproximar el $\sin \theta \cong \theta$ y obtener una EDO lineal muy similar a (7.3) pero sin amortiguamiento (no se considera en este problema). Para obtener un problema no lineal a todos los efectos y con grandes desplazamientos sin una respuesta temporal armónica, considérese que el instante $t = 0$ en el que se inician las oscilaciones corresponde a la **Figura 7.39**:

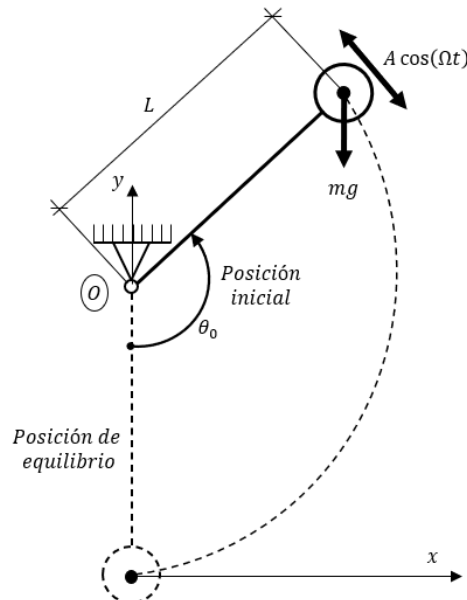


Figura 7.39. Instante $t = 0$ en el que comienzan las oscilaciones del péndulo.

Asumiendo que θ_0 es lo suficientemente grande como para no poder hacer la aproximación lineal, e imponiendo otra condición inicial en términos de la primera derivada $\dot{\theta}_0$ surge el PVI de 2º orden (7.44), el cual es del tipo (2.6) y está formulado en términos de (7.43).

$$\begin{cases} \ddot{\theta}(t) = -\frac{g}{L} \sin(\theta(t)) + A \cos(\Omega t) \\ \theta(0) = \theta_0, \quad \dot{\theta}(0) = \dot{\theta}_0 \\ t \in [0, t_f] \end{cases}$$

Ecuación (7.44)

Para obtener un problema con valores numéricos particularizados, se pueden considerar los valores de la **Tabla 7.39**, en donde también se definen

brevemente todas las magnitudes y condiciones que intervienen en el *PVI* anterior. Por comodidad, se elige trabajar con unidades del *Sistema Internacional*.

<i>Magnitud</i>	<i>Descripción</i>	<i>Valor</i>
L	Longitud de la varilla rígida	1 m
Ω	Frecuencia de excitación	2 rad/s
g	Aceleración de la gravedad	9.8 m/s^2
A	Amplitud de la excitación	0.7
t_f	Instante de tiempo final	30 s
θ_0	Condición inicial en desplazamiento	$\frac{179\pi}{180}\text{ rad}$
$\dot{\theta}_0$	Condición inicial en velocidad	0 rad/s

Tabla 7.39. Valores numéricos considerados para particularizar el *PVI* (7.43).

Finalmente, sustituyendo los valores numéricos en (7.44), se tiene (7.45):

$$\begin{cases} \ddot{\theta}(t) = -\frac{49}{5}\sin(\theta(t)) + \frac{7}{10}\cos(2t) \\ \theta(0) = \frac{179\pi}{180}\text{ rad}, \quad \dot{\theta}(0) = 0\text{ rad/s} \\ t \in [0,30]\text{ (s)} \end{cases}$$

Ecuación (7.45)

A continuación, se desarrolla un **extenso procedimiento con varios subapartados que pretende experimentar y probar, utilizando este problema y su realidad física, la reducción del coste computacional asociado al empleo de estrategias de interpolación** como algunas de las desarrolladas en este trabajo frente a la aproximación discreta mediante métodos numéricos en mallas muy densificadas.

También se estudian las prestaciones de algunos pares encajados de métodos *Runge-Kutta* implementados en *MATLAB* (expuestos brevemente las *Tablas 3.3* y *3.4*).

7.7.1. Solución numérica discreta con pares encajados de métodos *Runge-Kutta*. Estudio del coste computacional.

Multitud de situaciones reales implican el desconocimiento de la solución de un *PVI* como el dado por (7.45). Por ello, una de las prácticas comunes en la búsqueda de la misma consiste en emplear algún método numérico como los expuestos en el **Capítulo 3** para la obtención de soluciones discretas aproximadas utilizando una malla de la forma (2.11).

También suele ser un problema, si efectivamente se desconoce por completo la forma que ha de tomar la solución, la elección de una malla de nodos adecuada. Por ello, una alternativa suelen ser los *solvers* que implementan

pares encajados de métodos *Runge-Kutta*. *MATLAB* dispone de multitud de ellos implementados, los cuales han sido comentados en la sección 3.2.8. Estos algoritmos tienen la capacidad de computar automáticamente el paso y la discretización de la variable independiente del *PVI* conocido el intervalo en el que se ha de obtener la solución aproximada, empleando para ello un flujograma como el de la *Figura 3.3*. El cómputo automático de h_k responde a las tolerancias fijadas por el usuario que obtienen el **Error Local de Truncamiento o ELT [17]** cometido (por el método de orden p respecto del de orden $p + 1$). Estas tolerancias son absolutas o relativas y el criterio de error viene dado por (7.46):

$$|ELT_k| \leq \max(\text{RelTol} \cdot |y_k|, \text{AbsTol}(k)) \rightarrow \begin{cases} \text{RelTol}_{def} = 10^{-3} \\ \text{AbsTol}_{def} = 10^{-6} \end{cases}$$

Ecuación (7.46)

Partiendo de los valores por defecto de las tolerancias citados en la expresión anterior, se puede intentar obtener la solución numérica de (7.45) empleando los *solvers* para problemas estrictamente no rígidos de la *Tabla 3.3*. Procediendo, se tiene la *Figura 7.40*, en donde tan sólo se han obtenido valores discretos de la solución del *PVI* con los *solvers* citados, sin la utilización de ninguna estrategia de interpolación posterior.

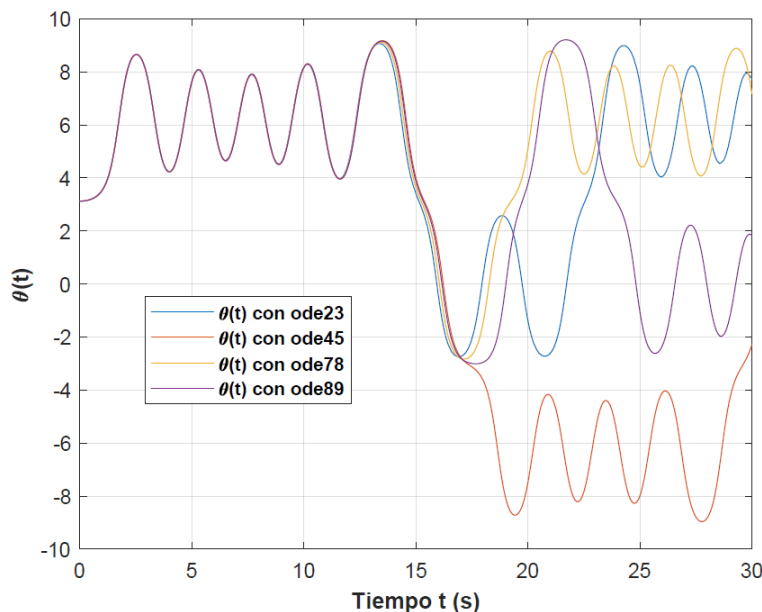


Figura 7.40. Solución numérica discreta del *PVI* (7.45) mediante solvers de la *Tabla 3.3*. Tolerancias por defecto de *MATLAB* para el cómputo del error en cada paso.

Los resultados no son correctos, pues la solución aproximada obtenida por cada solver mediante el cómputo automático de la malla (2.11) y el paso (2.12) no presenta similitudes con las demás a partir del nodo $t \cong 14$ (s). Haciendo un *zoom* a la figura anterior en el entorno en el que se aprecian las primeras diferencias, se tiene la *Figura 7.41*:

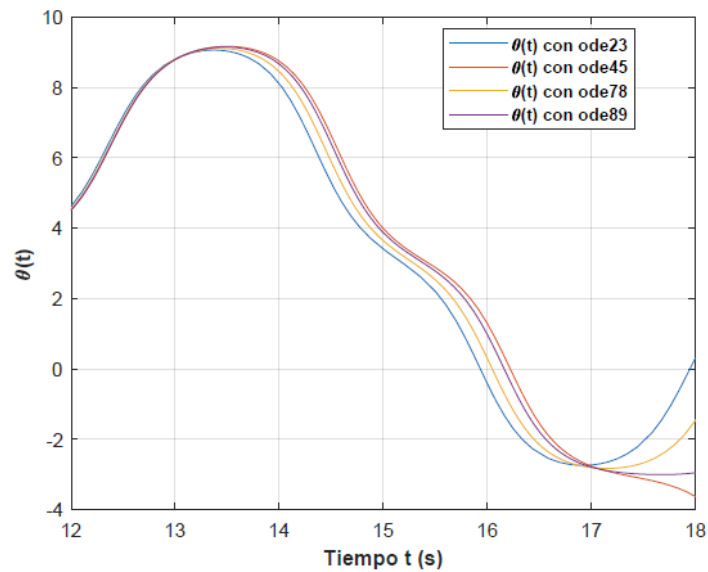


Figura 7.41. Zoom de la Figura 7.46 en el intervalo [12,18] (s).

Esto hace sospechar que las tolerancias de (7.46) son demasiado grandes, impidiendo una buena aproximación por el cómputo de un error excesivamente elevado. Reduciendo dichas tolerancias por defecto a los valores de (7.47), se vuelven a computar las soluciones numéricas, obteniendo los resultados de la Figura 7.42.

$$RelTol_{manual} = 10^{-6}, \quad AbsTol_{manual} = 10^{-6}$$

Ecuación (7.47)

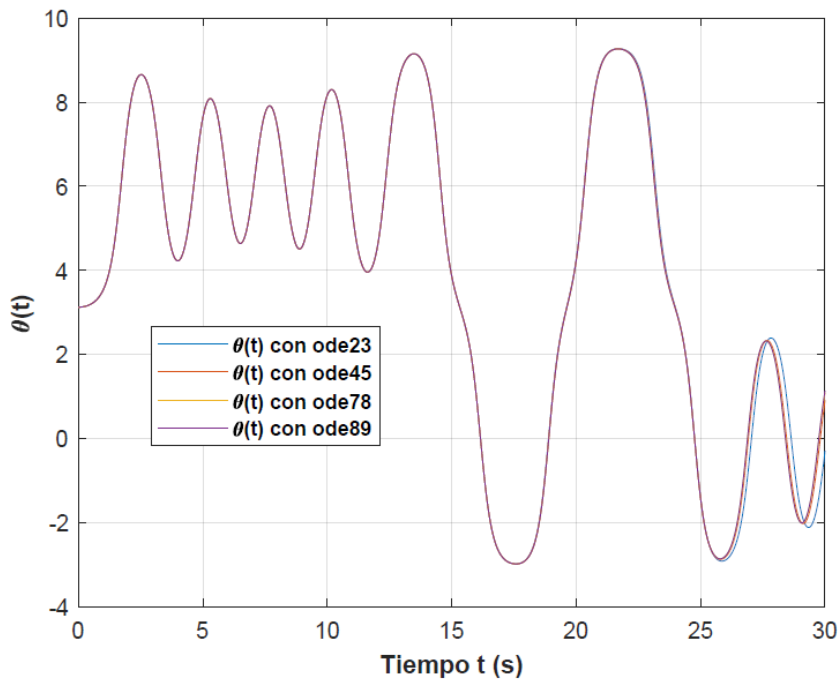


Figura 7.42. Solución numérica discreta del PVI (7.45) mediante solvers de la Tabla 3.3. Tolerancias mejoradas para el cómputo del error en cada paso.

Ahora, todos los solvers proporcionan aproximaciones a la solución de (7.45) cuyas características cualitativas son similares, puesto que tan sólo existen ciertas diferencias visuales a partir del intervalo [25,30] (s). Esto es perfectamente posible dado los órdenes de convergencia dispares de unos métodos y otros. Haciendo un zoom en dicho intervalo, se tiene la **Figura 7.43**. Nótese la variabilidad de la frecuencia de la respuesta temporal $\theta(t)$, la cual aflora dadas las no linealidades del problema abordado a pesar de excitar el sistema armónicamente a una frecuencia Ω .

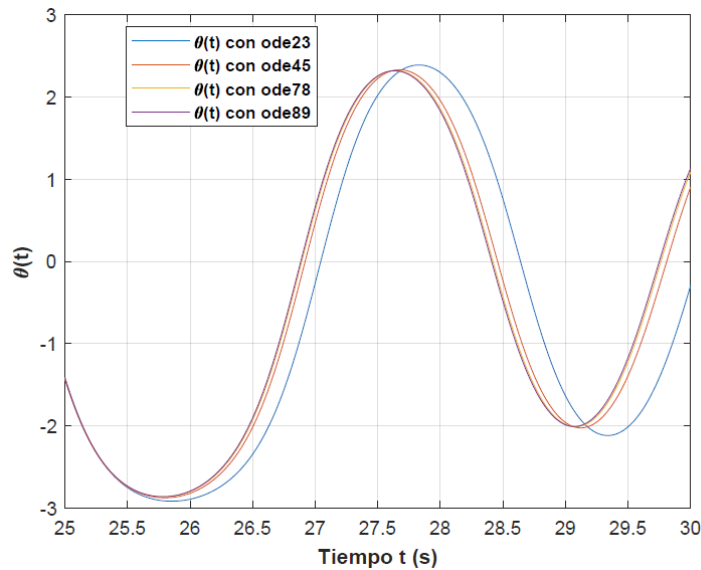


Figura 7.43. Zoom de la **Figura 7.42** en el intervalo [25,30] (s).

La solución que más se desvía de las demás es la obtenida por “ode23”, ya que los órdenes de convergencia de los métodos *Runge-Kutta* que componen el par encajado son, respectivamente, 2 y 3, en comparación con los órdenes más altos de “ode45”, “ode78” y “ode89”, en igualdad de tolerancias (7.47) para el cómputo del error y ajuste del paso h_k .

Las estadísticas más relevantes de todas las mallas computadas automáticamente por los solvers empleados se indican en la **Tabla 7.40**:

<i>Solver</i>	<i>Alias malla</i>	<i>Nodos totales</i>	h_{min}	<i>Nodo t_k de $[t_{k-1}, t_k]$ con h_{min}</i>
<i>ode23</i>	“t23m”	2924	0.0072	24.7637
<i>ode45</i>	“t45m”	1553	0.0134	18.9670
<i>ode78</i>	“t78m”	937	0.0054	29.9891
<i>ode89</i>	“t89m”	929	0.0123	29.9754

Tabla 7.40. Estadísticas de las mallas auto computadas por los solvers empleados para aproximar soluciones discretas del **PVI (7.45)**.

Se aprecia un incremento notable del coste computacional para los solvers de menores órdenes de convergencia (“ode23” y “ode45”), los cuales, además de aproximar peor, utilizan más evaluaciones de la **EDO** del **PVI (7.45)** (de su

segunda derivada $\ddot{\theta}(t)$ al expresar la ecuación en forma normal). Esto implica considerar, por tanto, el solver “*ode89*” como el más exacto y eficiente de todos, ya que el número de evaluaciones requerido es el más pequeño para satisfacer las tolerancias de (7.47).

Ahora, y sin perjuicio de que los valores de h_{min} en cada malla sean los dados por la **Tabla 7.40** en el subintervalo definido por su nodo extremo t_k , se puede observar la evolución global del paso h_k que ha sido necesario computar para obtener las aproximaciones con cada solver. Considérese, por tanto, la **Figura 7.44**:

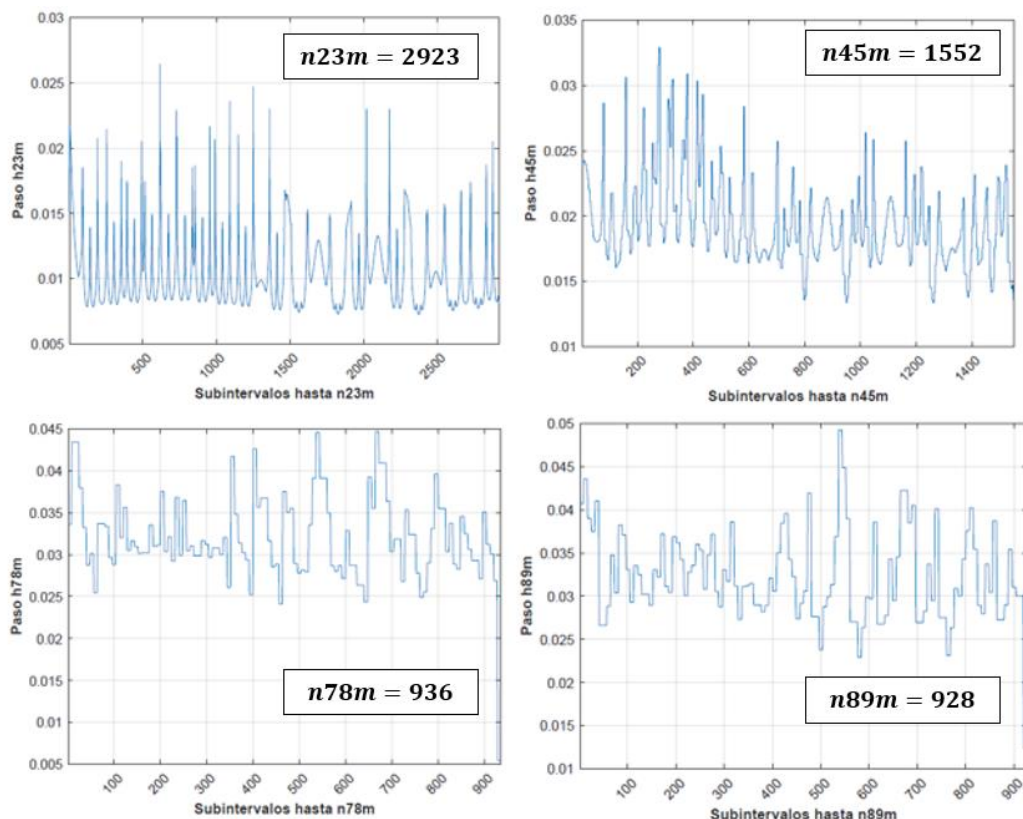


Figura 7.44. Evolución global del paso h_k para cada una de las mallas auto computadas con los solvers empleados para aproximar soluciones discretas del PVI (7.45).

Estas gráficas no hacen más que corroborar lo sospechado. En concordancia con una malla más densa y no equiespaciada, “*ode23*” y “*ode45*” precisan de un paso acotado en valores inferiores (máximos entre 0.025 y 0.03 y mínimos por debajo de 0.015 y 0.01) que el computado por “*ode78*” y “*ode89*” (máximos sobre 0.04 con mínimos mayores de 0.025 en la mayoría de los casos).

Dando ahora un nuevo enfoque al estudio, basado éste en asumir la solución discreta proporcionada por “*ode89*” en los nodos de la malla “*t89m*” como la más exacta, se pueden computar las soluciones que el resto de solvers proporcionan en dicha malla, con el objeto de estudiar el error cometido en estricto cumplimiento de (7.47) y respecto del método de mayor orden

supuesto exacto. Asumiendo unas soluciones cualitativamente con la forma de la **Figura 7.42**, se grafica el error cometido por “*ode23*”, “*ode45*” y “*ode78*” respecto de “*ode89*” en la **Figura 7.45**, en donde se indican claramente los subintervalos que cada gráfica tiene en cuenta, para no entrar en conflicto con los órdenes de magnitud variables del error en el dominio temporal de (7.45).

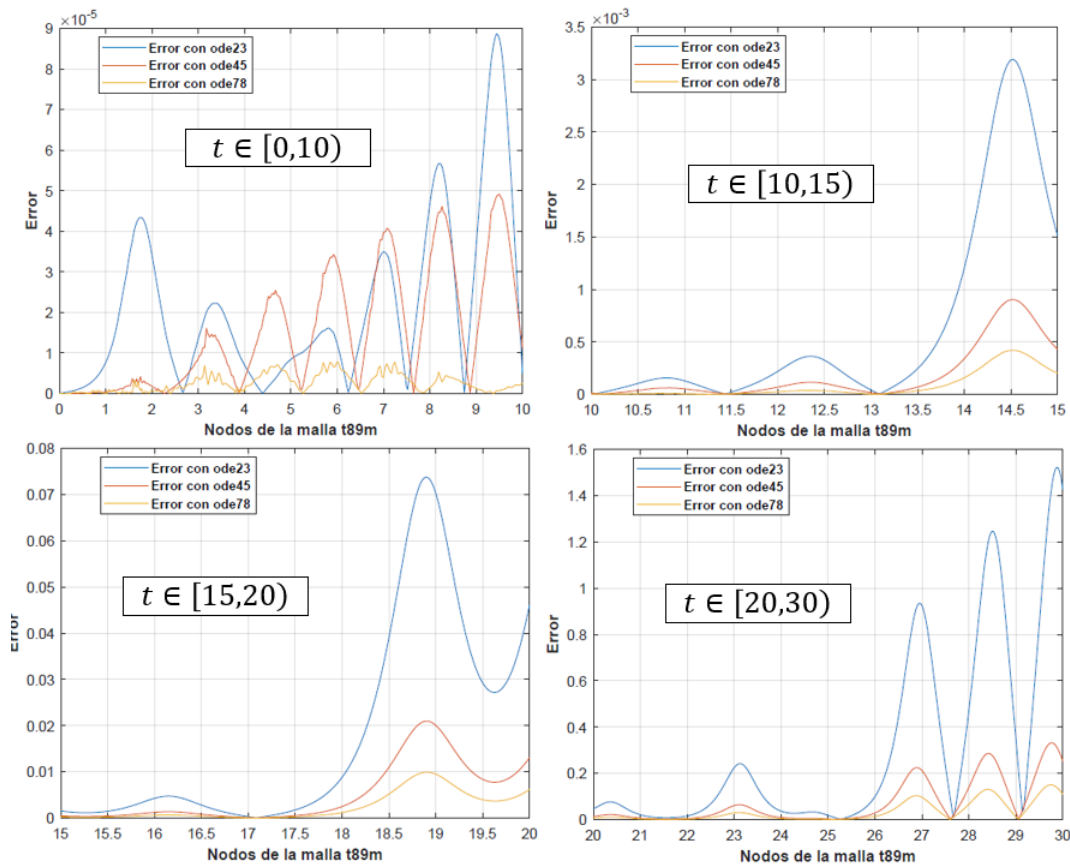


Figura 7.45. Evolución del error cometido por los solvers empleados respecto de la solución discreta aproximada de (7.45) dada por “*ode89*” en los nodos de la malla “*t89m*”.

El error respecto de “*ode89*” incrementa su orden de magnitud progresivamente, alcanzando valores grandes y superiores a la unidad en los últimos nodos de “*t89m*”. Es preciso recordar que los métodos *Runge-Kutta* son de un paso (sección 3.2 del **Capítulo 3**), por lo que la calidad de la aproximación en un determinado nodo t_k siempre depende de la aproximación en el nodo inmediatamente anterior t_{k-1} , pudiéndose amplificar el error cometido progresivamente. Los errores máximos de la **Figura 7.45** se indican en la **Tabla 7.41**, junto con el nodo t_k en el que tienen lugar.

<i>Solver</i>	<i>Error máximo</i>	<i>Nodo t_k</i>
<i>ode23</i>	1.52E+00	29.8714
<i>ode45</i>	3.32E-01	29.7515
<i>ode78</i>	1.50E-01	29.7515

Tabla 7.41. Errores máximos cometidos por los solvers “*ode23*”, “*ode45*” y “*ode78*” respecto a la solución discreta aproximada de (7.45) dada por “*ode89*” en los nodos de la malla “*t89m*”.

7.7.1.1. Conclusiones principales sobre el coste computacional. Propuesta de estrategias alternativas de resolución.

Una vez computadas las soluciones de (7.45) mediante el empleo de los solvers de **MATLAB** expuestos en la **Tabla 3.3**, se puede llegar a las siguientes conclusiones previas al empleo de otras estrategias de resolución:

- i. Conseguir buenas aproximaciones mediante una malla auto computada por algún método numérico de obtención de soluciones discretas aproximadas que tenga la capacidad de ello (como un par encajado de métodos *Runge-Kutta*) implica un número muy elevado de evaluaciones de la función que define un **PVI** (de 2^o orden o no) en forma normal, además de **no obtenerse una aproximación continua a la solución (aunque a simple vista lo parezca)**.
- ii. **El número de evaluaciones citado es inversamente proporcional a la disminución de tolerancias que el método ha de manejar para la obtención del error en cada paso** (cometido por el método de orden p respecto del de orden $p + 1$), tal y como indica la **Tabla 7.40**. Además, si los órdenes de convergencia son bajos, el coste se dispara para satisfacer tolerancias pequeñas (disminución del paso por densificado de malla, **Figura 7.44**) como las dadas por (7.47).
- iii. Relajar el número de evaluaciones o incrementar las tolerancias para computar una malla del tipo (2.11) menos densa y costosa puede conllevar, para ciertos solvers con órdenes de convergencia demasiado bajos aplicados a la resolución de un problema en cuestión, el fallo del método numérico, tal y como se puede observar en la **Figura 7.40** (disparidad de los resultados para cada solver a partir de cierta región del dominio temporal del **PVI**).
- iv. Los problemas relativos a las evaluaciones de una determinada función que define un **PVI** formulado en términos de una **EDO** expresada en forma normal **se agravan cuando dicha función es muy complicada o directamente sólo es posible su determinación y evaluación empírica**. Este ejemplo plantea un caso en el que la predicción de la respuesta frente a una excitación armónica con frecuencia fija Ω no es sencilla de hacer, pero existen multitud de supuestos mucho más complicados de abordar.

En estricta atención a las conclusiones anteriores y a todas las observaciones adicionales realizadas en esta sección, **se concluye que el empleo único de métodos numéricos para la obtención de soluciones discretas aproximadas de un PVI empleando mallas muy densas y pasos pequeños no resulta ser una buena y eficiente alternativa**, por lo que se requiere del empleo de estrategias más baratas computacionalmente pero fieles a la naturaleza del problema a abordar, con la capacidad adicional de proporcionar aproximaciones continuas.

Estando este trabajo orientado a las técnicas y estrategias de interpolación de soluciones de problemas formulados en términos de **EDOs** de 2^o orden, y



presentándose en la resolución del *PVI (7.45)* una problemática como la expuesta con anterioridad, se propone la experimentación con algunas estrategias ya utilizadas en anteriores experimentos numéricos de este capítulo. El objetivo es observar una disminución en el coste computacional (mallas menos densas y pasos mucho más grandes) respecto al obtenido empleando únicamente *solvers* de la *Tabla 3.3*, monitorizando el error de interpolación cometido.

7.7.2. Solución numérica continua empleando estrategias de interpolación. Tendencia del error.

El enfoque que toma la resolución de *(7.45)* es ahora distinto al de la sección *7.7.1*. Las consideraciones iniciales que han de ser tenidas en cuenta para comprender la estructura de este apartado son:

- i. Para poder interpolar, se precisa de un vector de soluciones discretas aproximadas θ_k en t_k . En concordancia con los resultados y comentarios de las *Figuras 7.43, 7.44, 7.45*, así como la *Tabla 7.40*, se elige el *solver "ode89"* para obtener dicho vector. No obstante, y aprovechando la alta precisión de la que goza el par encajado, se reduce notablemente el número de nodos en los que la solución ha de ser computada. Por ello, considérense las tres mallas equiespaciadas que discretizan el dominio temporal continuo de *(7.45)* dadas por la *Tabla 7.42*:

Malla	Alias	Subintervalos (nodos)	Paso uniforme h
1	t_1	$n_1 = 30$ (31)	$h_1 = 1$ (s)
2	t_2	$n_2 = 60$ (61)	$h_2 = 0.5$ (s)
3	t_3	$n_3 = 120$ (121)	$h_3 = 0.25$ (x)

Tabla 7.42. Mallas equiespaciadas y poco densificadas para el cómputo de soluciones discretas aproximadas del *PVI (7.45)* mediante *"ode89"*.

- ii. Debe de asumirse la obtención de soluciones discretas en dichas mallas como posibles valores empíricos proporcionados por una determinada observación sobre un sistema de estudio cuyo comportamiento viene formulado en términos de fenómenos de 2º orden con características desconocidas. Aquí, dado el problema que se está abordando, el punto de partida es más favorable al conocer la forma explícita del *PVI*, pero podría ser mucho más complejo y tedioso, como ya se ha expuesto en *7.7.1.1*.
- iii. Se dispone, desde un punto de vista instrumental, de la solución de *(7.45)* en un número muy grande de nodos y proporcionada por *"ode89"* para permitir el cómputo del error cometido por las estrategias de interpolación y enjuiciar el comportamiento de las mismas. En una situación real tan sólo se dispondría de θ_k en los nodos de las mallas de la *Tabla 7.42*, habiéndose demostrado un densificado de malla

desproporcionado como profundamente ineficiente y con un coste computacional elevado.

Se emplean, de ahora en adelante, estrategias de interpolación basadas en *splines* cuadráticos (sección 4.3 del **Capítulo 4**) y *pseudo-splines* quínticos (sección 5.3 del **Capítulo 5**). Los valores numéricos de la solución de (7.45) en los nodos de las mallas equiespaciadas propuestas y aproximadas mediante “ode89” en estricto cumplimiento de las tolerancias de (7.47) vienen dados por las **Tablas 7.43** (malla t_1), **7.44** (malla t_2) y **7.45** (malla t_3), utilizando aritmética de 4 cifras decimales significativas.

Malla t_1	$\theta(t_1)$	Malla t_1	$\theta(t_1)$
0.00	3.1241	16.00	0.9375
1.00	3.5431	17.00	-2.7988
2.00	7.6135	18.00	-2.8821
3.00	7.8410	19.00	0.6477
4.00	4.2268	20.00	4.2240
5.00	7.6410	21.00	8.8793
6.00	5.9375	22.00	9.2229
7.00	5.8393	23.00	6.9624
8.00	7.3869	24.00	3.0574
9.00	4.5761	25.00	-1.4313
10.00	8.1704	26.00	-2.7988
11.00	5.4097	27.00	0.6442
12.00	4.5191	28.00	1.8424
13.00	8.8006	29.00	-1.9882
14.00	8.6596	30.00	1.1112
15.00	3.8597		

Tabla 7.43. Valores numéricos de la solución discreta aproximada del **PVI (7.45)** proporcionada por “ode89” en los nodos de la malla t_1 .

Malla t_2	$\theta(t_2)$	Malla t_2	$\theta(t_2)$	Malla t_2	$\theta(t_2)$
0.00	3.1241	10.50	7.8112	21.00	8.8798
0.50	3.1972	11.00	5.4097	21.50	9.2522
1.00	3.5431	11.50	4.0027	22.00	9.2233
1.50	4.8876	12.00	4.5191	22.50	8.7984
2.00	7.6135	12.50	7.0538	23.00	6.9680
2.50	8.6570	13.00	8.8006	23.50	4.2595
3.00	7.8410	13.50	9.1540	24.00	3.0581
3.50	5.3460	14.00	8.6596	24.50	1.5217
4.00	4.2268	14.50	6.3714	25.00	-1.4312
4.50	5.3657	15.00	3.8596	25.50	-2.7385
5.00	7.6410	15.50	2.7671	26.00	-2.8015
5.50	7.8877	16.00	0.9374	26.50	-1.9288
6.00	5.9375	16.50	-1.8135	27.00	0.6253
6.50	4.6401	17.00	-2.7988	27.50	2.2519

7.00	5.8393	17.50	-2.9844	28.00	1.8555
7.50	7.7439	18.00	-2.8819	28.50	-0.4361
8.00	7.3869	18.50	-2.0558	29.00	-1.9861
8.50	5.2288	19.00	0.6496	29.50	-1.2188
9.00	4.5761	19.50	2.7968	30.00	1.0901
9.50	6.3389	20.00	4.2253		
10.00	8.1704	20.50	7.0814		

Tabla 7.44. Valores numéricos de la solución discreta aproximada del PVI (7.45) proporcionada por “ode89” en los nodos de la malla t_2 .

Malla t_3	$\theta(t_3)$	Malla t_3	$\theta(t_3)$	Malla t_3	$\theta(t_3)$	Malla t_3	$\theta(t_3)$
0.00	3.1241	7.75	7.8911	15.25	3.2619	22.75	8.1595
0.25	3.1411	8.00	7.3869	15.50	2.7671	23.00	6.9705
0.50	3.1972	8.25	6.3363	15.75	2.0850	23.25	5.4452
0.75	3.3136	8.50	5.2289	16.00	0.9374	23.50	4.2609
1.00	3.5431	8.75	4.5955	16.25	-0.5854	23.75	3.5553
1.25	4.0022	9.00	4.5761	16.50	-1.8136	24.00	3.0585
1.50	4.8876	9.25	5.1897	16.75	-2.4857	24.25	2.4806
1.75	6.2657	9.50	6.3389	17.00	-2.7988	24.50	1.5219
2.00	7.6135	9.75	7.5000	17.25	-2.9351	24.75	0.0400
2.25	8.4018	10.00	8.1704	17.50	-2.9844	25.00	-1.4312
2.50	8.6570	10.25	8.2763	17.75	-2.9737	25.25	-2.3285
2.75	8.4843	10.50	7.8112	18.00	-2.8818	25.50	-2.7387
3.00	7.8410	10.75	6.7433	18.25	-2.6309	25.75	-2.8657
3.25	6.6693	11.00	5.4097	18.50	-2.0553	26.00	-2.8027
3.50	5.3460	11.25	4.4391	18.75	-0.9252	26.25	-2.5339
3.75	4.4857	11.50	4.0027	19.00	0.6505	26.50	-1.9339
4.00	4.2268	11.75	4.0204	19.25	1.9694	26.75	-0.8267
4.25	4.5149	12.00	4.5191	19.50	2.7971	27.00	0.6167
4.50	5.3657	12.25	5.6102	19.75	3.4291	27.25	1.7310
4.75	6.6069	12.50	7.0538	20.00	4.2258	27.50	2.2506
5.00	7.6410	12.75	8.1827	20.25	5.4992	27.75	2.2834
5.25	8.0755	13.00	8.8006	20.50	7.0822	28.00	1.8614
5.50	7.8877	13.25	9.0808	20.75	8.2655	28.25	0.9056
5.75	7.1010	13.50	9.1540	21.00	8.8801	28.50	-0.4250
6.00	5.9375	13.75	9.0421	21.25	9.1509	28.75	-1.5059
6.25	4.9882	14.00	8.6595	21.50	9.2523	29.00	-1.9851
6.50	4.6401	14.25	7.8085	21.75	9.2682	29.25	-1.8901
6.75	4.9421	14.50	6.3714	22.00	9.2235	29.50	-1.2289
7.00	5.8393	14.75	4.8607	22.25	9.0954	29.75	-0.0729
7.25	6.9716	15.00	3.8596	22.50	8.7992	30.00	1.0804
7.50	7.7438						

Tabla 7.45. Valores numéricos de la solución discreta aproximada del PVI (7.45) proporcionada por “ode89” en los nodos de la malla t_3 .

7.7.2.1. Estrategia de interpolación con *splines* cuadráticos.

Se consideran *splines* cuadráticos como la primera estrategia de interpolación a ensayar para el cómputo de la solución continua del PVI (7.45). Se construyen un total de tres trazadores polinomiales con estas características, uno por cada malla de la **Tabla 7.42** empleando las soluciones discretas que proporciona “ode89” indicadas en las tres tablas anteriores. El alias asignado a cada *spline* creado, así como la estructura de todos los resultados para las soluciones continuas y el error de interpolación cometido respecto a la solución nodal muy densificada (también obtenida con “ode89”) a evitar por las consideraciones ya expuestas previamente se resumen todos ellos en la **Tabla 7.46**:

<i>Malla</i>	<i>Alias spline</i>	<i>Solución continua</i>	<i>Error cometido</i>
t_1	“s2int1”	Figura 7.46	Figura 7.47
t_2	“s2int2”	Figura 7.48	Figura 7.49
t_3	“s2int3”	Figura 7.50	Figura 7.51

Tabla 7.46. Cuadro sinóptico resumen de *splines* cuadráticos creados para aproximar soluciones continuas del PVI (7.45).

A continuación, se indican todas las figuras mencionadas, para su posterior análisis y comparación. Cabe destacar que se grafica la solución nodal muy densificada ya mencionada previamente $\theta(t)$, con el objeto de conocer la forma que tomaría la solución exacta en el caso de estar disponible, dejando claro que en una situación real con dificultad para evaluar la función del PVI en forma normal se debe de confiar la aproximación al propio trazador en los nodos de las mallas de la **Tabla 7.42**, sin disponer, evidentemente, de $\theta(t)$.

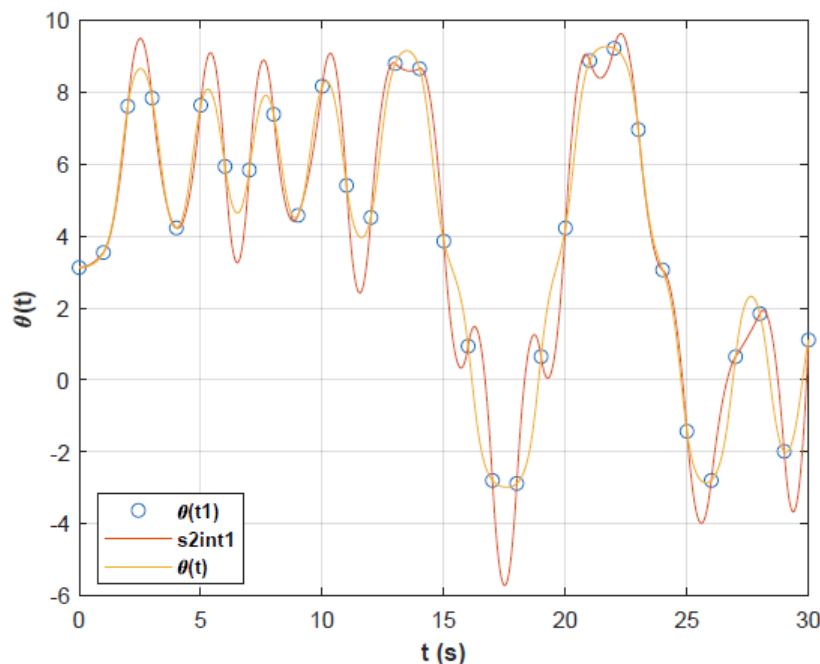


Figura 7.46. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_1 por el *spline* cuadrático “s2int1”.

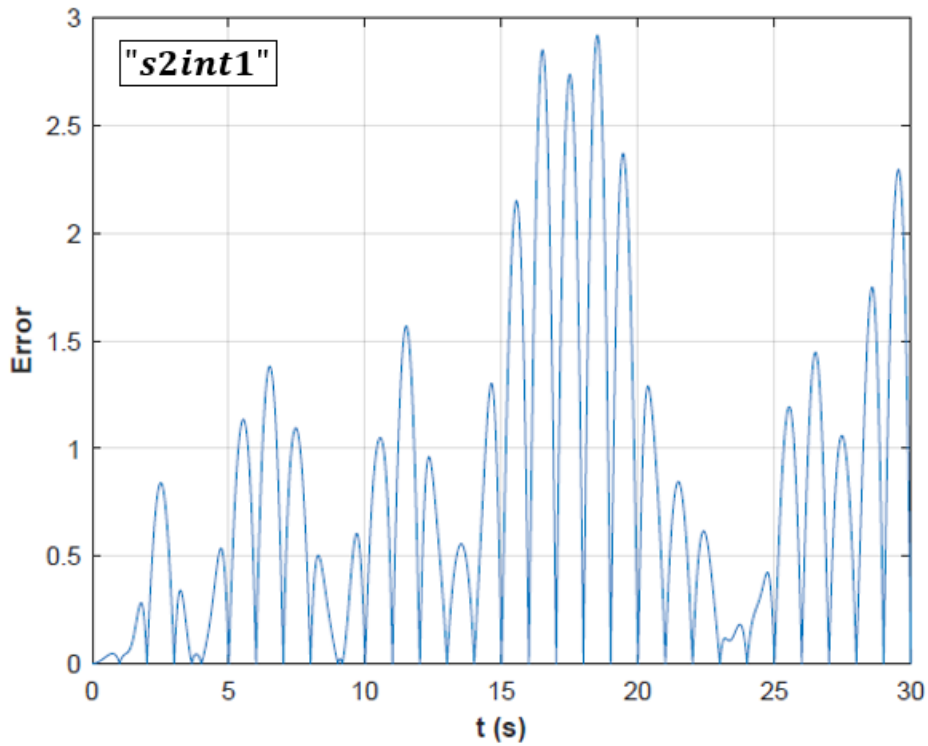


Figura 7.47. Error de interpolación cometido por el *spline* cuadrático “s2int1” respecto de la solución discreta muy densificada a evitar computada con “ode89”.

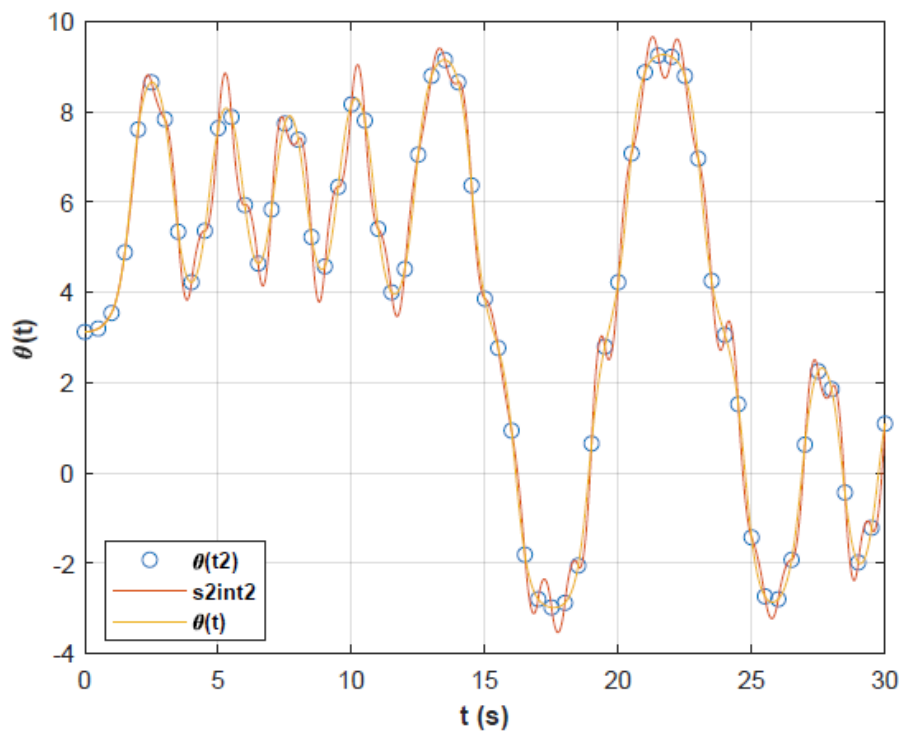


Figura 7.48. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_2 por el *spline* cuadrático “s2int2”.

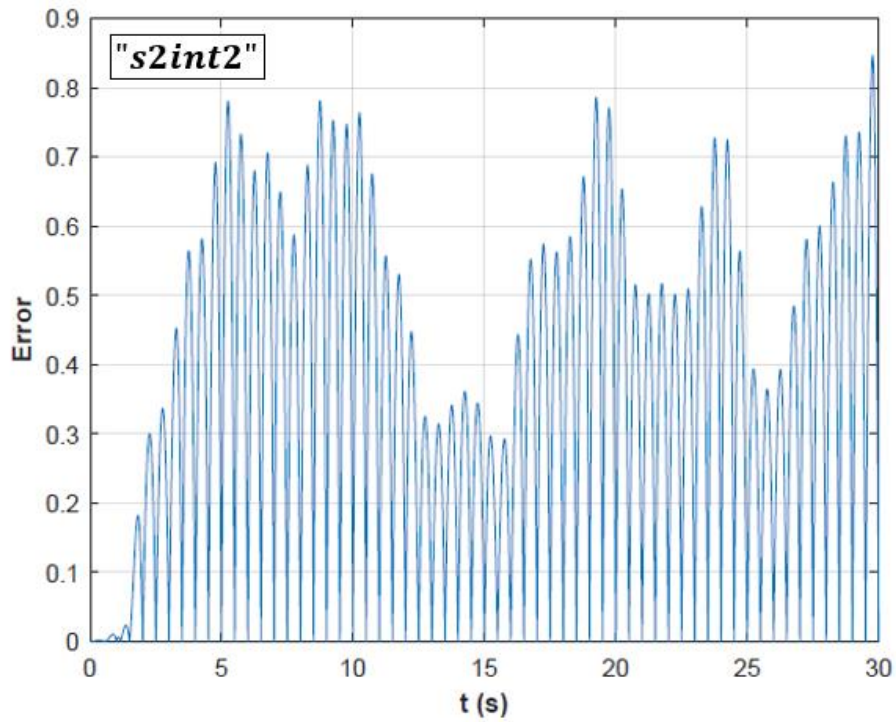


Figura 7.49. Error de interpolación cometido por el *spline* cuadrático "s2int2" respecto de la solución discreta muy densificada a evitar computada con "ode89".

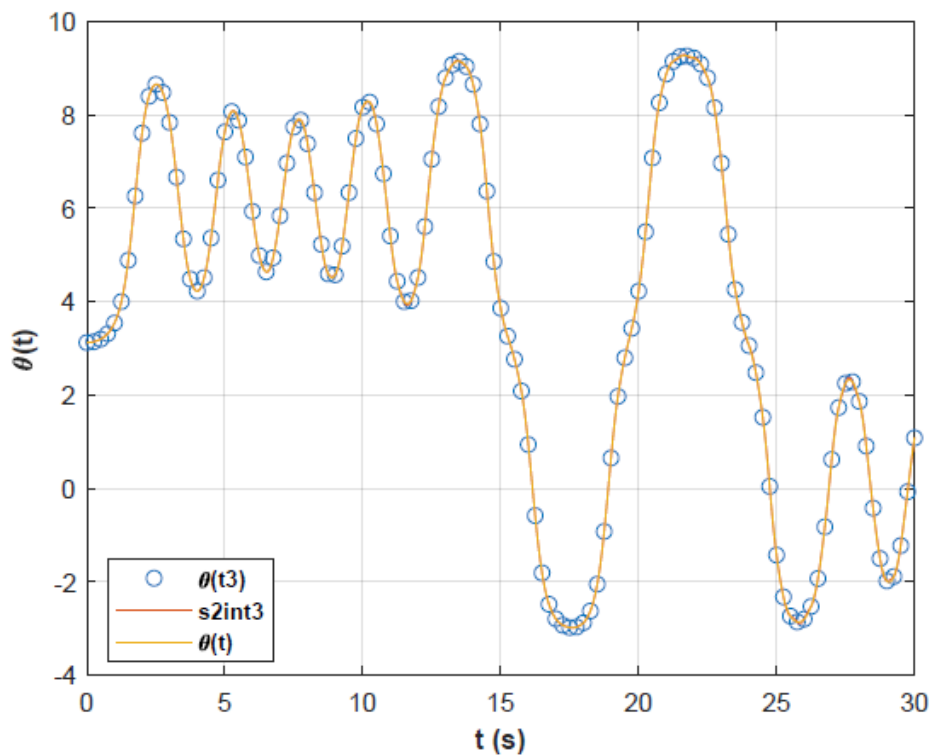


Figura 7.50. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_3 por el *spline* cuadrático "s2int3".

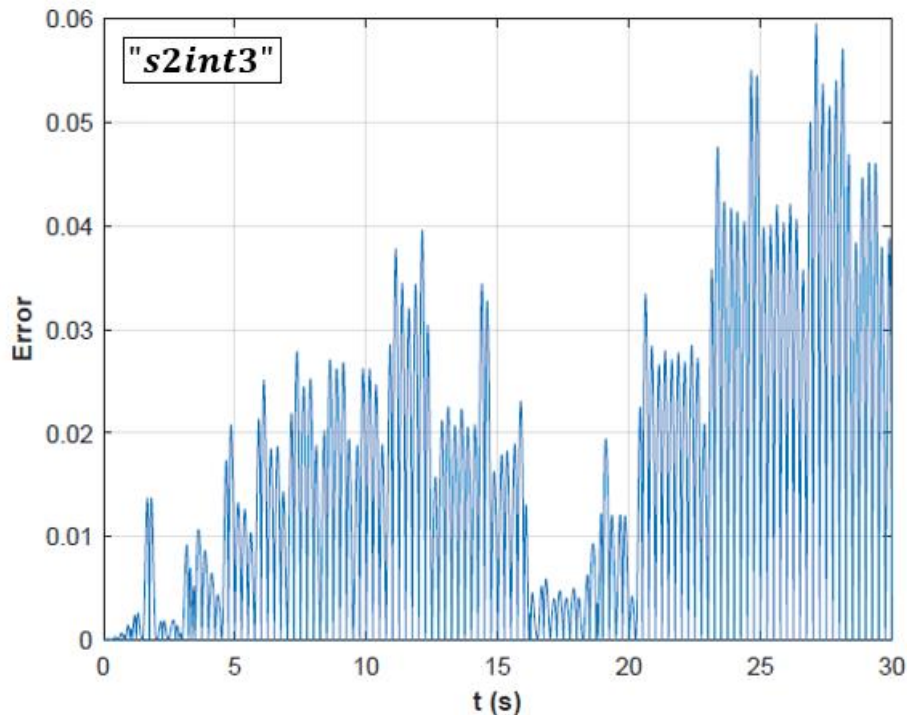


Figura 7.51. Error de interpolación cometido por el *spline* cuadrático “*s2int3*” respecto de la solución discreta muy densificada a evitar computada con “*ode89*”.

A la vista de todos los resultados anteriores, los *splines* cuadráticos interpoladores en las mallas t_1 y t_2 (**Figuras 7.46** y **7.48**, respectivamente) presentan dificultades para aproximar en zonas con cambio de curvatura (en este caso, oscilaciones a frecuencia variable surgidas de las no linealidades de **(7.45)**). El motivo por el cual esto ocurre ya ha sido comentado en detalle en el **experimento numérico 3 (sección 7.4)**, haciendo hincapié en la imposibilidad de que un trazador polinomial con continuidad (regularidad) C^1 que interpola soluciones discretas de *PVs* se comporte de una forma monótonamente aceptable y sin demasiados puntos de inflexión (ligados a las discontinuidades en la curvatura, a su vez dependiente de la segunda derivada que no es continua).

En cuanto al error de interpolación cometido con el empleo de las dos primeras mallas, destaca el hecho de que éste disminuye desde valores máximos muy por encima de la unidad (malla t_1 , **Figura 7.47**) hasta valores ligeramente inferiores a 1 (malla t_2 , **Figura 7.48**). Además, **su tendencia es estable**, no acumulándose y amplificándose conforme avanza el esquema recursivo indicado en la ecuación **(4.14)** (sección **4.3.3** del **Capítulo 4**).

Procediendo a observar los resultados para la última malla de la **Tabla 7.42** (t_3), no se aprecian diferencias visibles en la **Figura 7.50** entre el *spline* “*s2int3*” y la solución discreta dada por “*ode89*”, pudiendo comentar únicamente que el error de interpolación (**Figura 7.51**) disminuye respecto al obtenido para las mallas t_1 y t_2 (ahora acotado superiormente en valores no superiores a $6 \cdot 10^{-2}$). **Para observar mejor la naturaleza de la aproximación continua con este último trazador**, considérense los dos *zooms* de **la Figura**

7.50 indicados por las *Figuras 7.52* (subintervalo $[20.5, 22.5]$) y *7.53* (subintervalo $[25, 30]$).

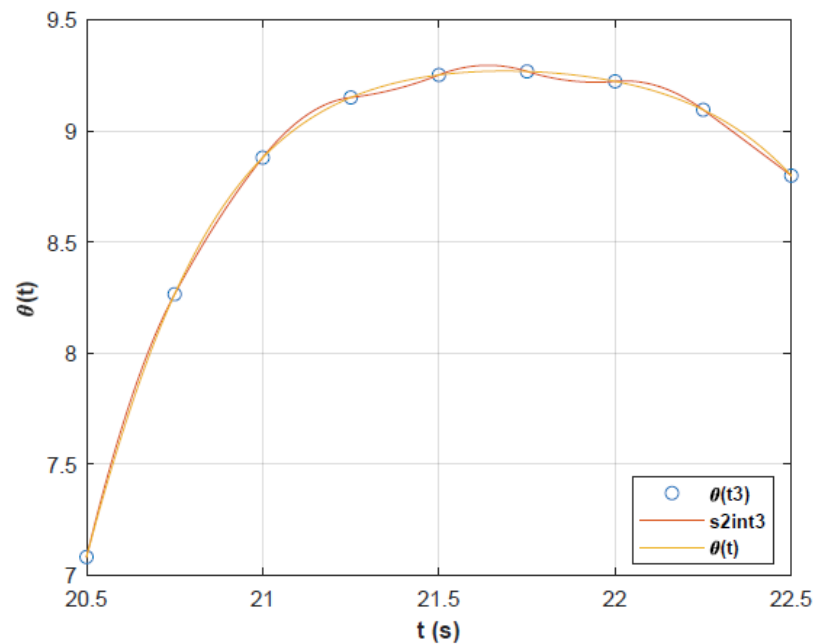


Figura 7.52. Zoom de la *Figura 7.50* en $[20.5, 22.5]$ (s).

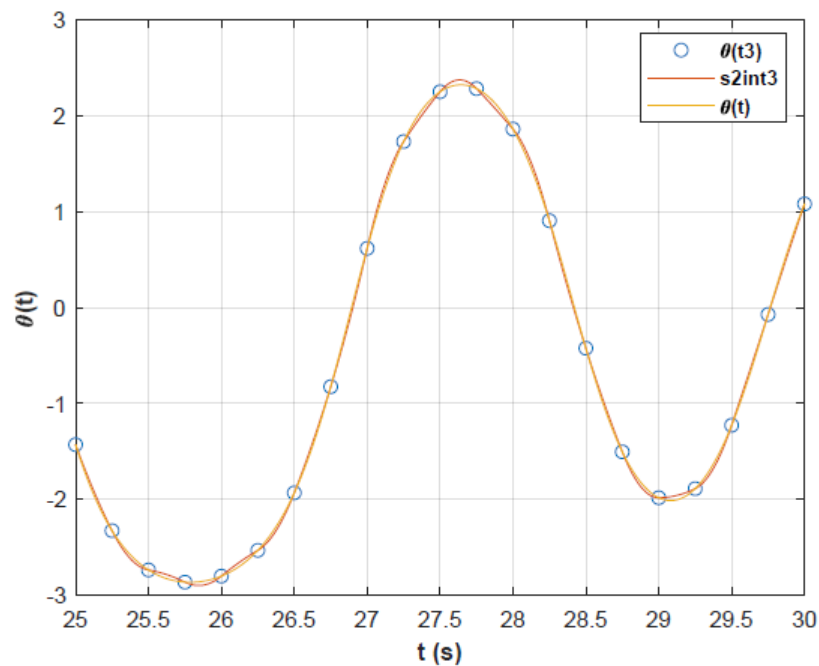


Figura 7.53. Zoom de la *Figura 7.50* en $[25, 30]$ (s).

Efectivamente, el problema en la regularidad del *spline* persiste aunque el error haya disminuido, por lo que las consideraciones cualitativas acerca del trazador “s2int3” y la malla t_3 son las mismas que las aportadas anteriormente para los otros casos.



Si el lector ha seguido y observado con atención todo lo comentado en esta sección hasta ahora, puede haberse percatado de que, efectivamente, y en concordancia con el principal objetivo de este experimento, el empleo de estrategias de interpolación (de momento, sólo con *splines* cuadráticos) de las soluciones discretas aproximadas del *PVI* (7.45) resultan ser de un **coste computacional significativamente inferior al empleo único de métodos numéricos discretos como los pares encajados RK expuestos en 7.7.1.**

Sin la posibilidad de establecer un criterio de error de interpolación máximo contextualizado con la realidad al estar únicamente experimentando sobre (7.45) y el coste asociado a la forma de aproximar su solución, **se desprenden las conclusiones cualitativas siguientes:**

- i. El coste computacional disminuye al precisar de un menor número de evaluaciones de la función del *PVI* a resolver expresado en forma normal con la segunda derivada explícita. Lógicamente, la evaluación de un *spline* que interpola a la solución de (7.45) en alguna de las mallas equiespaciadas de la **Tabla 7.42** es mucho más barata al ser un trazador estrictamente polinómico de grado bajo, frente a las no linealidades de un *PVI* como el del péndulo estudiado.
- ii. El empleo satisfactorio de mallas equiespaciadas favorece **el desacoplo de la discretización según (2.11) del dominio de un PVI respecto de la forma y sus posibles complejidades** (desde no linealidades hasta desconocer por completo las expresiones analíticas teniendo que recurrir a observaciones empíricas).
- iii. En el caso de haber determinado un fallo en la estrategia de interpolación utilizada, podría haber sido necesario establecer un criterio de mallado con densificados locales dependientes de algún factor adicional. No obstante, como se demuestra, y al margen de los problemas de regularidad subsanables por interpolación con otro trazador (que imponga continuidad mayor que C^1), esto no ocurre y es evitable.
- iv. A pesar de haber obtenido todos los errores de interpolación que cada *spline* presenta respecto a la solución asumida como exacta en malla muy densificada empleando el *solver* “*ode89*”, **se reserva la interpretación cuantitativa de los mismos con la correspondiente valoración de igual índole para cada una de las mallas de la Tabla 7.42 a la sección posterior a este experimento, en la cual se emplean trazadores polinomiales con mejores condiciones de regularidad.**

7.7.2.2. Estrategia de interpolación con *pseudo-splines* quínticos. Reducción porcentual del coste computacional.

Como alternativa surgida al empleo de *splines* cuadráticos y sus discontinuidades en curvatura ligadas a la continuidad C^1 , se puede repetir el procedimiento de 7.7.2.1 empleando trazadores polinomiales con continuidad

C^2 , fijando el uso de *pseudo-splines* quínticos interpoladores (sección 5.3.3 del **Capítulo 5**). Considérese la **Tabla 7.47** como guía a este apartado.

Malla	Alias spline	Solución continua	Zooms	Error
t_1	"ps5int1"	Figura 7.54	[5, 10]: Figura 7.55	Figura 7.56
			[20.5, 22.5]: Figura 7.55	
t_2	"ps5int2"	Figura 7.57	[23, 25]: Figura 7.58	Figura 7.59
t_3	"ps5int3"	Figura 7.60	[18, 21]: Figura 7.61	Figura 7.62

Tabla 7.47. Cuadro sinóptico resumen de *pseudo-splines* quínticos creados para aproximar soluciones continuas del **PVI (7.45)**.

Empleando los mismos criterios para la presentación de resultados gráficos que con los *splines* cuadráticos, se procede a continuación con las figuras citadas en la tabla.

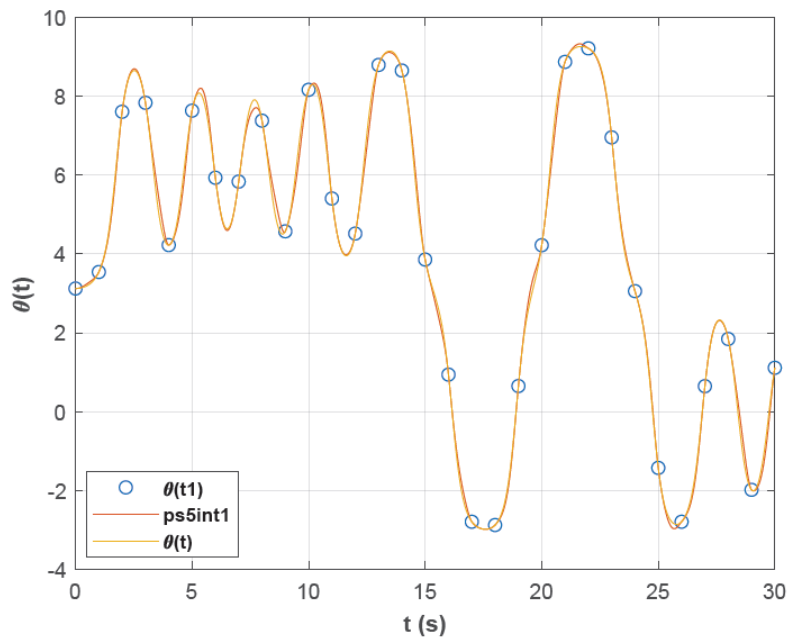


Figura 7.54. Solución continua del **PVI (7.45)** interpolada en los nodos de la malla t_1 por el *pseudo-spline* quíntico "ps5int1".

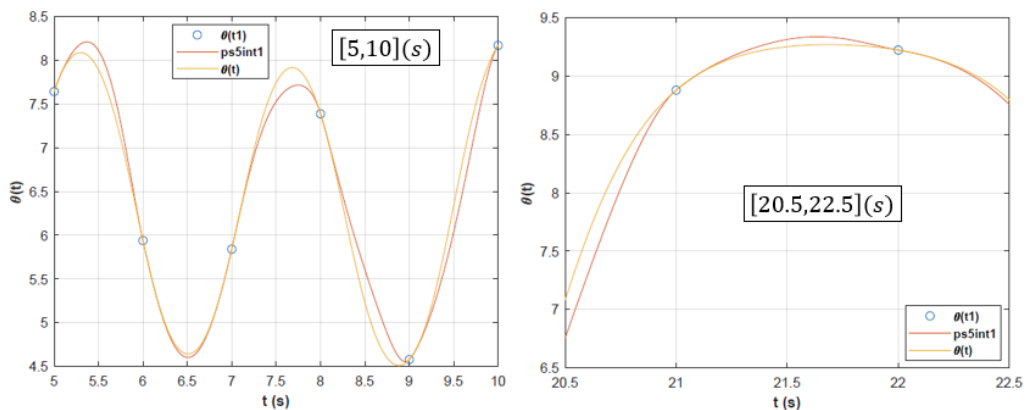


Figura 7.55. Zooms de la **Figura 7.54**.

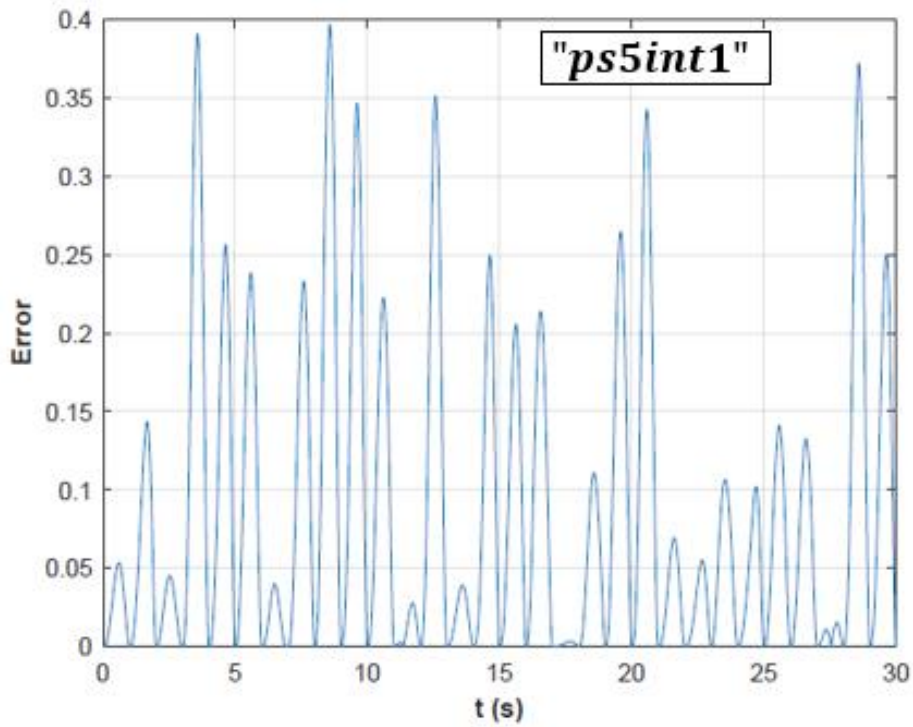


Figura 7.56. Error de interpolación cometido por el *pseudo-spline* quintico “*ps5int1*” respecto de la solución discreta muy densificada a evitar computada con “*ode89*”.

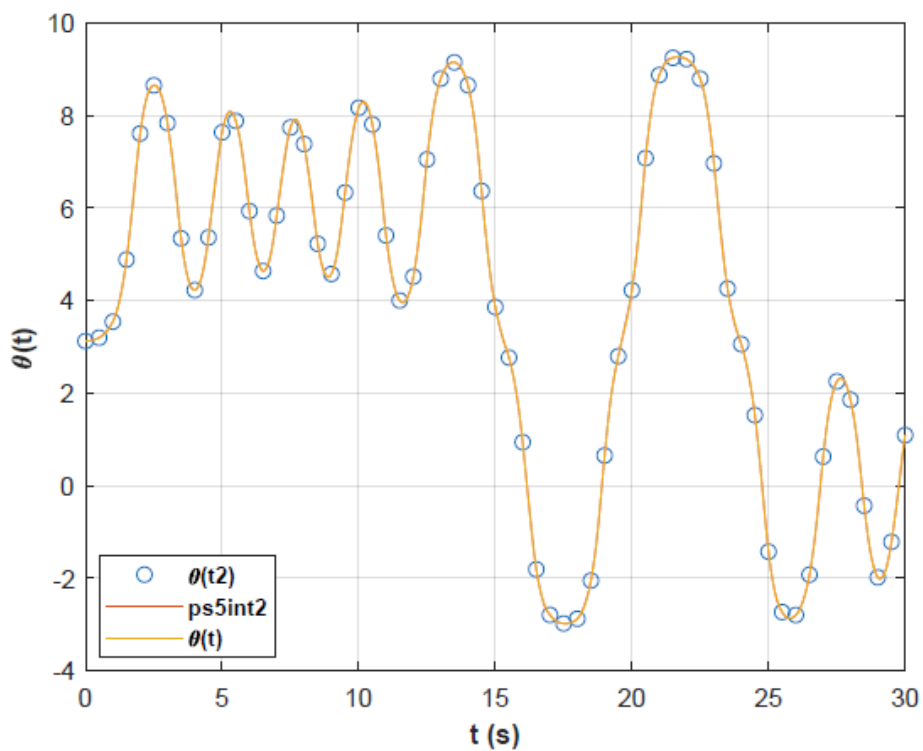


Figura 7.57. Solución continua del *PVI* (7.45) interpolada en los nodos de la malla t_2 por el *pseudo-spline* quintico “*ps5int2*”.

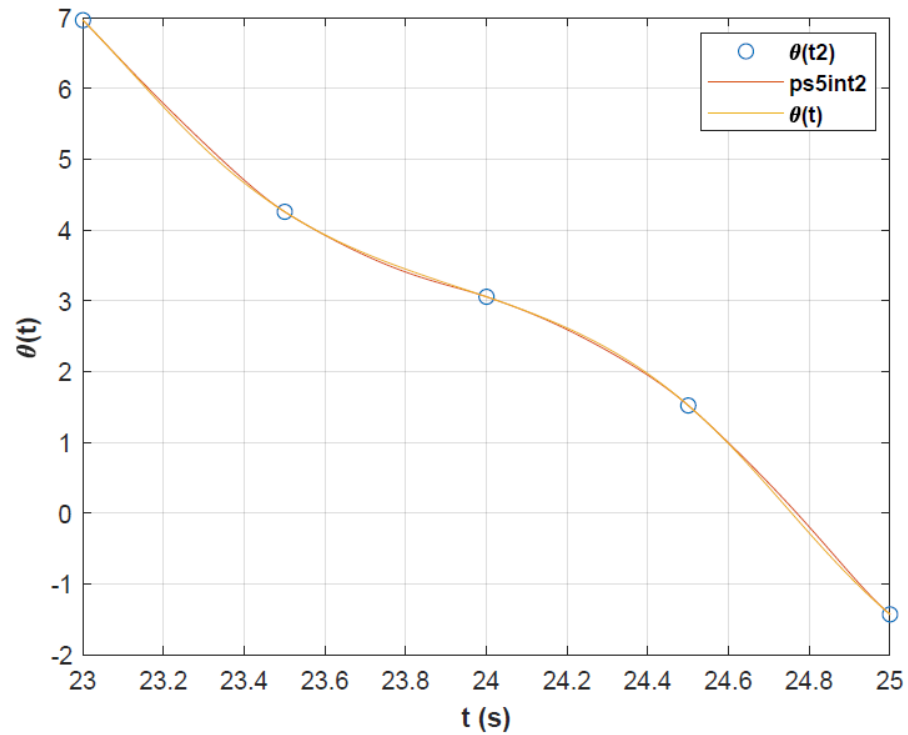


Figura 7.58. Zoom de la Figura 7.57 en [23,25] (s).

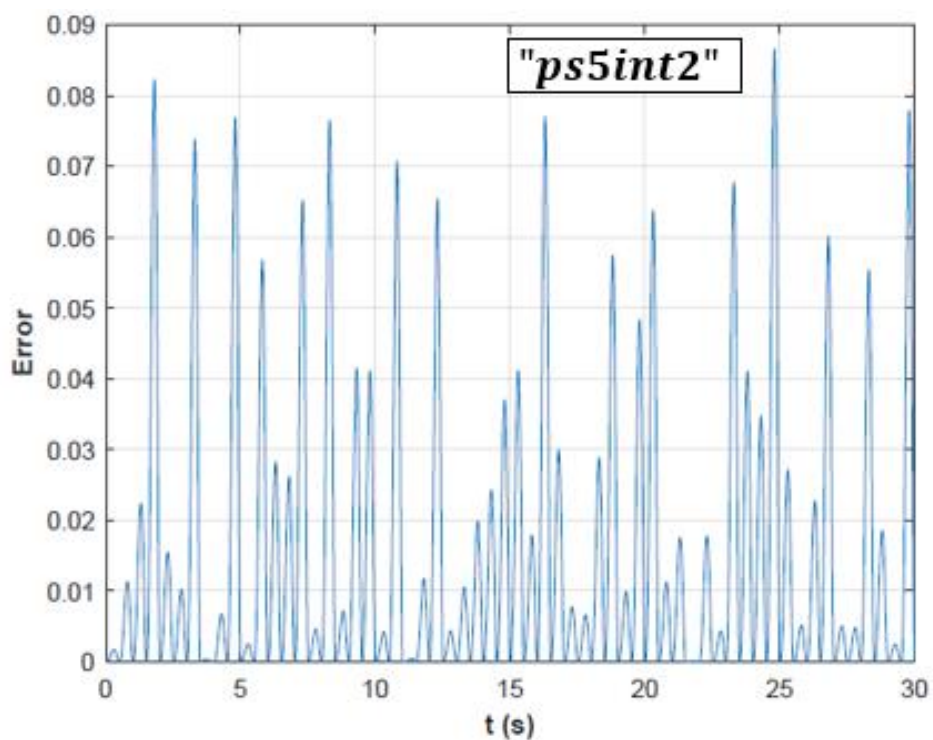


Figura 7.59. Error de interpolación cometido por el *pseudo-spline* quíntico “ps5int2” respecto de la solución discreta muy densificada a evitar computada con “ode89”.

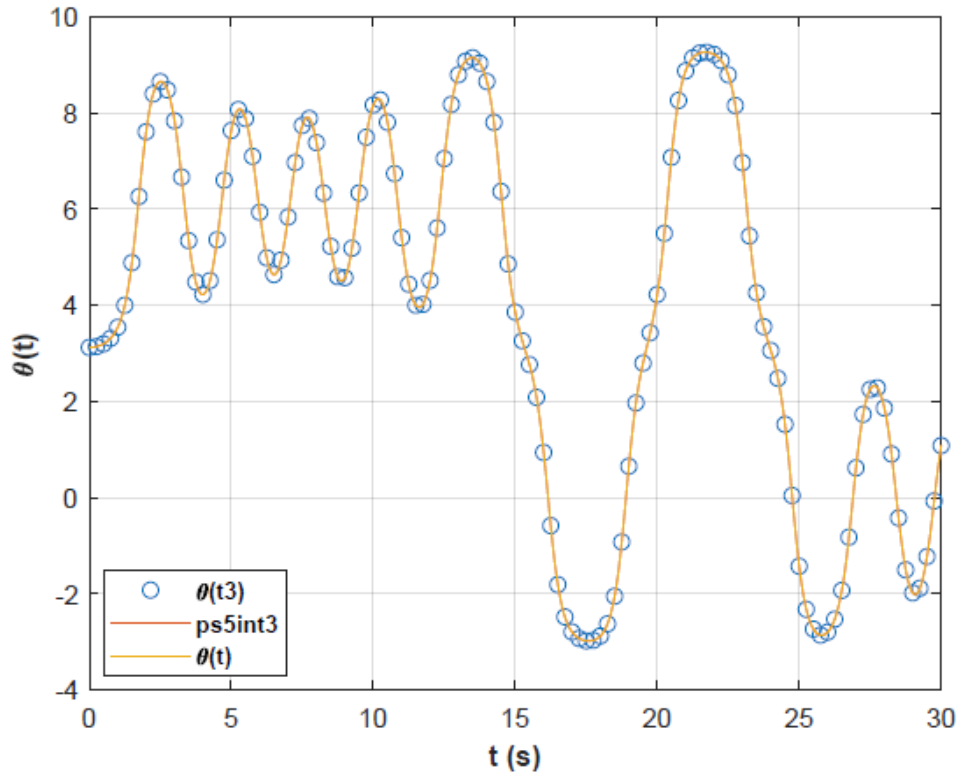


Figura 7.60. Solución continua del PVI (7.45) interpolada en los nodos de la malla t_3 por el pseudo-spline quintico "ps5int3".

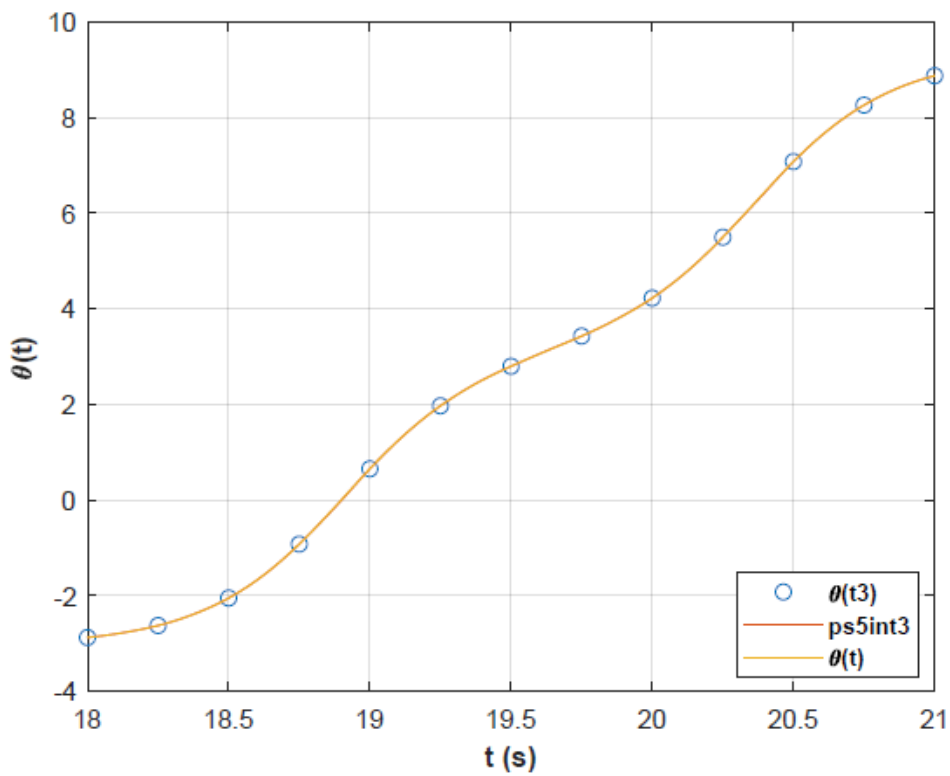


Figura 7.61. Zoom de la Figura 7.60 en $[18,21]$ (s).

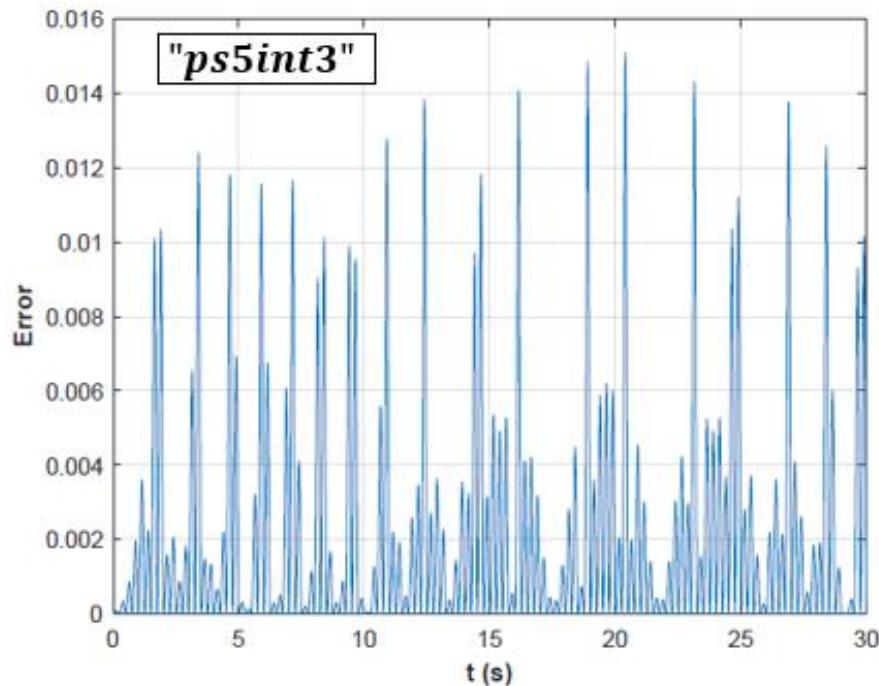


Figura 7.62. Error de interpolación cometido por el *pseudo-spline* quintico “ps5int3” respecto de la solución discreta muy densificada a evitar computada con “ode89”.

En contraposición con los resultados gráficos expuestos en 7.7.2.1 para los *splines* cuadráticos (ver *Tabla 7.46*), cabe destacar que ahora el comportamiento de los *pseudo-splines* quinticos es mucho mejor para todas las mallas de la *Tabla 7.42*, ya que estos imponen continuidad C^2 y se ajustan debidamente a los cambios de curvatura asociados a las oscilaciones de frecuencia variable. Se han aportado zooms para todos los trazadores que permiten corroborar este hecho en diversos subintervalos. Además, los errores de interpolación también son menores (mejorando prácticamente en un orden de magnitud con las mallas t_1 y t_2 , algo menos con t_3).

Retomando las tres primeras conclusiones de la sección 7.7.2.1 y haciéndolas aplicables a esta (las cuales eran únicamente cualitativas), se puede evaluar el error máximo cometido por cada trazador quintico, junto con el número de nodos en el que se ha de conocer la solución (mallas de ensayo). Si se compara éste último con, por ejemplo, el tamaño de la malla “t89m” que se construyó en 7.7.1 (*Tabla 7.40*), se tiene una estimación razonable de lo que ha disminuido el coste computacional (*Tabla 7.48*) asociado a la resolución del PVI (7.45) cuando se emplean estrategias de interpolación (*Capítulos 4 y 5*).

Malla	Nodos totales	Paso h	Error máx. quinticos	Nodo error máx.	% Ahorro eval. respecto “t89m”
t89m	929	Figura 7.44	-	-	-
t_1	31	1.00	3.97E-01	8.5760	96.66
t_2	61	0.50	8.66E-02	24.801	93.43
t_3	121	0.25	1.51E-02	20.400	86.98

Tabla 7.48. Evidencia de la reducción del coste computacional al emplear estrategias de interpolación para resolver de forma continua el PVI (7.45).



7.7.3. Conclusiones finales del experimento.

La realización de este extenso e ilustrativo experimento arroja una serie de **conclusiones generales importantes**, las cuales se deducen en concordancia con las observaciones realizadas en todas las secciones anteriores y las conclusiones ya aportadas en cada subapartado que las precisa. Por tanto, se puede destacar:

- i. La determinación de la solución aproximada del **PVI (7.45)** puede llevarse a cabo empleando únicamente métodos numéricos para la obtención de aproximaciones discretas en mallas muy densas (sección **7.1.1**), generalmente auto computadas según tolerancias de error, si se emplean *solvers* con pares encajados de métodos *Runge-Kutta* (marco teórico en **3.2.8** del **Capítulo 3**). Aunque la solución obtenida puede asumirse como continua al existir un número de nodos muy elevado (**Figura 7.42** y **Tabla 7.40** con tolerancias de *solvers* **(7.47)**), **no resulta ser una buena práctica por incrementar notablemente el coste computacional sin garantizar el éxito (Figura 7.40 con tolerancias insuficientes de solvers (7.46))**.
- ii. **Una alternativa eficiente y fiable** al procedimiento anterior consiste en emplear **estrategias de interpolación de soluciones basadas en trazadores polinomiales** (sección **7.7.2**) de construcción rápida y barata (en mallas equiespaciadas de la **Tabla 7.42**), los cuales **alivian el esfuerzo que un método discreto realiza** disminuyendo el número de nodos en los que han de computarse las soluciones discretas. Para no perder precisión, dicho método se procura escoger de alto orden ("**ode89**"), habiendo estudiado previamente las capacidades de cada uno de los candidatos (sección **7.1.1**, *solvers* tomados de la **Tabla 3.3** del **Capítulo 3**).
- iii. Una estrategia de interpolación de soluciones para **PVIs** de 2º orden (cualquiera dado por **(2.6)**, **(2.7)** y **(2.8)**) que involucre la **correcta imposición de condiciones de continuidad no sólo sobre la solución discreta, sino también sobre alguna (o ambas) de sus derivadas** (*pseudo-splines* quínticos, sección **7.7.2.2**, marco teórico en **5.3.3** del **Capítulo 5**) provoca unos **resultados muy fiables incluso empleando mallas equiespaciadas (Tabla 7.42) sin predicción alguna de las posibles oscilaciones y cambios de curvatura** a las que puede estar sometida la solución a determinar (caso particular de **(7.45)**). El empleo de trazadores polinomiales de menor regularidad (*splines* cuadráticos, sección **7.7.2.1**, marco teórico en **4.3** del **Capítulo 4**) no arroja resultados tan buenos, aunque reduce el coste computacional asociado a la resolución del **PVI** respecto al empleo de métodos discretos en mallas de nodos muy densas.
- iv. **Se puede demostrar cuantitativamente la reducción del coste computacional** sin más que tomar como referencia alguna de las mallas auto computadas por el *solver* escogido para asumir una solución

aproximada como exacta (malla “t89m” computada con “ode89”), calcular su tamaño y compararlo con aquellas mallas que, en combinación con interpolaciones posteriores (otra vez las de la **Tabla 7.42**), proporcionan una solución muy parecida con errores máximos no muy elevados (**Tabla 7.48**).

Una vez más se indica el interés notable por las estrategias de interpolación empleadas dualmente con métodos discretos, pues en aquellos problemas con evaluaciones muy complicadas o imposibles de la función del **PVI** en forma normal (algo ya comentado con anterioridad) evaluar un polinomio correctamente construido mediante interpolaciones respecto de datos, por ejemplo, empíricos, reduce notablemente la complejidad algorítmica.

El código de **MATLAB** “**PVI_Pendulo_No_Lineal**” necesario para la realización de este experimento numérico se incluye en el apartado correspondiente del **ANEXO IV**. No se ha mencionado, pero lógicamente las funciones que computan los *splines* y *pseudo-splines* vienen dadas en los **ANEXOS II** y **III**, respectivamente, en concordancia con lo expuesto en los **Capítulos 4** y **5**.

7.8. Experimento numérico 7. PVI rígido.

Para concluir el capítulo, se propone realizar un último experimento numérico cuyo punto de partida es el **PVI** de 2º orden rígido dado por (7.48), el cual es lineal del tipo (2.7) y homogéneo:

$$\begin{cases} y''(x) = -1001y'(x) - 1000.5y(x) \\ y(0) = 2, \quad y'(0) = 0 \\ x \in [0,7] \end{cases}$$

Ecuación (7.48)

Algunas consideraciones teóricas sobre las características de un problema rígido o *stiff* se comentan en la sección 3.2.9 del **Capítulo 3**. Los conocimientos pueden ser ampliados en [12] y [24]. El **PVI (7.48)** propuesto aquí tiene solución analítica y ésta se indica en (7.49):

$$y(x) = -\left(\frac{2s_2}{s_1 - s_2}\right)e^{(s_1x)} + \left(\frac{2s_1}{s_1 - s_2}\right)e^{(s_2x)} \rightarrow \begin{cases} s_1 = -\frac{1001}{2} + \frac{1}{2}\sqrt{997999} \\ s_2 = -\frac{1001}{2} - \frac{1}{2}\sqrt{997999} \end{cases}$$

Ecuación (7.49)

La naturaleza rígida de (7.48) puede explicarse en base a (7.49). La solución del **PVI** presenta términos transitorios del tipo e^{-ct} (tal y como se indica en 3.2.9), lo que a su vez implica la posibilidad de una variación muy pronunciada de la solución y de su primera derivada en zonas muy localizadas del dominio del problema. Gráficamente, se tiene la forma de $y(x)$ dada por la **Figura 7.63**:

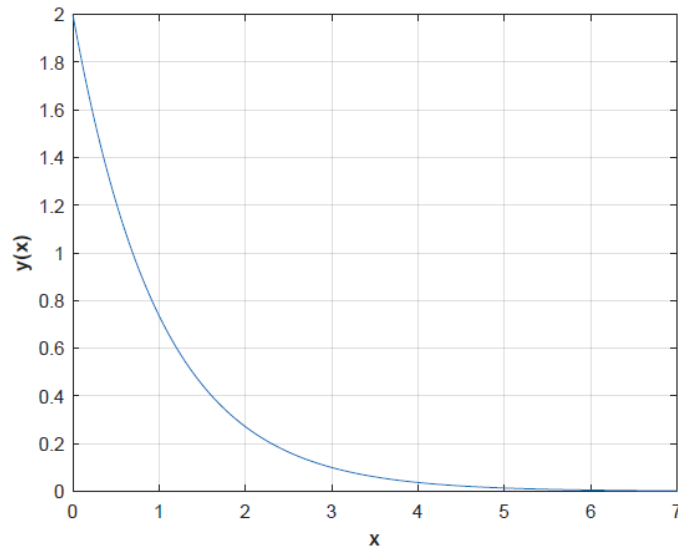


Figura 7.63. Solución exacta (7.49) del PVI (7.48).

La solución tiende asintóticamente a cero conforme avanza la variable independiente x . Al tratarse de una función exponencial como se indica en (7.49), la variación es muy rápida en el entorno de $x = 0$. El lector habrá podido percatarse de que la primera derivada varía de forma muy pronunciada, pues el cumplimiento de la segunda condición inicial de (7.48) no se refleja visualmente, pudiendo inducir a error. **Por ello, conviene indicar la forma que toma $y'(x)$, tanto analíticamente en (7.50) como gráficamente (Figura 7.64):**

$$y'(x) = -\left(\frac{2s_2s_1}{s_1 - s_2}\right)e^{(s_1x)} + \left(\frac{2s_1s_2}{s_1 - s_2}\right)e^{(s_2x)} \rightarrow \begin{cases} s_1 = -\frac{1001}{2} + \frac{1}{2}\sqrt{997999} \\ s_2 = -\frac{1001}{2} - \frac{1}{2}\sqrt{997999} \end{cases}$$

Ecuación (7.50)

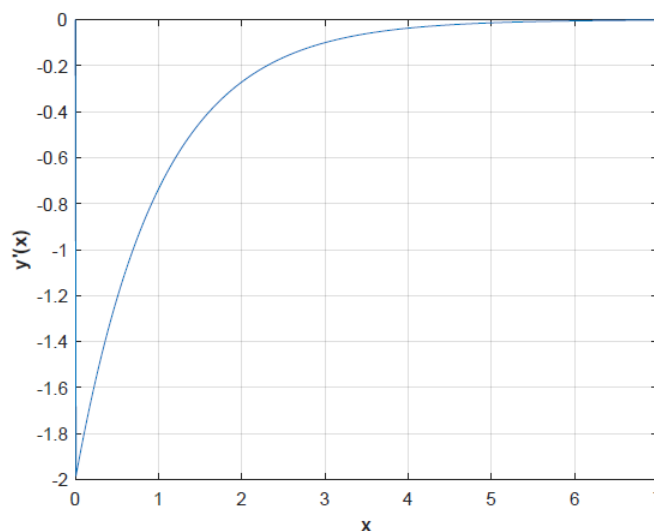


Figura 7.64. Primera derivada (7.50) de (7.49).

Haciendo un zoom (*Figura 7.65*) a la gráfica anterior en el intervalo $[0,0.02]$, se observa el cumplimiento de la segunda condición inicial de (7.48) y posterior caída rápidamente de la derivada, muy característica de un problema rígido monótonamente decreciente como el estudiado aquí.

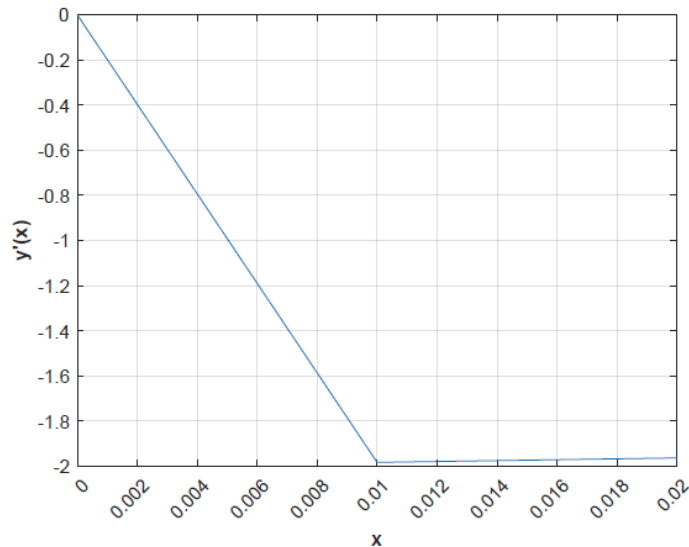


Figura 7.65. Zoom de la *Figura 7.64* en $[0,0.02]$.

7.8.1. Solución numérica discreta con pares encajados de métodos *Runge-Kutta*. Estudio del coste computacional.

El extenso experimento numérico 6 (sección 7.7) empleaba inicialmente *solvers* de *MATLAB* basados en métodos numéricos *Runge-Kutta* explícitos para la obtención de las soluciones discretas del *PVI* (7.45). Se utilizaban los *solvers* para problemas no rígidos expuestos en la *Tabla 3.3* del *Capítulo 3*. Posteriormente, se realizaba un estudio cuyo objetivo era determinar la reducción del coste computacional asociado al empleo de estrategias de interpolación de soluciones discretas en mallas poco densas obteniendo aproximaciones continuas a la solución del *PVI* (ver conclusiones en 7.7.3 para mayor información).

La idea aquí es similar, pero se parte de la gran ventaja de disponer de la solución exacta (7.49) de (7.48). No obstante, y teniendo en cuenta que se aborda un problema rígido, se propone utilizar inicialmente el *solver* “*ode23s*” de la *Tabla 3.4*, el cual es adecuado para abordar este tipo de problemas sin tener que recurrir a pasos ridículamente pequeños en el cómputo automático de la malla discreta no equiespaciada (2.11).

Por tanto, y empleando las tolerancias por defecto de *MATLAB* para la determinación del *ELT* (7.46), se computa la solución discreta aproximada con el *solver* citado, cuya forma viene dada por la *Figura 7.66*. Se emplea interpolación lineal por defecto para unir la malla de nodos generada “*x23s*”.

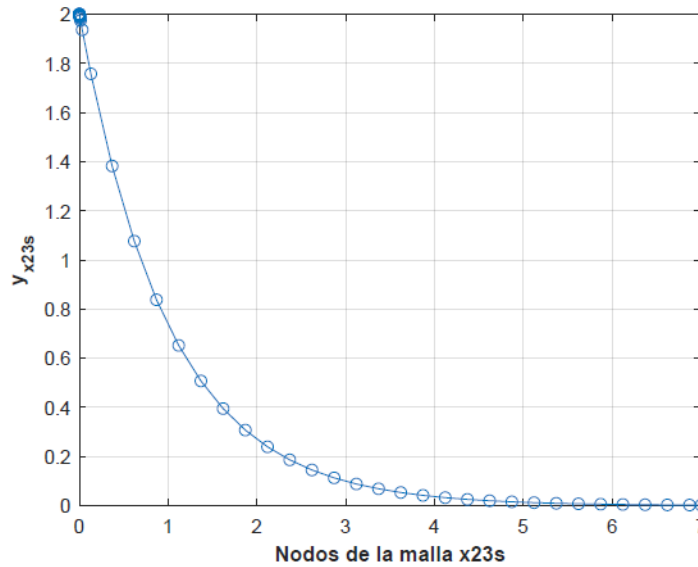


Figura 7.66. Solución discreta aproximada del PVI (7.48) en los nodos de la malla auto computada “x23s” mediante el empleo de “ode23s”.

Se aprecia un densificado grande en el entorno de $x = 0$, donde existe el conflicto entre la segunda condición inicial de (7.48) y la primera derivada en nodos cercanos al inicial del dominio (en virtud de lo observado en la Figura 7.65). Realizando un zoom a la figura anterior, surge la Figura 7.67:

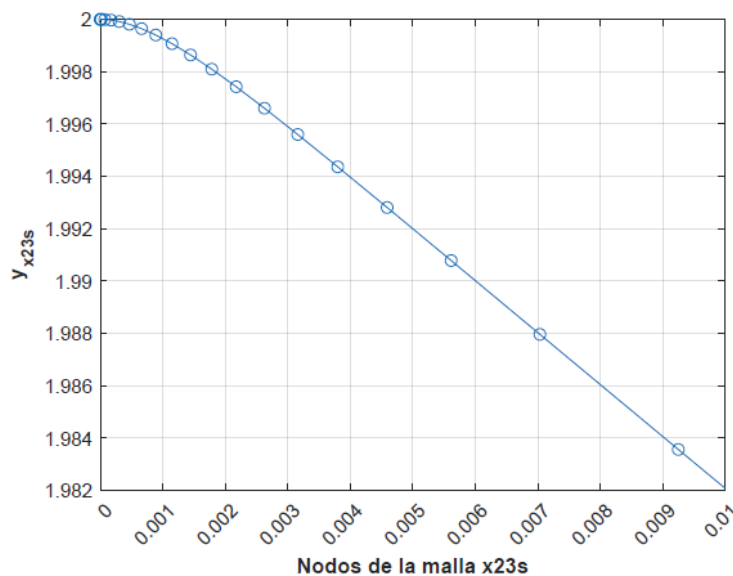


Figura 7.67. Zoom de la Figura 7.66 en $[0, 0.01]$.

Ahora, y conociendo la solución exacta (7.49), se puede obtener el EGT cometido por el método numérico aplicado. Nótese que esta situación es puramente didáctica e ideal (tal y como se ha indicado en anteriores experimentos), pues en la realidad es muy complicado conocer las soluciones analíticas de los problemas a abordar, razón de ser de los métodos numéricos aproximados. Representando el módulo del EGT, se tiene la Figura 7.68 (con zoom en el mismo intervalo que el de la Figura 7.67 en la Figura 7.69):

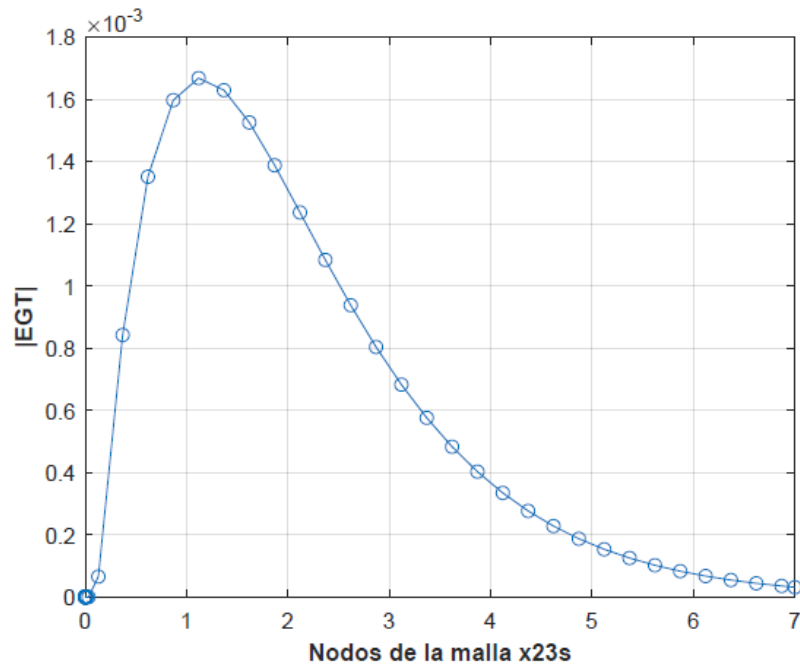


Figura 7.68. $|EGT|$ cometido por “ode23s” en la aproximación discreta de soluciones del PVI (7.48).

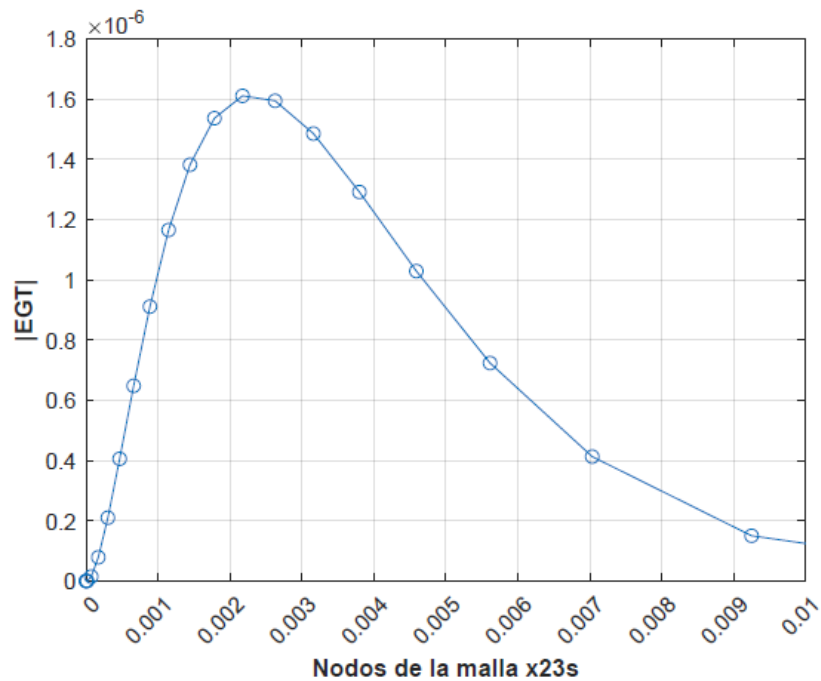


Figura 7.69. Zoom de la Figura 7.68 en $[0, 0.01]$.

El orden de magnitud máximo del EGT corresponde a 10^{-3} . Las estadísticas del mallado “x23s” auto computado por “ode23”, así como el nodo en el que se alcanza el máximo error se indican en la **Tabla 7.49**. La evolución del paso h_k (2.12) se da en la **Figura 7.70**.

Malla	Nodos totales	n23s	Paso h	$ EGT _{max}$	$x _{EGT _{max}}$
x23s	51	50	Figuras 7.70 y 7.71	1.668E-03	1.119

Tabla 7.49. Estadísticas principales de la malla “x23s” auto computada por “ode23s”.

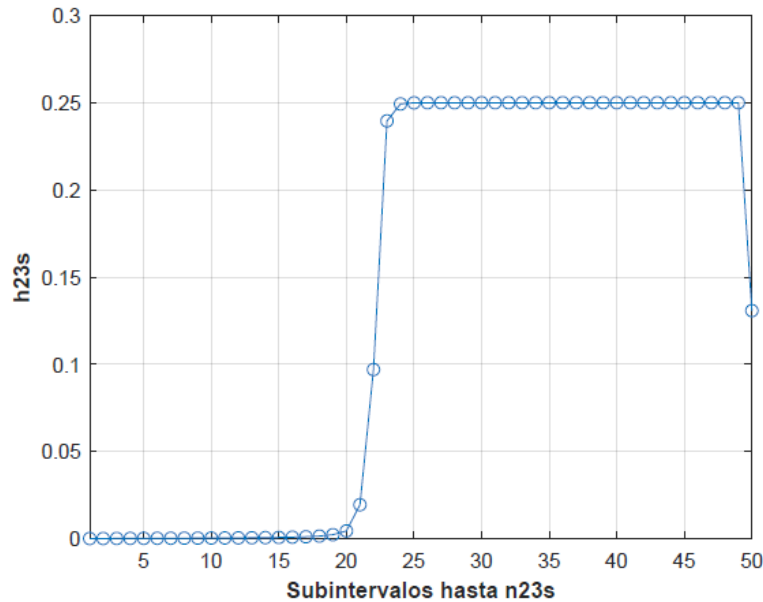


Figura 7.70. Evolución del paso “h23s” correspondiente a la malla “x23s”.

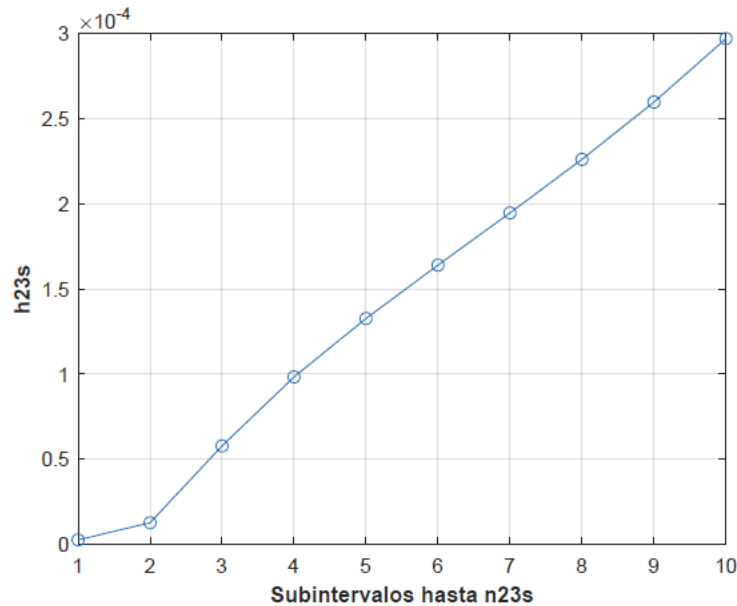


Figura 7.71. Zoom de la Figura 7.70.

En concordancia con las Figuras 7.66 y 7.67 se aprecia un densificado grande en el entorno del nodo inicial del PVI (7.48) $x = 0$, el cual permite una mejor aproximación ajustada a las tolerancias prefijadas por el solver y en presencia de la rigidez comentada al comienzo de este experimento (visible con claridad en la Figura 7.65).

Se concluye que, a pesar de que “*ode23s*” funciona bastante bien a la hora de aproximar soluciones discretas del *PVI* (7.48), éste no proporciona una aproximación continua, necesitando para ello el empleo de alguna estrategia de interpolación adicional que permita relajar el coste computacional asociado no sólo al empleo de una malla muy densa que pudiese asumirse como solución continua, sino también respecto a la malla “*x23s*”, la cual ha sido auto computada por el solver citado según las tolerancias por defecto para la evaluación del *ELT* en cada paso según (7.46).

7.8.2. Solución continua empleando *pseudo-splines* cúbicos. Monitorización del error y coste computacional.

En especial atención a la conclusión de la sección 7.8.1, se emplea a continuación un *pseudo-spline* cúbico interpolador (sección 5.3.2 del Capítulo 5) de las soluciones discretas del *PVI* (7.48) obtenidas con el solver “*ode23s*”. Como punto de partida a una posible reducción exitosa del coste computacional asociado a las evaluaciones de la función del *PVI* en forma normal precisadas por la malla “*x23s*”, considérese la malla equiespaciada a trozos “*xint*” y menos densa definida en la Tabla 7.50:

Subintervalo	n_i	Paso uniforme h_i
[0, 0.05)	10	0.005
[0.05, 2)	10	0.195
[2, 7]	5	1

Tabla 7.50. Malla equiespaciada a trozos “*xint*” que discretiza el dominio continuo del *PVI* (7.48).

Se realiza el densificado correspondiente pero menos pronunciado que en la malla “*x23s*” en el entorno de $x = 0$, con el objeto de proporcionar al trazador polinomial cúbico nodos suficientemente cercanos para que éste aproxime correctamente la tendencia de la derivada indicada en la Figura 7.65.

El *pseudo-spline* cúbico citado se elige debido a que permite la interpolación no sólo de la solución discreta conocida, sino también la de su primera derivada (información disponible y computada con “*ode23s*” o cualquier solver para *PVIs* de 2º orden). La importancia de una interpolación con derivadas es grande, demostrándose que su tendencia es determinante en el comportamiento rígido de la solución de (7.48).

Fijando como objetivo principal que el error de interpolación no sobrepase el máximo orden de magnitud del $|EGT|$ dado por la Tabla 7.49 y la Figura 7.68 (10^{-3}), se computan las nuevas soluciones en malla discreta “*xint*” empleando el solver “*ode23s*” y el correspondiente *pseudo-spline* cúbico interpolador continuo basado en polinomios interpoladores de *Hermite* (alias “*ps3int*”). El resultado gráfico es el siguiente (Figura 7.72, con zoom en el entorno de $x = 0$ dado por la Figura 7.73):

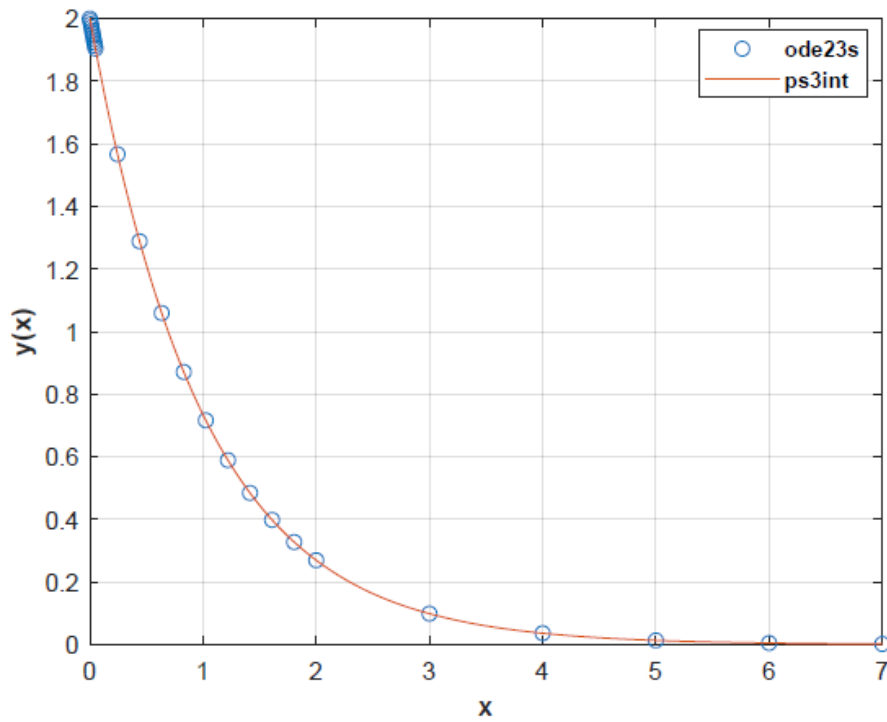


Figura 7.72. Solución continua aproximada del PVI (7.48) empleando el trazador “ps3int” y la malla “xint”.

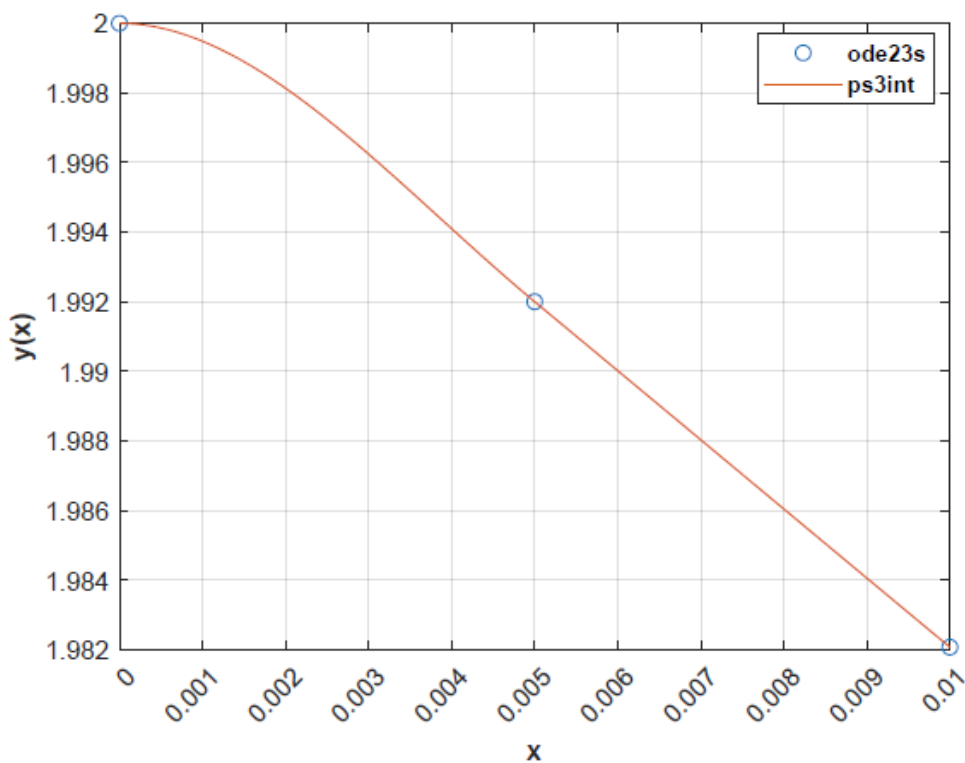


Figura 7.73. Zoom de la Figura 7.72 en $[0, 0.01]$.

Se aprecia cómo el trazador polinomial aproxima correctamente la tendencia de la derivada de la Figura 7.65 y verifica la segunda condición inicial del PVI

(7.48). Para enjuiciar esto último de forma más precisa, se obtiene el error asociado a la aproximación continua con “*ps3int*” y respecto a la solución exacta (7.49) (evaluando ahora en una malla auxiliar muy densa pero barata al emplear trazadores polinomiales cúbicos), en combinación con el nuevo [EGT] de “*ode23s*” en la nueva malla explícita indicada por la *Tabla 7.50* “*xint*”. Se tienen las *Figuras 7.74*, *7.75* y *7.76*:

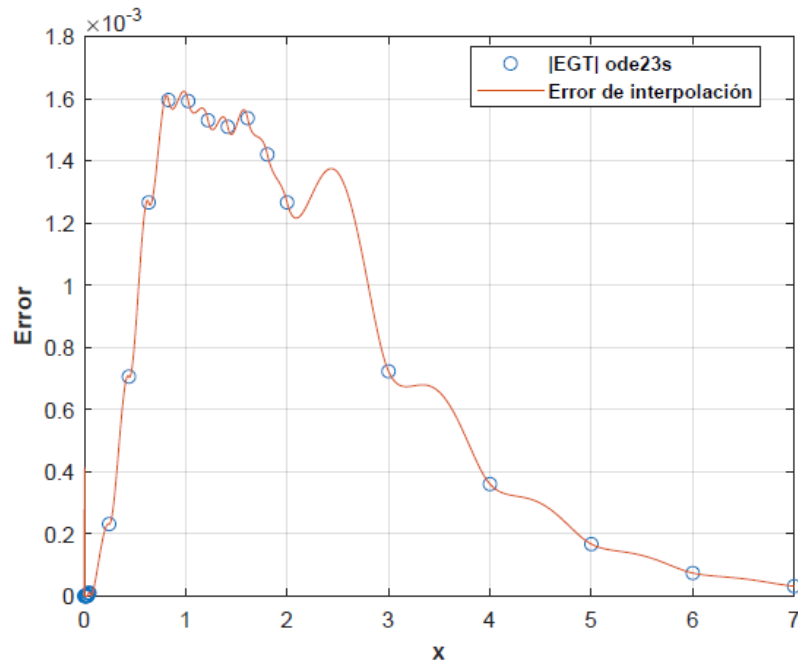


Figura 7.74. Error de interpolación correspondiente al trazador “*ps3int*” respecto de la solución exacta del PVI (7.48) dada por (7.49).

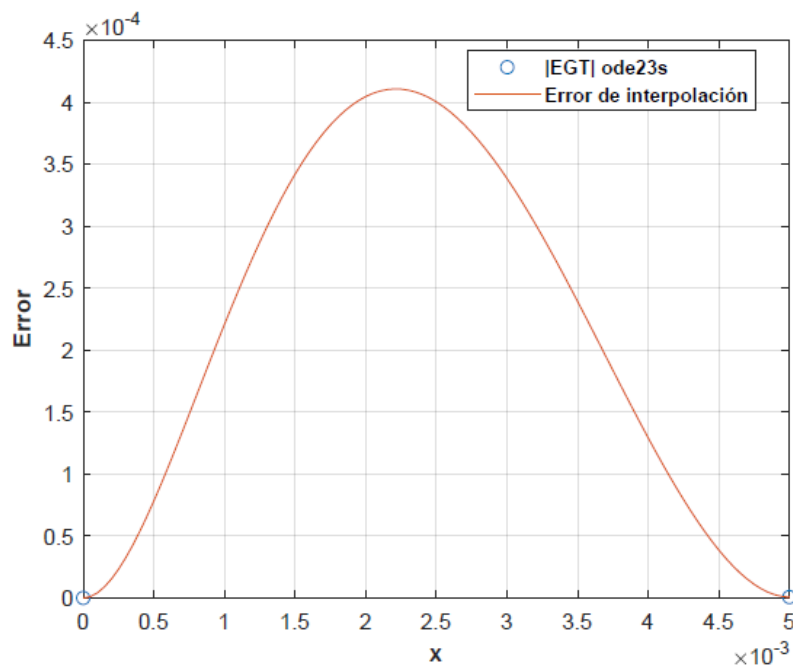


Figura 7.75. Zoom de la *Figura 7.74* en $[0, 0.005]$.

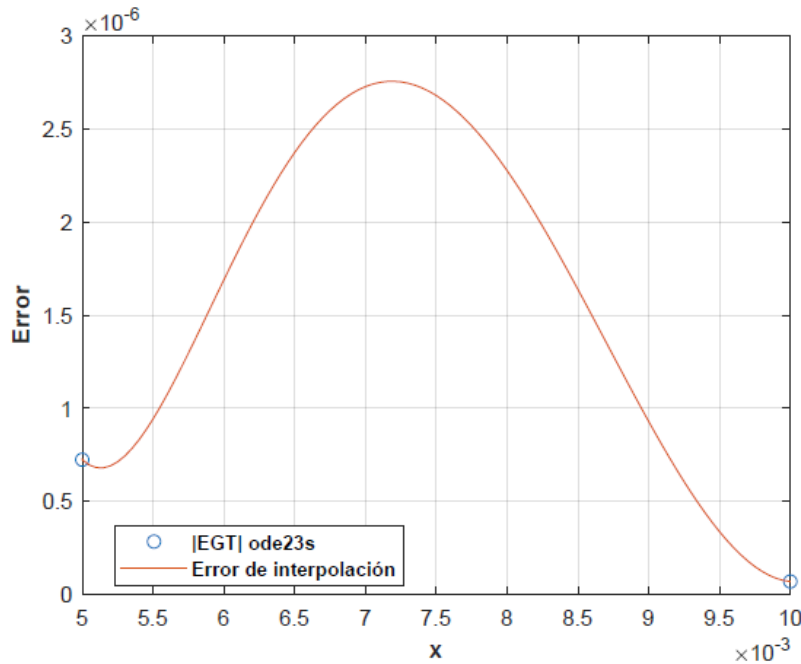


Figura 7.76. Zoom de la Figura 7.74 en $[0.005, 0.01]$.

A la vista de las figuras anteriores, se verifica que el orden de magnitud máximo del error cometido al interpolar y obtener la solución continua aproximada de (7.48) con “*ps3int*” es, efectivamente, 10^{-3} . En las zonas del dominio que muestran los zooms de las Figuras 7.75 y 7.76 también se aprecia el buen comportamiento del trazador cúbico respecto de la solución exacta (7.49), pues el orden de magnitud del error decrece respecto al máximo recientemente citado y presenta comportamiento estable al no amplificarse progresivamente.

En concordancia con los comentarios aportados en el párrafo anterior, se tienen las estadísticas de la Tabla 7.51. En esta tabla también se indica la reducción del coste computacional respecto al empleo único de una estrategia basada puramente en la aproximación discreta de soluciones con malla auto computada (“*x23s*”) con un par encajado de métodos *Runge-Kutta* (“*ode23s*”).

Malla	Nodos totales	n	Paso h	$ EGT _{max}$	$x_{ EGT _{max}}$
<i>xint</i>	26	25	Tabla 7.50	1.624E-03	0.982
<i>x23s</i>	51	50	Figuras 7.70 y 7.71	1.668E-03	1.119
% Ahorro coste	49.02%		% Incremento $ EGT _{max}$ (mismo orden)	2.64%	

Tabla 7.51. Estadísticas principales de la malla “*xint*” y ahorro de coste computacional respecto a la malla auto computada “*x23s*” y estrategia única discreta empleando “*ode23s*”.

7.8.3. Conclusiones del experimento.

La realización de este último experimento numérico arroja las siguientes conclusiones:

- i. El abordaje de un *PVI* rígido o *stiff* como el dado por (7.48) (con solución exacta (7.49)) implica el empleo de *solvers* y métodos numéricos con la capacidad de computar pasos, en ocasiones excesivamente pequeños, que permitan la correcta aproximación discreta de la solución en aquellas regiones en donde su variación es muy pronunciada, así como relajar el esfuerzo computacional en las otras zonas en donde la variación decrece progresivamente (asíntotas en el eje horizontal, por ejemplo).
- ii. Se tienen las mismas conclusiones que en el experimento numérico 6 (sección 7.7) pero con la ventaja de que ahora se han podido computar los errores respecto de la solución exacta (7.49) manteniendo la definición formal de la sección 3.2.2. Además, se podría haber utilizado cualquier otro *solver* de la Tabla 3.4 aparte de “*ode23s*”, incluso aquellos para *PVIs* no rígidos de la Tabla 3.3, con el problema de necesitar pasos ridículamente pequeños para cumplir las tolerancias de (7.46).
- iii. En el caso de haber identificado insatisfactoria o inexistente la reducción del coste computacional al emplear el *pseudo-spline* “*ps3int*” en combinación con las soluciones discretas aproximadas en malla “*xint*” obtenidas con “*ode23s*”, se podrían emplear estrategias de interpolación basadas en *pseudo-splines* quínticos (sección 5.3.3 del Capítulo 5), mayor densificado de “*xint*” respecto de la Tabla 7.50 o métodos numéricos discretos, por ejemplo implícitos de alto orden, si se demuestra que el problema viene dado por una mala aproximación al estar empleando pasos excesivamente altos para métodos de orden y precisión medios como el par encajado explícito utilizado (“*ode23s*”).

Por último, al igual que en el resto de experimentos, se recuerda que el código de *MATLAB* “*PVI_EDO_Stiff*” empleado para llevar a cabo este estudio puede consultarse en el ANEXO IV.





CAPÍTULO 8.

Conclusiones y líneas futuras de investigación.



8.1. Resultados y conclusiones generales.

Tras la finalización de los experimentos numéricos del **Capítulo 7**, algunos de ellos muy extensos, y teniendo en cuenta también el vasto marco teórico del que dichos experimentos beben y al que dan su aplicación directa, se pueden comentar los siguientes resultados globales concordantes con las conclusiones aportadas en cada sección del capítulo anterior:

- i. Todos los métodos numéricos de un paso para la obtención de soluciones discretas aproximadas de los **Problemas de Valores Iniciales (PVIs) de 2^o orden** definidos en el **Capítulo 2** y en los que se basan las posteriores estrategias de interpolación (**Capítulos 4 y 5**) que consiguen aproximaciones continuas son sometidos a experimentos debidamente diseñados que corroboran y demuestran su correcta implementación en **MATLAB (ANEXO I)**. Se obtienen órdenes de convergencia empíricos muy buenos.
- ii. Algunas de las estrategias de interpolación de las soluciones discretas de **PVIs** basadas en el empleo de *splines* cúbicos (sección **4.4** del **Capítulo 4**) y *pseudo-splines* cuárticos (sección **5.2** del **Capítulo 5**) resultan ser **inestables** utilizando la implementación basada en esquemas recursivos (códigos en **ANEXOS II y III**), por lo que es inviable en esta situación poderlas utilizar para computar soluciones aproximadas continuas con fiabilidad.
- iii. Las demás estrategias de interpolación utilizadas sí gozan de buenos resultados, tanto cualitativos como cuantitativos. Son especialmente interesantes aquellas basadas en polinomios osculadores (*pseudo-splines* cúbicos y quínticos), cuyo marco teórico puede consultarse en la sección **5.3** del **Capítulo 5**.
- iv. La experimentación concerniente a los **Problemas de Valores en la Frontera (PVFs) exclusivamente lineales** y su resolución mediante el método del disparo (expuesto teóricamente en el **Capítulo 6**) arroja resultados muy buenos en los problemas de Ingeniería propuestos. Este método es muy interesante por su facilidad de implementación y obtención de **PVIs** sencillos de resolver con los métodos expuestos en los **Capítulos 3, 4 y 5** (discretos y continuos).
- v. Las evidencias presentadas en los dos últimos experimentos numéricos (secciones **7.7 y 7.8** del **Capítulo 7**) refuerzan y complementan el marco teórico citado con anterioridad para todas las **estrategias de interpolación utilizadas dualmente con métodos discretos de un paso**, desembocando en resultados satisfactorios en cuanto a la reducción del coste computacional se refiere y permitiendo resolver **PVIs** más complicados. Además, también resulta interesante el empleo de *solvers* basados en pares encajados de métodos *Runge-Kutta* explícitos disponibles en **MATLAB**.



En atención a la síntesis de resultados aportada, y animando al lector a consultar las conclusiones individuales de cada experimento numérico llevado a cabo en el **Capítulo 7** (con códigos de **MATLAB** disponibles en el **ANEXO IV**), así como todo el marco teórico previo, surgen las siguientes conclusiones generales del trabajo:

- i. El empleo de métodos numéricos *Runge-Kutta-Nyström* (abreviadamente *RKN*) para la aproximación de soluciones discretas de **PVIs** exclusivamente de 2º orden permite una simplificación algorítmica sustancial respecto al empleo de métodos numéricos *Runge-Kutta* (*RK*), obteniéndose un orden de convergencia similar pero con un esquema de etapas más sencillo de entender e implementar (sección **3.4** del **Capítulo 3** y **ANEXO I**). Estos métodos suelen ser tratados teóricamente por multitud de investigadores (consúltense las referencias bibliográficas citadas) pero no es tan habitual su utilización práctica.
- ii. Emplear métodos numéricos de un paso (ya sean *RK* o *RKN*) explícitos puede acabar resultando insuficiente a efectos de obtener una aproximación discreta a la solución con la suficiente precisión requerida por el caso particular de estudio. Una solución es el **empleo de métodos implícitos**, los cuales no han sido abordados en profundidad, pero sí comentados y ensayados utilizando como ejemplo el método *Runge-Kutta* de *Gauss* (3 etapas y orden 6, sección **3.2.6** del **Capítulo 3**).
- iii. Cuando se utiliza un esquema recursivo como el que provoca el fallo (debido a su carácter inestable) de los *splines* cúbicos y *pseudo-splines* cuárticos interpoladores, conviene recurrir a *splines* de grado bajo (cuadráticos, sección **4.3** del **Capítulo 4**) incluso prescindiendo de la continuidad en la segunda derivada (trazadores polinomiales con continuidad C^1).
- iv. Las estrategias de interpolación de soluciones de **PVIs** de 2º orden basadas en *pseudo-splines* cúbicos y quínticos con continuidad C^2 son estables y convergen a la solución exacta del problema al que aproximan de forma continua conforme se refina el mallado en el que se obtienen las soluciones discretas a interpolar. Estas estrategias resultan ser muy interesantes cuando se plantean problemas del ámbito de la Ingeniería que requieren de aproximaciones certeras a segundas derivadas (aceleraciones), principal cuestión que suelen abordar los fenómenos de 2º orden formulados en términos de **EDOs**.
- v. Si bien los *pseudo-splines* cúbicos basados en polinomios interpoladores de *Hermite* son de conocimiento y aplicación genérica (además de los *splines* del **Capítulo 4**), otras estrategias de interpolación como los *pseudo-splines* quínticos no resultan ser tan comunes (sin muchas referencias bibliográficas disponibles), aportando este trabajo un esfuerzo investigador importante acerca de ellos (en su aplicación estricta a la aproximación e interpolación de soluciones del **PVI** de 2º orden).

- vi. Se consiguen estrategias de interpolación con la capacidad de desacoplar las no linealidades o elevadas complejidades de las funciones a evaluar en cada caso de la determinación unívoca de los trazadores polinomiales empleados en las aproximaciones de las soluciones. No obstante, se mantiene la naturaleza de los problemas a los que estas estrategias son aplicadas sin hacer ninguna hipótesis en el camino. **Esto reduce sustancialmente el coste computacional asociado a la resolución** de posibles problemas del ámbito ingenieril cuyo conocimiento del fenómeno se limite a datos puramente empíricos y de obtención muy compleja, ya que la evaluación de un *spline* o *pseudo-spline* definido a tramos por polinomios no resulta ser dificultoso en ninguna medida para un ordenador.
- vii. Los programas adjuntos de **MATLAB (ANEXOS I-V)** que implementan todas las técnicas y estrategias llevadas a cabo en este trabajo, así como los experimentos numéricos contextualizados con la Ingeniería Industrial, no requieren de un conocimiento muy avanzado de programación, permitiendo su ejecución y manejo fácilmente aprovechando el lenguaje de alto nivel que posee el software empleado.

En concordancia con las conclusiones expuestas, **se determina la consecución de todos los objetivos del Capítulo 1**, tanto en el desarrollo del marco teórico concerniente a los conocimientos de suma importancia que todo ingeniero ha de tener, como a las competencias adicionales surgidas de dicho marco y con aplicación directa en investigación y experimentación.

8.2. Líneas futuras de investigación.

Las líneas de continuación que este trabajo puede tener se citan a continuación:

- i. Completar la implementación de *splines* cúbicos y *pseudo-splines* cuárticos interpoladores mediante alguna estrategia alternativa a los esquemas recursivos que sea estable, bien mediante algún cambio de base, o algoritmos diversos que mantengan la naturaleza del marco teórico de los **Capítulos 4 y 5**.
- ii. Planteamiento de estrategias de resolución aproximadas basadas en métodos de colocación para **PVIs** de 2^º orden o incluso de orden superior, manteniendo las premisas en cuanto a la reducción del coste computacional. También es interesante la investigación relativa a las estrategias basadas en el método de mínimos cuadrados.
- iii. Utilización de métodos numéricos para la obtención de aproximaciones en malla discreta con especial énfasis a los métodos de un paso implícitos y a los métodos multipaso (algoritmos *Störmer*, por ejemplo). El aumento en la precisión que pueden proporcionar estos métodos puede relajar el esfuerzo computacional en problemas complejos y



depositar toda la confianza en estrategias de interpolación con su eficiencia ya demostrada en este trabajo.

- iv. Realización de un programa robusto en **MATLAB** (u otro lenguaje como *Python*) que implemente y mejore la interfaz para trabajar con todos los métodos que han sido planteados (así como los propuestos a mayores) como si fuese un paquete de cálculo comercial.



BIBLIOGRAFÍA.



Se indican a continuación todas las referencias bibliográficas consultadas en el desarrollo de este trabajo, numeradas en el mismo orden de su primera aparición en el texto.

[1] Gray, J. Change and Variations: A History of Differential Equations to 1900. (1ª ed.). Springer. (2021).

[2] Braun, M. Differential Equations and Their Applications. (4ª ed.). Springer. (1993).

[3] Logan, J.D. A First Course in Differential Equations. (3ª ed.). Springer. (2010).

[4] Archibald, T., Fraser, C. y Grattan-Guinness, I. The History of Differential Equations, 1670-1950. Oberwolfach Reports. Págs. 2729-2794. (2004).

[5] Lützen, J. The Mathematization Of The Physical Sciences - Differential Equations Of Nature. Encyclopedia of Life Support Systems (EOLSS). (2007).

[6] Wolf, T. y Brand. A. Examples of the Investigation of Differential Equations with Modularized Programs. Mathematical and Computer Modelling, vol. 25. Págs. 133-139. (1997).

[7] Cortés, J.C., Navarro-Quiles, A., Romero, J.V. y Roselló, M.D. Solving second-order linear differential equations with random analytic coefficients about ordinary points: A full probabilistic solution by the first probability density function. Applied Mathematics and Computation, vol. 331. Págs. 33-45. (2018).

[8] El-Sayed, A.M, Hashem, H. y Al-Issa, S.M. Study on the stability for implicit second-order differential equation via integral boundary conditions. Journal of Mathematical Modeling, vol. 10. Págs 331-348. (2022).

[9] Santra, S.S, Bazighifan, O., Ahmad, H. y Chu. Y.M. Second-Order Differential Equation: Oscillation Theorems and Applications. Mathematical Problems in Engineering. (2020).

[10] Fernández, F.J, Márquez, I. y Tojo, A. On first and second order linear Stieltjes differential equations. Journal of Mathematical Analysis and Applications, vol. 511. (2022).

[11] "A Brief History of MATLAB":
<https://es.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.html>. (Último acceso: 14 de julio de 2022).

[12] Burden, R.L., Faires, D.J., y Burden, A.M. Análisis numérico. (10ª ed.). Cengage Learning. (2017).

[13] *Runge-Kutta Method - An overview*:
<https://www.sciencedirect.com/topics/mathematics/runge-kutta>

[method#:~:text=Runge%E2%80%93Kutta%20method%20is%20an,high%20order%20derivatives%20of%20functions](#) (Último acceso: 14 de julio de 2022).

[14] Rabiél, F., Senu, N., Ismail, F. y Abasi, N. Construction of Improved *Runge-Kutta Nyström* method for Solving-Second Order Ordinary Differential Equations. *World Applied Sciences Journal*, vol. 20. Págs. 1685-1695. (2012).

[15] Duruflé, M. y N'Diaye, M. Optimized High Order Explicit *Runge-Kutta-Nyström* Schemes. *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016 Selected Papers from the ICOSAHOM conference*. (2016).

[16] Álvarez, J. Métodos GRK para ecuaciones diferenciales ordinarias. Tesis doctoral, Universidad de Valladolid. (2002).

[17] Hairer, E., Nørsett, S. y Wanner, G. Solving Ordinary Differential Equations I: Nonstiff Problems. (2ª ed.). Springer Series in Computational Mathematics. (2008).

[18] Dormand, J.R. Numerical Methods for Differential Equations. A computational Approach. (1ª ed.). CRC Press. Taylor & Francis Group. (1996).

[19] Butcher, J.C. Numerical Methods for Ordinary Differential Equations. (3ª ed.). John Wiley & Sons. (2016).

[20] Edwards, C.H. y Penney, D.E. Ecuaciones diferenciales y Problemas con Valores en la Frontera. *Cómputo y Modelado*. (4º ed.). Pearson Prentice Hall. (2009).

[21] Granados, A. Curso sobre métodos numéricos: Un curso introductorio para ingenieros y científicos. Editorial Académica Española. (2016).

[22] Teherán, J.D. Formulación, deducción e implementación de métodos Runge-Kutta-Nyström implícitos de alto orden para sistemas de ecuaciones diferenciales ordinarias de segundo orden. Tesis doctoral, Universidad del Norte. (2020).

[23] Dormand, J.R. y Prince, P.J. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, vol. 6. Págs. 19-26. (1980).

[24] Hairer, E. y Wanner, G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. (2ª ed.). Springer Series in Computational Mathematics. (1996).

[25] MATLAB “Choose an ODE Solver”. Mathworks. | <https://es.mathworks.com/help/matlab/math/choose-an-ode-solver.html>. (Último acceso: 14 de julio de 2022).

- [26] Young Yang, W., Chung, T., Cao, W. y Morris, J. Applied numerical methods using MATLAB. John Wiley & Sons. (2005).
- [27] Bogacki, P. y Shampine, L.F. A 3(2) pair of Runge-Kutta formulas. Applied Mathematics Letters, vol. 2. Págs. 321-325. (1989).
- [28] Verner, J.H. Numerically Optimal Runge-Kutta Pairs with Interpolants. Numerical Algorithms, vol 53. Págs. 383-396. (2010).
- [29] Shampine, L.F. y Reichelt, M.W. The MATLAB ODE Suite. SIAM Journal on Scientific Computing, vol 18. Págs. 1-22. (1997).
- [30] Shampine, L.F., Reichlet, M.W. y Kierzenka, J.A. Solving Index-1 DAEs in MATLAB and Simulink. SIAM Review, vol. 41. Págs. 538-552. (1999).
- [31] Shampine, L.F. y Hosea, M.E. Analysis and Implementation of TR-BDF2. Applied Numerical Mathematics, vol 20. (1996).
- [32] Garrosa, A.G. Métodos numéricos tipo Runge-Kutta-Nyström para la integración eficiente de problemas oscilatorios. Tesis doctoral, Universidad de Valladolid. (2001).
- [33] Kilicman, A., Srivastava, H.M, Mursaleen, M y Majid, Z.M. Recent Advances in Mathematical Sciences: Selected Papers from ICREM7 2015. Springer Singapore. (2016).
- [34] Wu, X., You., X. y Wang, B. Structure-Preserving Algorithms for Oscillatory Differential Equations. SpringerLink: Bücher, Springer Berlin Heidelberg. (2013).
- [35] de Iaco Veris, A. Practical Astrodynamics. Springer Aerospace Technology, Springer International Publishing. (2017).
- [36] Johnson, S.A. Splines. In: Gass S.I., Fu, M.C. (eds) Encyclopedia of Operations Research and Management Science. Springer. (2013).
- [37] Young Kinh-Nhue, T. y Sarfraz M. Topics in Splines and Applications. Intechopen. (2018).
- [38] Philips, G.M. Interpolation and Approximation by Polynomials. (2003).
- [39] Li, X. y Sederberg, W. S-splines: A simple surface solution for IGA and CAD. Computer Methos in Applied Mechanics and Engineering, vol. 350. Págs. 664-678. (2019).
- [40] Mayo, J.A., Stevenson, D.K. y Shaw, G.M. Population-based associations between maternal pre-pregnancy body mass index and spontaneous and medically indicated preterm birth using restricted cubic splines in California. Annals of Epidemiology, vol. 72. Págs. 65-73. (2022).

- [41] Gazi, S., Saha, A. y Sucu, D.Y. A collocation algorithm based on septic B-splines and bifurcation of traveling waves for Sawada-Kotera Equation. *Mathematics and Computers in Simulation*. (2022).
- [42] Luo, J., Liu, Q., Zhou, Z. y Wei, Q. The temperature field modeling for metal plate surface based on PSO-B-spline interpolation and FBG sensors. *Measurement*, vol. 159. (2020).
- [43] Dong, B. Pseudo-splines, wavelets and framelets. *Applied and Computational Harmonic Analysis*, vol. 22. Págs. 78-104. (2007).
- [44] Li, S. y Shen, Y. Pseudo box splines. *Applied and Computational Harmonic Analysis*, vol 26. Págs. 344-356. (2009).
- [45] Zhuang, Z. y Yang, J. A class of generalized pseudo-splines. *Journal of Inequalities and Applications*. (2014).
- [46] Gautschi, W. *Numerical Analysis*. (2^a ed.). Birkhäuser. (2012).
- [47] Stoer, J. y Bulirsch, R. *Introduction to Numerical Analysis*. (1^a ed.). Springer. (1980).
- [48] Hamming, R.W. *Numerical Methods for Scientists and Engineers*. (2^a ed.). Dover Publications. (1973).
- [50] Lamnii, A., Lamnii, M. y Oumellal, F. A new basis for osculatory interpolation problems and applications. *Applied Mathematics and Computation*, vol 283. Págs 355-368. (2016).
- [51] Sakai, M. Osculatory interpolation. *Computer Aided Geometric Design*, vol 18. Págs. 739-750. (2001).
- [52] Keller, H.B. *Numerical Solution of Two Point Boundary Value Problems*. (4^a ed.). Society for Industrial and Applied Mathematics. (1990).
- [53] Trench, W.F. *Elementary Differential Equations with Boundary Value Problems*. Trinity University (San Antonio, Texas, USA). (2013).
- [54] Auzinger, W., Burkotová, J., Rachundová, I. y Wenin, V. Shooting methods for state-dependent impulsive boundary value problems, with applications. *Applied Numerical Mathematics*, vol. 128. Págs. 217-229. (2018).
- [55] Filipov, S.M., Gospodinov, I.V. y Faragó, I. Replacing the finite difference methods for nonlinear two-point boundary value problems by successive application of the linear shooting method. *Journal of Computational and Applied Mathematics*, vol. 358. Págs. 46-60. (2019).
- [56] Attili, B.S. y Syam, M.I. Efficient shooting method for solving two point boundary value problems. *Chaos, Solitons and Fractals*, vol. 35. Págs. 895-903. (2008).

[57] Salahuddin, T. Carreau fluid model towards a stretching cylinder: Using Keller box and shooting method. *Ain Shams Engineering Journal*, vol. 11. Págs. 495-500. (2020).

[58] Puente, R. Estrategias de resolución de problemas de Ingeniería formulados en términos de EDOs de 2º orden. TFG, Universidad de Valladolid. (2017).

[59] Weaver, W., Timoshenko, S.P. y Young D.H. *Vibration problems in engineering*. (5ª ed). John Wiley & Sons. (1990).

[60] Mass-Spring System – an overview | ScienceDirect Topics. <https://www.sciencedirect.com/topics/engineering/mass-spring-system>. (Último acceso: 14 de julio de 2022).

[61] Aoki, T., Yamashita, Y. y Tsubakino, D. Vibration Suppresion of Mass-Spring-Damper System with Dynamic Dampers using IDA-PBC. 4th IFAC Workshop on Lagrangian and Hamiltonian Methods for Non Linear Control. International Federation of Automatic Control. (2012).

[62] Shahabpoor, E., Pavic, A. y Racic V. Identification of mass-spring-damper model of walking humans. *Structures*, vol. 5. Págs. 233-246. (2016).

[63] Cai, Y., Chen, L., Yu, W., Zhou, J., Wan, F., Suh, M. y Hung-kay Chow, D. A piecewise mass-spring-damper model of the human breast. *Journal of Biomechanics*, vol. 67. Págs. 137-143. (2018).

[64] Zhang, H., Ni, H., Wang, Z., Huang., B., Liu, S., Xu., X. y Liu, C. Discrete element modeling and simulation study on cutting rock behavior under spring-mass-damper system loading. *Journal of Petroleum Science and Engineering*, vol 209. (2022).

[65] Fung, Y.C. *Foundations of Solid Mechanics* (International Series in Dynamics) (1ª ed.). Prentice Hall. (1965).

[66] Rabascall, D. Análisis de las respuestas de elementos a flexión con grandes desplazamientos ante cargas seguidoras (Trabajo Fin de Máster). Universidad Politécnica de Madrid. (2011).

[67] Bathe, K.J. y Bolourchi, S. Large displacement analysis of three-dimensional beam structures. *International Journal for Numerical Methods in Engineering*, vol. 14. Págs. 961-986. (1979).

[68] Lienhard IV, J.H. y Lienhard V, J.H. *A Heat Transfer Textbook* (5ª ed.). Capítulo 4. Phlogiston Press. (2020).

[69] Gardner, K.A. Efficiency of extended surfaces. *Journal of heat transfer*, *Trans. ASME*, vol. 67. Págs. 621-631. (1945).



[70] Arauzo, I., Campo, A. y Cortés, C. Quick estimate of the heat transfer characteristics of annular fins of hyperbolic profile with the power series method. *Applied Thermal Engineering*, vol 25. Págs. 623-634. (2005).

[71] Darvishi, M.T., Khani, F y Aziz, A. Numerical investigation for a hyperbolic annular fin with temperature dependent thermal conductivity. *Propulsion and Power Research*, vol 5. Págs. 55-62. (2016).



ANEXOS A LA MEMORIA



ANEXO I. Métodos numéricos discretos de un paso.

En este anexo se incluyen todos los métodos numéricos *Runge-Kutta* y *Runge-Kutta-Nyström* para la obtención de soluciones discretas aproximadas de Problemas de Valores Iniciales de 2º orden expuestos en el **Capítulo 3** de la memoria e implementados mediante funciones en **MATLAB**.

- **Método de Euler explícito.**

```
function [y,z] = ERK1s1o(x,f,y0,z0)
%-----
% La función "ERK1s1o" implementa el método de Euler explícito (método de
% las tangentes) de primer orden como un método Runge-Kutta explícito de 1º
% orden para obtener la solución discreta aproximada y del PVI genérico de 2º
% orden (2.6) y su primera derivada z mediante la resolución del sistema de
% EDOS equivalente (3.12) en el que se puede reescribir dicho PVI. Las
% consideraciones sobre este método numérico, así como su Tablero de Butcher
% asociado pueden consultarse en la sección 3.2.3.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI ( ) para poder computar las etapas del
% método de Euler explícito de 1º orden. Esta función debe de ser
% pasada como input de la siguiente manera empleando el "function_handle":
% f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)
% NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.6)
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI ( ) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
(2.6) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
aproximada z del PVI (2.6) en x:
z = zeros(n+1,1);
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
nodos de malla x:
```



```

for i = 1:n
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*z(i);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*f(x(i),y(i),z(i));
end
end

```

- **Método de Heun de orden 3.**

```

function [y,z] = HRK3s1o(x,f,y0,z0)
%-----
% La función "HRK3s1o" implementa el método Runge-Kutta explícito de Heun
% de 3º orden para obtener la solución discreta aproximada y del PVI genérico
% de 2º orden (2.6) y su primera derivada z mediante la resolución del sistema
% de EDOS equivalente (3.12) en el que se puede reescribir dicho PVI. Las
% consideraciones sobre este método numérico, así como su Tablero de Butcher
% asociado pueden consultarse en 3.2.4.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI (2.6) para poder computar las etapas del
% método numérico Runge-Kutta de Heun y 3º orden. Esta función debe de ser
% pasada como input de la siguiente manera empleando el "function_handle":
%     f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)
% NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.6)
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI (2.6) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.6) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.6) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta de Heun de 3º orden:
k11 = 0; k21 = 0;
k12 = 0; k22 = 0;
k13 = 0; k23 = 0;
% Imposición de las condiciones iniciales:

```



```
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    K11 = z(i);
    K21 = f(x(i),y(i),z(i));
    K12 = z(i)+1/3*h(i)*K21;
    K22 = f(x(i)+1/3*h(i),y(i)+1/3*h(i)*K11,z(i)+1/3*h(i)*K21);
    K13 = z(i)+2/3*h(i)*K22;
    K23 = f(x(i)+2/3*h(i),y(i)+2/3*h(i)*K12,z(i)+2/3*h(i)*K22);
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*(1/4*K11+3/4*K13);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(1/4*K21+3/4*K23);
end
end
```

- **Método Runge-Kutta explícito de orden 4.**

```
function [y,z] = RK4s1o(x,f,y0,z0)
%-----
% La función "RK4s1o" implementa el método Runge-Kutta explícito
% de 4º orden para obtener la solución discreta aproximada y del PVI genérico
% de 2º orden (2.6) y su primera derivada z mediante la resolución del sistema
% de EDOs equivalente (3.12) en el que se puede reescribir dicho PVI. Las
% consideraciones sobre este método numérico, así como su Tablero de Butcher
% asociado pueden consultarse en 3.2.5.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI (2.6) para poder computar las etapas del
% método numérico Runge-Kutta de 4º orden. Esta función debe de ser
% pasada como input de la siguiente manera empleando el "function_handle":
%     f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)
% NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.6)
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI (2.6) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.6) en x:
```



```

y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.6) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta de 4º orden:
K11 = 0; K21 = 0;
K12 = 0; K22 = 0;
K13 = 0; K23 = 0;
K14 = 0; K24 = 0;
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    K11 = z(i);
    K21 = f(x(i),y(i),z(i));
    K12 = z(i)+1/2*h(i)*K21;
    K22 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*K11,z(i)+1/2*h(i)*K21);
    K13 = z(i)+1/2*h(i)*K22;
    K23 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*K12,z(i)+1/2*h(i)*K22);
    K14 = z(i)+h(i)*K23;
    K24 = f(x(i)+h(i),y(i)+h(i)*K13,z(i)+h(i)*K23);
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*(1/6*K11+1/3*K12+1/3*K13+1/6*K14);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(1/6*K21+1/3*K22+1/3*K23+1/6*K24);
end
end

```

- **Método de Gauss implícito de 3 etapas y orden 6.**

```

function [y,z] = GRK6s1o(x,f,y0,z0)
%-----
% La función "GRKs1o" implementa el método Runge-Kutta de Gauss implícito
% de 6º orden y 3 etapas para obtener la solución discreta aproximada y del
% PVI genérico de 2º orden (2.6) y su primera derivada z mediante la
% resolución del sistema de EDos equivalente (3.12) en el que se puede
% reescribir dicho PVI. Las consideraciones sobre este método numérico, así
% como su Tablero de Butcher asociado pueden consultarse en 3.2.6.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI (2.6) para poder computar las etapas del
% método numérico .... Esta función debe de ser
% pasada como input de la siguiente manera empleando el "function_handle":
% f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)
% NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.6)
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:

```

```
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI (2.6) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.6) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.6) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta de Gauss de 6º orden:
K = ones(6,1);
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    options = optimset('Display','off');
    K = fsolve(@(K) root6D_GRK6s1o(f,x(i),y(i),z(i),h(i),K),
zeros(1,6),options);
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*(5/18*K(1)+4/9*K(2)+5/18*K(3));
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(5/18*K(4)+4/9*K(5)+5/18*K(6));
end
end
```

- **Función "root6D GRK6s1o"**

```
function F = root6D_GRK6s1o(f,x,y,z,h,K)
%-----
% La función "root6D_GRK6s1o" monta el sistema no lineal correspondiente a
% las etapas en cada paso del método de Gauss de orden 6 y 3 etapas que
% resuelve de forma aproximada el PVI de 2º orden genérico dado por (2.6).
% Posteriormente, este sistema deberá de ser resuelto mediante la
% utilización de algún algoritmo de resolución numérica de sistemas no
% lineales ("fsolve", por ejemplo).
%
% ENTRADA DE LA FUNCIÓN:
%
% x = Nodo correspondiente al paso anterior de la malla discreta en la que
% se está aplicando el método numérico
% y = Valor de la solución discreta aproximada obtenida en el paso
% anterior
% z = Valor de la primera derivada de la solución discreta aproximada
% obtenida en el paso anterior
% h = Paso inmediatamente anterior del método numérico
% K = Vector fila o columna que contiene las etapas correspondientes al
```



```
% paso anterior del método numérico
%
% SALIDA DE LA FUNCIÓN:
%
% [F] = vector columna que contiene el sistema no lineal listo para su
% resolución numérica
%-----
F = zeros(6,1);
F(1) = z+h*(5/36*k(4)+(10-3*sqrt(15))/45*k(5)+(25-6*sqrt(15))/180*k(6))-k(1);
F(2) = f(x+(5-sqrt(15))/10*h,y+h*(5/36*k(1)+(10-3*sqrt(15))/45*k(2)+(25-
6*sqrt(15))/180*k(3)),z+h*(5/36*k(4)+(10-3*sqrt(15))/45*k(5)+(25-
6*sqrt(15))/180*k(6)))-k(4);
F(3) = z+h*((10+3*sqrt(15))/72*k(4)+2/9*k(5)+(10-3*sqrt(15))/72*k(6))-k(2);
F(4) = f(x+1/2*h,y+h*((10+3*sqrt(15))/72*k(1)+2/9*k(2)+(10-
3*sqrt(15))/72*k(3)),z+h*((10+3*sqrt(15))/72*k(4)+2/9*k(5)+(10-
3*sqrt(15))/72*k(6)))-k(5);
F(5) = z+h*((25+6*sqrt(15))/180*k(4)+(10+3*sqrt(15))/45*k(5)+5/36*k(6))-k(3);
F(6) =
f(x+(5+sqrt(15))/10*h,y+h*((25+6*sqrt(15))/180*k(1)+(10+3*sqrt(15))/45*k(2)+5/
36*k(3)),z+h*((25+6*sqrt(15))/180*k(4)+(10+3*sqrt(15))/45*k(5)+5/36*k(6)))-
k(6);
end
```

- **Método RKN directo de orden 3.**

```
function [y,z] = RKN3(x,f,y0,z0)
%-----
% La función "RKN3" implementa el método Runge-Kutta-Nyström explícito de 3º
% orden y 3 etapas para obtener la solución discreta aproximada y del PVI
% genérico de 2º orden (2.6) y su primera derivada z mediante un algoritmo
% directo que no implica considerar un Sistema de EDOS de primer orden (3.12).
% Las consideraciones sobre este método numérico, así como su Tablero de
% Butcher asociado pueden consultarse en 3.3.4.1.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI ( ) para poder computar las etapas del
% método numérico Runge-Kutta-Nyström explícito de 3º orden. Esta función
% debe de ser pasada como input de la siguiente manera empleando el
% "function_handle":
%     f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)
% NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.6)
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI (2.6) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
```

```
%-----  
format long  
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada  
% intervalo a partir de x:  
[n,h] = subint_pasos(x);  
% Declaración e inicialización de la solución discreta aproximada y del PVI  
(2.6) en x:  
y = zeros(n+1,1);  
% Declaración e inicialización de la primera derivada de la solución discreta  
% aproximada z del PVI (2.6) en x:  
z = zeros(n+1,1);  
% Inicialización de las etapas del método Runge-Kutta-Nyström explícito de 3º  
% orden:  
k1 = 0; k2 = 0; k3 = 0;  
% Imposición de las condiciones iniciales:  
y(1) = y0; z(1) = z0;  
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los  
% nodos de malla x:  
for i = 1:n  
    k1 = f(x(i),y(i),z(i));  
    k2 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*z(i)+1/8*h(i)^2*k1,z(i)+1/2*h(i)*k1);  
    k3 = f(x(i)+3/4*h(i),y(i)+3/4*h(i)*z(i)+9/32*h(i)^2*k2,z(i)+3/4*h(i)*k2);  
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:  
    y(i+1) = y(i)+h(i)*z(i)+h(i)^2*(2/9*k1+1/6*k2+1/9*k3);  
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:  
    z(i+1) = z(i)+h(i)*(2/9*k1+1/3*k2+4/9*k3);  
end  
end
```

- **Método RKN transformado de orden 3.**

```
function [y,z] = RKN3t(x,f,y0,z0)  
%-----  
% La función "RKN3t" implementa el método Runge-Kutta-Nyström explícito de  
% 3º orden y 3 etapas modificado para obtener la solución discreta aproximada  
% y del PVI genérico de 2º orden (2.6) y su primera derivada z mediante un  
% algoritmo directo que no implica considerar un Sistema de EDOs de primer  
% orden (3.12). Las consideraciones sobre este método numérico, así como su  
% Tablero de Butcher asociado pueden consultarse en 3.3.4.2.  
%  
% ENTRADAS A LA FUNCIÓN:  
%  
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados  
% crecientemente, cumpliendo x_1=a y x_{n+1}=b  
% f = Función genérica del PVI (2.6) para poder computar las etapas del  
% método numérico Runge-Kutta-Nyström explícito de 3º orden modificado.  
% Esta función debe de ser pasada como input de la siguiente manera  
% empleando el "function_handle":  
% f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)  
% NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)  
% y0 = 1ª condición inicial del PVI (2.6)  
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera  
% derivada  
%
```



```

% SALIDAS DE LA FUNCIÓN:
%
%   y = Vector columna que contiene los valores de la solución discreta
%   aproximada del PVI (2.6) en los n+1 nodos de x
%   z = Vector columna que contiene los valores de la primera derivada de
%   la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.6) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.6) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta-Nyström explícito de 3º
% orden:
K1 = 0; K2 = 0; K3 = 0;
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    K1 = f(x(i),y(i),z(i));
    K2 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*z(i),z(i)+1/2*h(i)*K1);
    K3 = f(x(i)+3/4*h(i),y(i)+3/4*h(i)*z(i)+3/8*h(i)^2*K1,z(i)+3/4*h(i)*K2);
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*z(i)+h(i)^2*(1/6*K1+1/3*K2);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(2/9*K1+1/3*K2+4/9*K3);
end
end

```

- **Método RKN de orden 4.**

```

function [y,z] = RKN4(x,f,y0,z0)
%-----
% La función "RKN4" implementa el método Runge-Kutta-Nyström explícito de 4º
% orden y 4 etapas para obtener la solución discreta aproximada y del PVI
% genérico de 2º orden (2.6) y su primera derivada z mediante un algoritmo
% directo que no implica considerar un Sistema de EDOs de primer orden (3.12).
% Las consideraciones sobre este método numérico, así como su Tablero de
% Butcher asociado pueden consultarse en 3.3.4.3.
%
% ENTRADAS A LA FUNCIÓN:
%
%   x = Vector columna que contiene los n+1 nodos distintos t y ordenados
%   crecientemente, cumpliendo x_1=a y x_{n+1}=b
%   f = Función genérica del PVI ( ) para poder computar las etapas del
%   método numérico Runge-Kutta-Nyström explícito de 4º orden. Esta función
%   debe de ser pasada como input de la siguiente manera empleando el
%   "function_handle":

```

```
%      f = @(x,y,z) [...Expresión...] y=y(x), z=y'(x)
%      NOTA: Para PVI's lineales del tipo (2.7) tomar f=p(x)y'(x)+q(x)y(x)+r(x)
%      y0 = 1ª condición inicial del PVI (2.6)
%      z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
%      derivada
%
% SALIDAS DE LA FUNCIÓN:
%
%      y = Vector columna que contiene los valores de la solución discreta
%      aproximada del PVI (2.6) en los n+1 nodos de x
%      z = Vector columna que contiene los valores de la primera derivada de
%      la solución discreta aproximada y del PVI (2.6) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
(2.6) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.6) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta-Nyström explícito de 3º
% orden:
k1 = 0; k2 = 0; k3 = 0; k4 = 0;
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    k1 = f(x(i),y(i),z(i));
    k2 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*z(i)+1/8*h(i)^2*k1,z(i)+1/2*h(i)*k1);
    k3 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*z(i)+1/8*h(i)^2*k1,z(i)+1/2*h(i)*k2);
    k4 = f(x(i)+h(i),y(i)+h(i)*z(i)+1/2*h(i)^2*k3,z(i)+h(i)*k3);
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*z(i)+h(i)^2*(1/6*k1+1/6*k2+1/6*k3);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(1/6*k1+1/3*k2+1/3*k3+1/6*k4);
end
end
```

- **Método RKN simple de orden 4 para problemas independientes de y' .**

```
function [y,z] = RKN4_no_dy(x,f,y0,z0)
%-----
% La función "RKN4_no_dy" implementa el método Runge-Kutta-Nyström explícito
% de 4º orden y 3 etapas para obtener la solución discreta aproximada y del
% PVI de 2º orden independiente de  $y'$  (2.8) y su primera derivada z mediante
% un algoritmo directo que no implica considerar un Sistema de EDOS de primer
% orden (3.12). Las consideraciones sobre este método numérico, así como su
% Tablero de Butcher asociado pueden consultarse en 3.3.5.1.
%
% ENTRADAS A LA FUNCIÓN:
```



```

%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI (2.8) para poder computar las etapas del
% método numérico Runge-Kutta-Nyström explícito de 4º orden. Esta función
% debe de ser pasada como input de la siguiente manera empleando el
% "function_handle":
%         f = @(x,y) [...Expresión...]
% NOTA: Para PVI's lineales e independientes de y' tomar f=q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.8)
% z0 = 2ª condición inicial del PVI (2.8) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI (2.8) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.8) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.8) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.8) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta-Nyström explícito de 3º
% orden:
k1 = 0; k2 = 0; k3 = 0;
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    k1 = f(x(i),y(i));
    k2 = f(x(i)+1/2*h(i),y(i)+1/2*h(i)*z(i)+1/8*h(i)^2*k1);
    k3 = f(x(i)+h(i),y(i)+h(i)*z(i)+1/2*h(i)^2*k2);
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*z(i)+h(i)^2*(1/6*k1+1/3*k2);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(1/6*k1+2/3*k2+1/6*k3);
end
end

```

- **Método RKN simple de orden 5 para problemas independientes de y' .**

```

function [y,z] = RKN51_no_dy(x,f,y0,z0)
%-----
% La función "RKN51_no_dy" implementa el método Runge-Kutta-Nyström explícito
% de 5º orden simple y 4 etapas para obtener la solución discreta aproximada y

```



```
% del PVI de 2º orden independiente de y' (2.8) y su primera derivada z
% mediante un algoritmo directo que no implica considerar un Sistema de EDOS
% de primer orden (3.12). Las consideraciones sobre este método numérico, así
% como su Tablero de Butcher asociado pueden consultarse en 3.3.5.2.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% f = Función genérica del PVI ( ) para poder computar las etapas del
% método numérico Runge-Kutta-Nyström explícito de 5º orden. Esta función
% debe de ser pasada como input de la siguiente manera empleando el
% "function_handle":
%           f = @(x,y) [...Expresión...]
% NOTA: Para PVIs lineales e independientes de y' tomar f=q(x)y(x)+r(x)
% y0 = 1ª condición inicial del PVI (2.8)
% z0 = 2ª condición inicial del PVI (2.8) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI ( ) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.8) en los n+1 nodos de x
%-----
format long
% cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.8) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.8) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta-Nyström explícito de 3º
% orden:
k1 = 0; k2 =0; k3 = 0; k4 = 0;
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    k1 = f(x(i),y(i));
    k2 = f(x(i)+1/5*h(i),y(i)+1/5*h(i)*z(i)+1/50*h(i)^2*k1);
    k3 = f(x(i)+2/3*h(i),y(i)+2/3*h(i)*z(i)+h(i)^2*(-1/27*k1+7/27*k2));
    k4 = f(x(i)+h(i),y(i)+h(i)*z(i)+h(i)^2*(3/10*k1-2/35*k2+9/35*k3));
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
    y(i+1) = y(i)+h(i)*z(i)+h(i)^2*(1/24*k1+25/84*k2+9/56*k3);
    % Almacenado sucesivo de las primeras derivadas de y en el vector z:
    z(i+1) = z(i)+h(i)*(1/24*k1+125/336*k2+27/56*k3+5/48*k4);
end
end
```



- **Método RKN especial de orden 5 para problemas independientes de y' .**

```
function [y,z] = RKN52_no_dy(x,f,y0,z0)
%-----
% La función "RKN52_no_dy" implementa el método Runge-Kutta-Nyström explícito
% de 5º orden especial y 4 etapas para obtener la solución discreta aproximada
% y del PVI de 2º orden independiente de  $y'$  (2.8) y su primera derivada z
% mediante un algoritmo directo que no implica considerar un Sistema de EDOS
% de primer orden (3.12). Las consideraciones sobre este método numérico, así
% como su Tablero de Butcher asociado pueden consultarse en 3.3.5.3.
%
% ENTRADAS A LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo  $x_1=a$  y  $x_{n+1}=b$ 
% f = Función genérica del PVI ( ) para poder computar las etapas del
% método numérico Runge-Kutta-Nyström explícito de 5º orden. Esta función
% debe de ser pasada como input de la siguiente manera empleando el
% "function_handle":
%           f = @(x,y) [...Expresión...]
% NOTA: Para PVI's lineales e independientes de  $y'$  tomar  $f=q(x)y(x)+r(x)$ 
% y0 = 1ª condición inicial del PVI (2.8)
% z0 = 2ª condición inicial del PVI (2.8) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% y = Vector columna que contiene los valores de la solución discreta
% aproximada del PVI (2.8) en los n+1 nodos de x
% z = Vector columna que contiene los valores de la primera derivada de
% la solución discreta aproximada y del PVI (2.8) en los n+1 nodos de x
%-----
format long
% Cálculo del número de subintervalos n en [a,b] y el paso variable h de cada
% intervalo a partir de x:
[n,h] = subint_pasos(x);
% Declaración e inicialización de la solución discreta aproximada y del PVI
% (2.8) en x:
y = zeros(n+1,1);
% Declaración e inicialización de la primera derivada de la solución discreta
% aproximada z del PVI (2.8) en x:
z = zeros(n+1,1);
% Inicialización de las etapas del método Runge-Kutta-Nyström explícito de 3º
% orden:
K1 = 0; K2 = 0; K3 = 0; K4 = 0;
% Imposición de las condiciones iniciales:
y(1) = y0; z(1) = z0;
% Obtención de la solución aproximada (y) y de su primera derivada (z) en los
% nodos de malla x:
for i = 1:n
    K1 = f(x(i),y(i));
    K2 = f(x(i)+2/5*h(i),y(i)+2/5*h(i)*z(i)+2/25*h(i)^2*K1);
    K3 = f(x(i)+2/3*h(i),y(i)+2/3*h(i)*z(i)+2/9*h(i)^2*K1);
    K4 = f(x(i)+4/5*h(i),y(i)+4/5*h(i)*z(i)+h(i)^2*(4/25*K1+4/25*K2));
    % Almacenado sucesivo de las soluciones aproximadas en el vector y:
```



```
y(i+1) = y(i)+h(i)*z(i)+h(i)^2*(23/192*k1+25/64*k2-9/64*k3+25/192*k4);  
% Almacenado sucesivo de las primeras derivadas de y en el vector z:  
z(i+1) = z(i)+h(i)*(23/192*k1+125/192*k2-27/64*k3+125/192*k4);  
end  
end
```



ANEXO II. Estrategias de interpolación basadas en splines.

Este anexo contiene las funciones de **MATLAB** concernientes al cálculo de parámetros (coeficientes) a_{ki} de los *splines* interpoladores que corresponden a las estrategias de interpolación de soluciones discretas del **PVI (2.6)** desarrolladas teóricamente en el **Capítulo 4** de la memoria. Los *splines* en cuestión han de construirse posteriormente mediante las funciones de **MATLAB** “*ppval*” y “*mkpp*”, pasándoles como dato los coeficientes calculados y los nodos de la malla definida en (2.11).

Cabe destacar que la obtención y posterior salida de las funciones con los coeficientes calculados y almacenados en una matriz se realiza en orden creciente. Las funciones evaluadoras comentadas en el párrafo anterior trabajan con ellos de forma decreciente, por lo que es preciso emplearlos invirtiendo el orden de las filas de la matriz salida, utilizando la función “*flip*”.

Todas las funciones y subrutinas auxiliares empleadas para la construcción de las de este anexo pueden ser consultadas en el **ANEXO V**.

- **Cálculo de los coeficientes del spline cuadrático interpolador.**

```
function [A] = spline2int(x,y,z0)
%-----
% La función "spline2int" obtiene los coeficientes del spline cuadrático
% interpolador (expuesto en la sección 4.3) de la solución discreta del PVI
% de 2º orden genérico dado por (2.6).
%
% En cada intervalo [t_k-1,t_k] de la malla unidimensional (2.11) con extremos
% a y b el spline s viene descrito por un polinomio s_k de grado <=3 definido
% por la expresión (4.3) que aquí toma la forma dada por:
%
%   S_k(x) = A(k,1)+A(k,2)*(x-t_k-1)+A(k,3)*(x-t_k-1)^2,
%   Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
%   x = Vector columna que contiene los n+1 nodos distintos t y ordenados
%   crecientemente, cumpliendo x_1=a y x_{n+1}=b
%   y = Vector columna que contiene los valores de la solución discreta del
%   PVI (2.6)
%   en los n+1 nodos de x calculada previamente mediante métodos numéricos
%   apropiados (Capítulo 3)
%   z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
%   derivada
%
% SALIDAS DE LA FUNCIÓN:
%
%   [A] = Matriz de dimensión n*3 que contiene los 3n coeficientes para
%   determinar unívocamente el spline cuadrático s interpolador de la
%   solución discreta y del PVI (2.6). La fila k-ésima de [A] contiene los
```



```

% coeficientes del tramo s_k en el intervalo [t_{k-1},t_k], con k=1,2,...,n y
% en la base [1, (x-t_k), (x-t_k)^2]
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del spline cuadrático S:
if (size(x) == size(y))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    % cada intervalo a partir de x:
    [n,h] = subint_pasos(x);
    % Declaración e inicialización de la matriz de coeficientes [A]:
    A = zeros(n,3);
    % Cálculo de los coeficientes de s en el primer intervalo [t_0=a,t_1]
    % según lo expuesto en X.X:
    A(1,1) = y(1); %a10
    A(1,2) = z0; %a11
    A(1,3) = 1/h(1)^2*(y(2)-A(1,1)-h(1)*A(1,2)); %a12
    % Cálculo de los coeficientes de s en los n-1 intervalos restantes del
    % dominio según lo expuesto en X.X mediante el empleo de un esquema
    % recursivo:
    for i = 2:n
        A(i,1) = A(i-1,1)+h(i-1)*(A(i-1,2)+h(i-1)*A(i-1,3)); %ak0 Horner
        A(i,2) = A(i-1,2)+2*h(i-1)*A(i-1,3); %ak1
        A(i,3) = 1/h(i)^2*(y(i+1)-A(i,1)-h(i)*A(i,2)); %ak2
    end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \Las dimensiones de los vectores de entrada no concuerdan
entre sí. Por favor, revise las entradas.')
end
end
end

```

- **Cálculo de los coeficientes del spline cúbico interpolador.**

```

function [A] = spline3int(x,y,f0,z0)
%-----
% La función "spline3int" obtiene los coeficientes del spline cúbico
% interpolador (expuesto en la sección 4.4) de la solución discreta del PVI
% de 2º orden genérico dado por (2.6). En cada intervalo [t_{k-1},t_k] de la
% malla unidimensional (2.11) con extremos a y b el spline s viene descrito
% por un polinomio s_k de grado <=3 definido por la expresión (2.6) que aquí
% toma la forma dada por:
%
% S_k(x) = A(k,1)+A(k,2)*(x-t_{k-1})+A(k,3)*(x-t_{k-1})^2+A(k,4)*(x-t_{k-1})^3,
% Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
% x = vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% y = vector columna que contiene los valores de la solución discreta del
% PVI (2.6)
% en los n+1 nodos de x calculada previamente mediante métodos numéricos

```

```
% apropiados (Capítulo 3)
% f0 = valor de la función f(x,y(x),y'(x)) del PVI () en x=t_0=a evaluado
% previamente
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% [A] = Matriz de dimensión n*4 que contiene los 4n coeficientes para
% determinar unívocamente el spline cúbico S interpolador de la solución
% discreta y del PVI (2.6). La fila k-ésima de [A] contiene los
% coeficientes del tramo S_k en el intervalo [t_{k-1},t_k], con k=1,2,...,n y
% en la base [1, (x-t_k), (x-t_k)^2, (x-t_k)^3]
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del spline cúbico S:
if (size(x) == size(y))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    % cada intervalo a partir de x:
    [n,h] = subint_pasos(x);
    % Declaración e inicialización de la matriz de coeficientes [A]:
    A = zeros(n,4);
    % Cálculo de los coeficientes de s en el primer intervalo [t_0=a,t_1]
    % según lo expuesto en X.X:
    A(1,1) = y(1); %a10
    A(1,2) = z0; %a11
    A(1,3) = 1/2*f0; %a12
    A(1,4) = 1/h(1)^3*(y(2)-A(1,1)-h(1)*A(1,2)-h(1)^2*A(1,3)); %a13
    % Cálculo de los coeficientes de s en los n-1 intervalos restantes del
    % dominio según lo expuesto en X.X mediante el empleo de un esquema
    % recursivo:
    for i = 2:n
        A(i,1) = A(i-1,1)+h(i-1)*(A(i-1,2)+h(i-1)*(A(i-1,3)+h(i-1)*A(i-1,4)));
        %ak0 Horner
        A(i,2) = A(i-1,2)+h(i-1)*(2*A(i-1,3)+3*h(i-1)*A(i-1,4)); %ak1 Horner
        A(i,3) = A(i-1,3)+3*h(i-1)*A(i-1,4); %ak2
        A(i,4) = 1/h(i)^3*(y(i+1)-A(i,1)-h(i)*A(i,2)-h(i)^2*A(i,3)); %ak3
    end
end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \Las dimensiones de los vectores de entrada no concuerdan
entre sí. Por favor, revise las entradas.')
end
end
```



ANEXO III. Estrategias de interpolación basadas en *pseudo-splines*.

Este anexo contiene las funciones de **MATLAB** para el cálculo de parámetros (coeficientes) a_{ki} de los *pseudo-splines* interpoladores correspondientes a las estrategias de interpolación de soluciones discretas de los **PVIs** (2.6), (2.7) y (2.8) desarrolladas teóricamente en el **Capítulo 5** de la memoria. Los *pseudo-splines* cuárticos (sección 5.2) en cuestión han de construirse (evaluarse) posteriormente mediante las funciones de **MATLAB** “ppval” y “mkpp”, pasándoles como dato los coeficientes calculados y los nodos de la malla definida en (2.11). En cambio, los *pseudo-splines* cúbicos (5.3.2) y quínticos (5.3.3), al no ser obtenidos en la base de polinomios de los cuárticos (la que utiliza **MATLAB** por defecto), deben ser evaluados mediante las funciones aquí incluidas y citadas en la **Tabla 5.12** del **Capítulo 5**.

Al igual que las funciones para los splines del **ANEXO II**, es preciso emplear “flip” para posteriormente poder evaluar los *pseudo-splines* cuárticos correctamente. **Esto no es necesario para los cúbicos y los quínticos**. Todas las subrutinas o funciones auxiliares empleadas en la construcción de las de este anexo pueden ser consultadas en el **ANEXO V**.

- Cálculo de los coeficientes del *pseudo-spline* cuártico interpolador empleando la estrategia de colocación (secciones 5.2.4.1.1 y 5.2.5.1.1) para problemas estrictamente independientes de y' (2.8).

```
function [A] = pspline4int_col1(x,y,f,z0)
%-----
% La función "pspline4int_col1" obtiene los coeficientes del pseudo-spline
% cuártico interpolador (expuesto en las secciones 5.2.4.1 y 5.2.5.1) de la
% solución discreta del PVI de 2º orden independiente de  $y'$  dado por (2.8)
% utilizando la estrategia de colocación (sección 5.2.1).
%
% En cada intervalo  $[t_{k-1}, t_k]$  de la malla unidimensional (2.11) con extremos
% a y b el pseudo-spline  $S$  viene descrito por un polinomio  $S_k$  de grado  $\leq 4$ 
% definido por la expresión (5.1) que aquí toma la forma dada por:
%
%  $S_k(x) = A(k,1) + A(k,2)*(x-t_{k-1}) + A(k,3)*(x-t_{k-1})^2 + A(k,4)*(x-t_{k-1})^3 +$ 
%  $+ A(k,5)*(x-t_{k-1})^4,$ 
% Para cada  $k = 1, 2, \dots, n$  (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
% x = vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo  $x_1=a$  y  $x_{n+1}=b$ 
% y = vector columna que contiene los valores de la solución discreta del
% PVI (2.8)
% en los n+1 nodos de x calculada previamente mediante métodos numéricos
% apropiados (Capítulo 3)
% f = vector columna que contiene los valores en los n+1 nodos de x de la
% función  $f(x, y(x))$  del PVI (2.8) evaluada previamente
```



```

% z0 = 2ª condición inicial del PVI (2.8) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% [A] = Matriz de dimensión n*5 que contiene los 5n coeficientes para
% determinar unívocamente el pseudo-spline cuártico S interpolador de la
% solución discreta y del PVI (2.8). La fila k-ésima de [A] contiene los
% coeficientes del tramo s_k en el intervalo [t_{k-1},t_k], con k=1,2,...,n y
% en la base [1, (x-t_k), (x-t_k)^2, (x-t_k)^3], (x-t_k)^4]
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del pseudo-spline cuártico S:
if (size(x) == size(y) & size(y) == size(f))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    % cada intervalo a partir de x:
    [n,h] = subint_pasos(x);
    % Declaración e inicialización de la matriz de coeficientes [A]:
    A = zeros(n,5);
    % Cálculo de los coeficientes de s en el primer intervalo [t_0=a,t_1]
    % según lo expuesto en X.X:
    A(1,1) = y(1); %a10
    A(1,2) = z0; %a11
    A(1,3) = 1/2*f(1); %a12
    A(1,4) = 2/h(1)^3*(-A(1,1)-h(1)*A(1,2)-5/6*h(1)^2*A(1,3)+y(2)-
    1/12*h(1)^2*f(2)); %a13
    A(1,5) = 1/h(1)^4*(A(1,1)+h(1)*A(1,2)+2/3*h(1)^2*A(1,3)+1/6*h(1)^2*f(2)-
    y(2)); %a14
    % Cálculo de los coeficientes de s en los n-1 intervalos restantes del
    % dominio según lo expuesto en X.X mediante el empleo de un esquema
    % recursivo:
    for i = 2:n
        A(i,1:3) = pspline4int_coefs_com(A(i-1,:),h(i-1)); %ak0, ak1, ak2
        A(i,4) = 2/h(i)^3*(-A(i,1)-h(i)*A(i,2)-5/6*h(i)^2*A(i,3)+y(i+1)-
        1/12*h(i)^2*f(i+1)); %ak3
        A(i,5) =
        1/h(i)^4*(A(i,1)+h(i)*A(i,2)+2/3*h(i)^2*A(i,3)+1/6*h(i)^2*f(i+1)-y(i+1)); %ak4
    end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \Las dimensiones de los vectores de entrada no concuerdan
    entre sí. Por favor, revise las entradas.')
end
end

```

- **Cálculo de los coeficientes del pseudo-spline cuártico interpolador empleando la estrategia de colocación (secciones 5.2.4.1.2 y 5.2.5.1.2) para problemas estrictamente lineales (2.7).**

```

function [A] = pspline4int_col2(x,y,p,q,r,z0)
%-----
% La función "pspline4int_col2" obtiene los coeficientes del pseudo-spline
% cuártico interpolador (expuesto en las secciones 5.2.4.1 y 5.2.4.2) de la

```

```
% solución discreta del PVI de
% 2º orden lineal dado por (2.7) utilizando la estrategia de colocación
% (sección 5.2.1).
%
% En cada intervalo [t_k-1,t_k] de la malla unidimensional (2.11) con extremos
% a y b el pseudo-spline S viene descrito por un polinomio S_k de grado <=4
% definido por la expresión (5.1) que aquí toma la forma dada por:
%
% S_k(x) = A(k,1)+A(k,2)*(x-t_k-1)+A(k,3)*(x-t_k-1)^2+A(k,4)*(x-t_k-1)^3
% +A(k,5)*(x-t_k-1)^4,
% Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b
% y = Vector columna que contiene los valores de la solución discreta del
% PVI (2.7)
% en los n+1 nodos de x calculada previamente mediante métodos numéricos
% apropiados (Capítulo 3)
% p = Vector columna que contiene los valores en los n+1 nodos de x del
% coeficiente p(x) que acompaña a y'(x) del PVI (2.7)
% q = Vector columna que contiene los valores en los n+1 nodos de x del
% coeficiente q(x) que acompaña a y(x) del PVI (2.7)
% r = Vector columna que contiene los valores en los n+1 nodos de x del
% coeficiente r(x) del PVI (2.7)
% z0 = 2ª condición inicial del PVI (2.7) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% [A] = Matriz de dimensión n*5 que contiene los 5n coeficientes para
% determinar unívocamente el pseudo-spline cuártico S interpolador de la
% solución discreta y del PVI (2.7). La fila k-ésima de [A] contiene los
% coeficientes del tramo S_k en el intervalo [t_k-1,t_k], con k=1,2,...,n y
% en la base [1, (x-t_k), (x-t_k)^2, (x-t_k)^3], (x-t_k)^4]
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del pseudo-spline cuártico S:
if (size(x) == size(y) & size(y) == size(p) & size(p) == size(q) & size(q) == size(r))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    cada intervalo a partir de x:
    [n,h] = subint_pasos(x);
    % Declaración e inicialización de la matriz de coeficientes [A]:
    A = zeros(n,5);
    % Cálculo de los coeficientes de S en el primer intervalo [t_0=a,t_1]
    % según lo expuesto en X.X:
    A(1,1) = y(1); %a10
    A(1,2) = z0; %a11
    A(1,3) = 1/2*(p(1)*z0+q(1)*y(1)+r(1)); %a12
        % Vectores columna de agrupaciones para evaluar los coeficientes a13 y a14:
        u = zeros(n,1);
        v = zeros(n,1);
        % Se aprovecha para evaluar todos los intervalos en un único bucle:
```



```

for i = 1:n
    u(i) = 3*h(i)*(2-p(i)*h(i)); % Agrupación A
    v(i) = 4*h(i)^2*(3-p(i)*h(i)); % Agrupación B
end
A(1,4) = h(1)/(u(1)*h(1)-
v(1))*(v(1)/h(1)^3*A(1,1)+(p(2)+v(1)/h(1)^3)*A(1,2)+(2*p(2)*h(1)-
2+v(1)/h(1)^2)*A(1,3)+(q(2)-v(1)/h(1)^4)*y(2)+r(2)); %a13
A(1,5) = 1/h(1)^4*(y(2)-A(1,1)-h(1)*A(1,2)-h(1)^2*A(1,3)-h(1)*A(1,4)); %a14
% cálculo de los coeficientes de s en los n-1 intervalos restantes del
% dominio según lo expuesto en X.X mediante el empleo de un esquema
% recursivo:
for i=2:n
    A(i,1:3) = pspline4int_coefs_com(A(i-1,:),h(i-1)); %ak0, ak1, ak2
    A(i,4) = h(i)/(u(i)*h(i)-
v(i))*(v(i)/h(i)^3*A(i,1)+(p(i+1)+v(i)/h(i)^3)*A(i,2)+(2*p(i+1)*h(i)-
2+v(i)/h(i)^2)*A(i,3)+(q(i+1)-v(i)/h(i)^4)*y(i+1)+r(i+1)); %ak3
    A(i,5) = 1/h(i)^4*(y(i+1)-A(i,1)-h(i)*A(i,2)-h(i)^2*A(i,3)-
h(i)*A(i,4)); %a14
end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \Las dimensiones de los vectores de entrada no concuerdan
entre sí. Por favor, revise las entradas.')
end
end
end

```

- **Cálculo de los coeficientes del pseudo-spline cuártico interpolador empleando la estrategia de interpolación intermedia (secciones 5.2.4.2 y 5.2.5.2).**

```

function [A] = pspline4int_intm(x_inicial,y_densa,f0,z0)
%-----
% La función "pspline4int_intm" obtiene los coeficientes del pseudo-spline
cuártico interpolador (expuesto en la sección 5.2.4.2 y 5.2.5.2) de la
solución discreta del PVI de 2º orden genérico dado por (2.6) utilizando la
estrategia de interpolación intermedia (sección 5.2.2).
%
% En cada intervalo [t_k-1,t_k] de la malla unidimensional (2.11) con extremos
a y b el pseudo-spline s viene descrito por un polinomio s_k de grado <=4
definido por la expresión (5.1) que aquí toma la forma dada por:
%
%   S_k(x) = A(k,1)+A(k,2)*(x-t_k-1)+A(k,3)*(x-t_k-1)^2+A(k,4)*(x-t_k-1)^3
%   +A(k,5)*(x-t_k-1)^4,
%   Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
%   x_inicial = vector columna que contiene los n+1 nodos distintos t y
ordenados crecientemente, cumpliendo x_inicial_1=a y x_inicial_n+1=b
%   y_densa = vector columna que contiene 2n+1 los valores de la solución
discreta
%   del PVI (2.6) (calculada mediante métodos numéricos apropiados) en los
2n+1 nodos de x tras haber densificado el mallado con nodos intermedios
%   en cada subintervalo

```

```
% f0 = valor de la función f(x,y(x),y'(x)) del PVI (2.6) en x=t_0=a
% evaluado previamente
% z0 = 2ª condición inicial del PVI (2.6) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% [A] = Matriz de dimensión n*5 que contiene los 5n coeficientes para
% determinar unívocamente el pseudo-spline cuártico S interpolador de la
% solución discreta y del PVI (2.6). La fila k-ésima de [A] contiene los
% coeficientes del tramo S_k en el intervalo [t_{k-1},t_k], con k=1,2,...,n
% y en la base [1, (x-t_k), (x-t_k)^2, (x-t_k)^3, (x-t_k)^4]
%
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del pseudo-spline cuártico S:
if (size(x_inicial,1)-1) == (size(y_densa,1)-size(x_inicial,1))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    % cada intervalo a partir de x:
    [n,h] = subint_pasos(x_inicial);
    % Declaración e inicialización de la matriz de coeficientes [A]:
    A = zeros(n,5);
    % Cálculo de los coeficientes de s en el primer intervalo [t_0=a,t_1]
    % según lo expuesto en X.X:
    A(1,1) = y_densa(1); %a10
    A(1,2) = z0; %a11
    A(1,3) = 1/2*f0; %a12
    A(1,4) = 1/h(1)^3*(16*y_densa(2)-y_densa(3)-15*A(1,1)-7*h(1)*A(1,2)-
    3*h(1)^2*A(1,3)); %a13
    A(1,5) = 1/h(1)^4*(y_densa(3)-
    16*y_densa(2)+14*A(1,1)+6*h(1)*A(1,2)+2*h(1)^2*A(1,3)); %a14
    % Cálculo de los coeficientes de s en los n-1 intervalos restantes del
    % dominio según lo expuesto en X.X mediante el empleo de un esquema
    % recursivo:
    for i = 2:n
        A(i,1:3) = pspline4int_coefs_com(A(i-1,:),h(i-1)); %ak0, ak1, ak2
        A(i,4) = 1/h(i)^3*(16*y_densa(2*i)-y_densa(2*i+1)-15*A(i,1)-
        7*h(i)*A(i,2)-3*h(i)^2*A(i,3)); %ak3
        A(i,5) = 1/h(i)^4*(y_densa(2*i+1)-
        16*y_densa(2*i)+14*A(1,1)+6*h(i)*A(i,2)+2*h(i)^2*A(i,3)); %ak4
    end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \La dimensión de la tercera entrada no concuerda. Por favor,
    revise la tercera entrada.')
end
end
```



- Cálculo de los coeficientes del pseudo-spline cuártico interpolador empleando la estrategia de colocación intermedia para problemas estrictamente lineales (2.7) (secciones 5.2.4.3.1 y 5.2.5.3.1).

```
function [A] = pspline4int_colm_lin(x_inicial,y,p_denso,q_denso,r_denso,z0)
%-----
% La función "pspline4int_colm_lin" obtiene los coeficientes del pseudo-spline
% cuártico interpolador (expuesto en la sección 5.2.4.3 y 5.2.5.3) de la
% solución discreta del PVI de 2º orden lineal dado por (2.7) utilizando la
% estrategia de colocación intermedia (sección 5.2.3).
%
% En cada intervalo [t_k-1,t_k] de la malla unidimensional (2.11) con extremos
% a y b el pseudo-spline S viene descrito por un polinomio S_k de grado <=4
% definido por la expresión (5.1) que aquí toma la forma dada por:
%
% S_k(x) = A(k,1)+A(k,2)*(x-t_k-1)+A(k,3)*(x-t_k-1)^2+A(k,4)*(x-t_k-1)^3
% +A(k,5)*(x-t_k-1)^4,
% Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
% x_inicial = vector columna que contiene los n+1 nodos distintos t y
% ordenados crecientemente, cumpliendo x_inicial_1=a y x_inicial_n+1=b
% y = vector columna que contiene los valores de la solución discreta
% del PVI (2.7) en los n+1 nodos de x_inicial calculada previamente
% mediante métodos numéricos apropiados (Capítulo 3)
% p_denso = vector columna que contiene los valores en los 2n+1 nodos
% de x del coeficiente p(x) que acompaña a y'(x) del PVI (2.7) tras haber
% densificado el mallado con nodos intermedios en cada subintervalo
% q_denso = vector columna que contiene los valores en los 2n+1 nodos de x
% del coeficiente q(x) que acompaña a y(x) del PVI (2.7) tras haber
% densificado el mallado con nodos intermedios en cada subintervalo
% r_denso = vector columna que contiene los valores en los 2n+1 nodos de x
% del coeficiente r(x) del PVI (2.7) tras haber densificado el mallado con
% nodos intermedios en cada subintervalo
% z0 = 2ª condición inicial del PVI (2.7) en términos de la primera
% derivada
%
% SALIDAS DE LA FUNCIÓN:
%
% [A] = Matriz de dimensión n*5 que contiene los 5n coeficientes para
% determinar unívocamente el pseudo-spline cuártico S interpolador de la
% solución discreta y del PVI (2.7). La fila k-ésima de [A] contiene los
% coeficientes del tramo S_k en el intervalo [t_k-1,t_k], con
% k=1,2,...,n y en la base [1, (x-t_k), (x-t_k)^2, (x-t_k)^3], (x-t_k)^4]
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del pseudo-spline cuártico S:
if (size(x_inicial) == size(y) & size(p_denso) == size(q_denso) &
size(q_denso) == size(r_denso))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    cada intervalo a partir de x:
```

```
[n,h] = subint_pasos(x_inicial);
% Declaración e inicialización de la matriz de coeficientes [A]:
A = zeros(n,5);
% Cálculo de los coeficientes de s en el primer intervalo [t_0=a,t_1]
% según lo expuesto en X.X:
A(1,1) = y(1); %a10
A(1,2) = z0; %a11
A(1,3) = 1/2*(p_denso(1)*z0+q_denso(1)*y(1)+r_denso(1)); %a12
% Vectores columna de agrupaciones para evaluar los coeficientes a13 y a14:
u = zeros(n,1);
v = zeros(n,1);
% Se aprovecha para evaluar todos los intervalos en un único bucle:
for i = 1:n
    u(i) = h(i)*(3-h(i))*(3/4*p_denso(2*i)+1/8*q_denso(2*i)*h(i));
% Agrupación A
    v(i) = h(i)^2*(3-h(i))*(1/2*p_denso(2*i)+1/16*q_denso(2*i)*h(i));
% Agrupación B
end
A(1,4) = h(1)/(u(1)*h(1)-
v(1))*((q_denso(2)+v(1)/h(1)^4)*A(1,1)+(p_denso(2)+1/2*q_denso(2)*h(1)+v(1)/h(
1)^3)*A(1,2)+ ...
(p_denso(2)+1/4*q_denso(2)*h(1)^4+v(1)/h(1)^2-
2)*A(1,3)+r_denso(2)-v(1)*y(2)/h(1)^4); %a13
A(1,5) = 1/h(1)^4*(y(2)-A(1,1)-h(1)*A(1,2)-h(1)^2*A(1,3)-
h(1)^3*A(1,4)); %a14
% Cálculo de los coeficientes de s en los n-1 intervalos restantes del
% dominio según lo expuesto en X.X mediante el empleo de un esquema
% recursivo:
for i = 2:n
    A(i,1:3) = pspline4int_coefs_com(A(i-1,:),h(i-1)); %ak0, ak1, ak2
    A(i,4) = h(i)/(u(i)*h(i)-v(i))*((q_denso(2*i)-
v(i)/h(i)^4)*A(i,1)+(p_denso(2*i)+1/2*q_denso(2*i)*h(i)+v(i)/h(i)^3)*A(i,2)+
(p_denso(2*i)+1/4*q_denso(2*i)*h(i)^2+v(i)/h(i)^2-2)*A(i,3)
+r_denso(2*i)-v(i)*y(i+1)/h(i)^4); %ak3
    A(i,5) = 1/h(i)^4*(y(i+1)-A(i,1)-h(i)*A(i,2)-h(i)^2*A(i,3)-
h(i)^3*A(i,4)); %ak4
end
else
% FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
error('Error. \Las dimensiones de los vectores de entrada no concuerdan
entre sí. Por favor, revise las entradas.')
end
end
```

- **Cálculo de los coeficientes del pseudo-spline cúbico interpolador empleando la estrategia de interpolación de Hermite para el PVI (2.6) (sección 5.3.2).**

```
function [A] = pspline3int(x,y,z)
%-----
% La función "pspline3int" obtiene los coeficientes del pseudo-spline cúbico
% interpolador (expuesto en la sección 5.3.2) de la solución discreta del PVI
% de 2º orden genérico dado por (2.6) y su primera derivada empleando
% interpolación de Hermite con diferencias divididas de Newton y argumentos
```



```

% repetidos.
%
% En cada intervalo [t_k-1,t_k] de la malla unidimensional (2.11) con extremos
% a y b el pseudo-spline S viene descrito por un polinomio S_k de grado <=3
% definido por la expresión (5.66) que toma la forma dada por:
%
%   S_k(x) = A(k,1)+A(k,2)*(x-t_k-1)+A(k,3)*(x-t_k-1)^2+A(k,4)*(x-t_k-
%   1)^2*(x-t_k)
%   Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
%   x = vector columna que contiene los n+1 nodos distintos t y ordenados
%   crecientemente, cumpliendo x_1=a y x_{n+1}=b
%   y = vector columna que contiene los valores de la solución discreta
%   del PVI (2.6) en los n+1 nodos de x calculada previamente mediante
%   métodos numéricos apropiados (Capítulo 3)
%   z = vector columna que contiene los valores de la primera derivada de
%   la solución discreta y en los n+1 nodos de x calculada previamente por
%   el mismo método numérico que evalúa la solución y del PVI (2.6)
%
% SALIDAS DE LA FUNCIÓN:
%
%   [A] = Matriz de dimensión n*4 que contiene los 4n coeficientes para
%   determinar unívocamente el pseudo-spline cúbico S interpolador de la
%   solución discreta y del PVI () y de su primera derivada z. La fila k-
%   ésima de [A] contiene los coeficientes del tramo s_k en el intervalo
%   [t_k-1,t_k], con k=1,2,...,n y en la base [1, (x-t_k-1), (x-t_k-1)^2, (x-
%   t_k-1)^2*(x-t_k)]
%-----
format long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del pseudo-spline cúbico S:
if (size(x) == size(y) & size(y) == size(z))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    cada intervalo a partir de x:
    [n,h] = subint_pasos(x);
    % Declaración e inicialización de la matriz de numeradores de coeficientes
    [A]:
    A = zeros(n,4);
    % Cálculo de los numeradores de los coeficientes de S en los n intervalos
    del dominio según lo expuesto en X.X mediante el empleo de un esquema
    recursivo:
    for i = 1:n
        A(i,1) = y(i); %y[t_k-1], D.D. o/0
        A(i,2) = z(i); %y[t_k-1,t_k-1], D.D o/1
        A(i,3) = 1/h(i)^2*(y(i+1)-y(i)-h(i)*z(i));
        %y[t_k-1,t_k-1,t_k], D.D o/2
        A(i,4) = 2/h(i)^3*(y(i)-y(i+1))+1/h(i)^2*(z(i+1)+z(i));
        %y[t_k-1,t_k-1,t_k,t_k], D.D o/3
    end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \Las dimensiones de los vectores de entrada no concuerdan

```



```
entre sí. Por favor, revise las entradas.')
```

```
end
```

```
end
```

- **Evaluación intervalo a intervalo del pseudo-spline cúbico interpolador mediante interpolación de Hermite.**

```
function [PS3int_k] = eval_pspline3int(A,xlocal,b)
%-----
% La función "eval_pspline3int" evalúa el tramo k-ésimo S_k del pseudo-spline
% cúbico interpolador S de las soluciones discretas del PVI de 2º orden
% genérico dado por (2.6) y que ha sido calculado mediante interpolación de
% Hermite con argumentos repetidos en la sección 5.3.2 del Capítulo 5.
%
% En el intervalo [t_k-1,t_k] el pseudo-spline cúbico S viene descrito por un
% polinomio S_k de grado <=3 definido por la expresión (5.66) que aquí toma la
% forma dada por:
%
%   S_k(x) = A(1,1)+A(1,2)*(x-t_k-1)+A(1,3)*(x-t_k-1)^2+A(1,4)*(x-t_k-
%   1)^2*(x-t_k),
%   con h_k = t_k-t_k-1
%
% La razón de ser de esta función es la imposibilidad de recurrir a "mkpp"
% y posteriormente "ppval" para evaluar el polinomio en el intervalo en
% cuestión al no emplear la base de polinomios que dichas funciones utilizan
%
% ENTRADAS DE LA FUNCIÓN:
%
%   A = Vector fila que contiene los 4 coeficientes del tramo k-ésimo
%   (intervalo [t_k-1,t_k] del pseudo-spline cúbico interpolador del PVI
%   (2.6) ordenados crecientemente. Este vector ha de corresponder a una de
%   las filas de la salida de la función "pspline3int".
%   xlocal = Densificado local de la variable independiente para producir
%   una salida por pantalla con mayor nitidez
%   b = Nodo extremo final del intervalo el que el PVI debe de ser
%   resuelto e interpolado
%
% SALIDAS DE LA FUNCIÓN:
%
%   [PS3int_k] = vector columna que contiene los valores del polinomio
%   evaluado que define el pseudo-spline cúbico interpolador del PVI (2.6)
%   en el intervalo [t_k-1,t_k]
%-----
format long
% Nodos inicial y final del intervalo considerado:
tk1 = xlocal(1); tk = xlocal(end);
% Polinomio que debe de ser evaluado en cada nodo de entrada:
Sk = @(x) A(1,1)+A(1,2).*(x-tk1)+A(1,3).*(x-tk1).^2+A(1,4).*(x-tk1).^2.*(x-
tk);
% Declaración e inicialización del vector que almacena los valores del
% polinomio evaluado:
PS3int_k = zeros(size(xlocal),1,1);
% Evaluación progresiva del polinomio:
if tk ~= b
```



```

for i = 1:size(xlocal,1)-1
    PS3int_k(i) = sk(xlocal(i));
end
else
for i = 1:size(xlocal,1)
    PS3int_k(i) = sk(xlocal(i));
end
end
end
end

```

- **Cálculo de los coeficientes del pseudo-spline quintico interpolador para el PVI (2.6) (sección 5.3.3).**

```

function [A] = pspline5int(x,y,z,w)
%-----
% La función "pspline5int" obtiene los coeficientes del pseudo-spline quintico
% interpolador (expuesto en la sección 5.3.3) de la solución discreta del PVI
% de 2º orden genérico dado por (2.6) y sus derivadas primera y segunda
% empleando interpolación con polinomios osculadores, diferencias divididas de
% Newton y argumentos repetidos.
%
% En cada intervalo [t_k-1,t_k] de la malla unidimensional (2.11) con extremos
% a y b el pseudo-spline S viene descrito por un polinomio S_k de grado <=5
% definido por la expresión (5.68) que toma la forma dada por:
%
%   S_k(x) = A(k,1)+A(k,2)*(x-t_k-1)+A(k,3)*(x-t_k-1)^2+A(k,4)*(x-t_k-1)^3+
%   +A(k,5)*(x-t_k-1)^3*(x-t_k)+A(k,6)*(x-t_k-1)^3*(x-t_k)^2
%   Para cada k = 1,2,...,n (n lo calcula la función)
%
% ENTRADAS DE LA FUNCIÓN:
%
%   x = vector columna que contiene los n+1 nodos distintos t y ordenados
%   crecientemente, cumpliendo x_1=a y x_{n+1}=b
%   y = vector columna que contiene los valores de la solución discreta
%   del PVI (2.6) en los n+1 nodos de x calculada previamente mediante
%   métodos numéricos apropiados (Capítulo 3)
%   z = vector columna que contiene los valores de la primera derivada de
%   la solución discreta y en los n+1 nodos de x calculada previamente por
%   el mismo método numérico que evalúa la solución y del PVI (2.6)
%   w = vector columna que contiene los valores de la segunda derivada de
%   la solución discreta y en los n+1 nodos de x obtenida mediante una
%   estrategia de colocación sobre el PVI genérico (2.6) y evaluada
%   previamente
%
% SALIDAS DE LA FUNCIÓN:
%
%   [A] = Matriz de dimensión n*6 que contiene los 6n coeficientes para
%   determinar unívocamente el pseudo-spline quintico S interpolador de
%   la solución discreta y del PVI (2.6), así como de su primera (z) y
%   segunda (w) derivada. La fila k-ésima de [A] contiene los coeficientes
%   del tramo S_k en el intervalo [t_{k-1},t_k], con k=1,2,...,n y en la base
%   [1, (x-t_{k-1}), (x-t_{k-1})^2, (x-t_{k-1})^3, (x-t_{k-1})^3*(x-t_k), (x-t_{k-1})^3*(x-t_k)^2]
%-----

```

```
format Long
% Comprobación de los tamaños de las entradas antes de proceder a calcular
% los coeficientes del pseudo-spline quintico S:
if (size(x) == size(y) & size(y) == size(z) & size(z) == size(w))
    % TRUE: EL PROGRAMA CONTINÚA
    % Cálculo del número de subintervalos n en [a,b] y el paso variable h de
    % cada intervalo a partir de x:
    [n,h] = subint_pasos(x);
    % Declaración e inicialización de la matriz de numeradores de coeficientes
    [A]:
    A = zeros(n,6);
    % Cálculo de los numeradores de los coeficientes de S en los n intervalos
    % del dominio según lo expuesto en X.X mediante el empleo de un esquema
    % recursivo:
    for i = 1:n
        A(i,1) = y(i); %y[t_k-1] D.D. o/0
        A(i,2) = z(i); %y[t_k-1,t_k-1] D.D. o/1
        A(i,3) = 1/2*w(i); %y[t_k-1,t_k-1,t_k-1] D.D. o/2
        A(i,4) = 1/h(i)^3*(y(i+1)-y(i)-h(i)*z(i))-w(i)/(2*h(i));
        %y[t_k-1,t_k-1,t_k-1,t_k] D.D. o/3
        A(i,5) = w(i)/(2*h(i)^2)+1/h(i)^4*(3*(y(i)-
        y(i+1))+h(i)*(2*z(i)+z(i+1))); %y[t_k-1,t_k-1,t_k-1,t_k,t_k] D.D. o/4
        A(i,6) = (w(i+1)-w(i))/h(i)^3-3/h(i)^5*(2*(y(i)-
        y(i+1))+h(i)*(z(i)+z(i+1))); %y[t_k-1,t_k-1,t_k-1,t_k,t_k,t_k] D.D. o/5
    end
else
    % FALSE: SE LANZA UN MENSAJE DE ERROR PARA COMPROBAR LAS ENTRADAS
    error('Error. \Las dimensiones de los vectores de entrada no concuerdan
    entre sí. Por favor, revise las entradas.')
end
end
```

- **Evaluación intervalo a intervalo del pseudo-spline quintico interpolador.**

```
function [PS5int_k] = eval_pspline5int(A,xlocal,b)
%-----
% La función "eval_pspline5int" evalúa el tramo k-ésimo S_k del pseudo-spline
% quintico interpolador S de las soluciones discretas del PVI de 2º orden
% genérico dado por () y que ha sido calculado mediante interpolación con
% polinomios osculadores y argumentos repetidos en la sección 5.3.3 del
% Capítulo 5.
% En el intervalo [t_k-1,t_k] el pseudo-spline quintico S viene descrito por
% un polinomio S_k de grado <=5 definido por la expresión (5.68) que aquí toma
% la forma dada por:
%
% S_k(x) = A(1,1)+A(1,2)*(x-t_k-1)+A(1,3)*(x-t_k-1)^2+A(1,4)*(x-t_k-
% 1)^2*(x-t_k)+A(1,5)*(x-t_k-1)^3*(x-t_k)+A(1,6)*(x-t_k-1)^3*(x-t_k)^2,
% con h_k = t_k-t_k-1
%
% La razón de ser de esta función es la imposibilidad de recurrir a "mkpp"
% y posteriormente "ppval" para evaluar el polinomio en el intervalo en
% cuestión al no emplear la base de polinomios que dichas funciones
% utilizan.
%
```



```
% ENTRADAS DE LA FUNCIÓN:
%
%   A = vector fila que contiene los 6 coeficientes del tramo k-ésimo
%   (intervalo [t_k-1,t_k] del pseudo-spline quíntico interpolador del PVI
%   (2.6) ordenados crecientemente. Este vector ha de corresponder a una de
%   las filas de la salida de la función "pspline5int"
%   xlocal = Densificado local de la variable independiente para producir
%   una salida por pantalla con mayor nitidez
%   b = Nodo extremo final del intervalo el que el PVI debe de ser
%   resuelto e interpolado
%
% SALIDAS DE LA FUNCIÓN:
%
%   [PS5int_k] = vector columna que contiene los valores del polinomio
%   evaluado que define el pseudo-spline quíntico interpolador del PVI (2.6)
%   en el intervalo [t_k-1,t_k]
%-----
format long
% Nodos inicial y final del intervalo considerado:
tk1 = xlocal(1); tk = xlocal(end);
% Polinomio que debe de ser evaluado en cada nodo de entrada:
sk = @(x) A(1,1)+A(1,2).*(x-tk1)+A(1,3).*(x-tk1).^2+A(1,4).*(x-
tk1).^3+A(1,5).*(x-tk1).^3.*(x-tk1)+A(1,6).*(x-tk1).^3.*(x-tk).^2;
% Declaración e inicialización del vector que almacena los valores del
% polinomio evaluado:
PS5int_k = zeros(size(xlocal,1),1);
% Evaluación progresiva del polinomio:
if tk ~= b
    for i = 1:size(xlocal,1)-1
        PS5int_k(i) = sk(xlocal(i));
    end
else
    for i = 1:size(xlocal,1)
        PS5int_k(i) = sk(xlocal(i));
    end
end
end
```

ANEXO IV. Códigos de los experimentos numéricos.

En este anexo se incluyen, en el mismo orden en el que han sido realizados y documentados en el **Capítulo 7**, los códigos de **MATLAB** correspondientes a todos los experimentos numéricos. Estos programas se han llevado a cabo mediante el uso de todas las funciones programadas y adjuntas en los **ANEXOS I, II y III**, además de las funciones auxiliares del **ANEXO V**.

Los nombres de los *scripts* son concordantes con los indicados en la **Tabla 7.1**.

Experimento numérico 1. Vibraciones lineales de 1 grado de libertad (sección 7.2 de la memoria).

```
%-----  
% EXPERIMENTO NUMÉRICO 1: PROBLEMA DE VIBRACIONES LINEALES DE 1 GRADO DE  
% LIBERTAD.  
%  
% Este script recoge toda la información necesaria para computar las  
% soluciones analíticas y numéricas del ejemplo propuesto en 7.2,  
% correspondiente a un problema de vibraciones lineales de 1 grado de libertad  
% clásico o sistema "mkc", formulado en términos de un PVI lineal del tipo  
% (2.7).  
%  
% A diferencia de los demás experimentos numéricos propuestos, este puede  
% entenderse como un experimento "piloto", pues su razón de ser es la de  
% comprobar el orden empírico de los métodos numéricos para la obtención de  
% soluciones discretas expuestos en el Capítulo 3 (los empleados únicamente  
% para resolver PVI completos de la forma de (2.6) y (2.7)) a partir de la  
% notable ventaja de disponer de la solución analítica, siendo esta última de  
% obtención inmediata sin coste computacional asociado elevado.  
%  
% ORDEN EMPÍRICO DE MÉTODOS NUMÉRICOS PARA LA OBTENCIÓN DE SOLUCIONES  
% EN MALLA DISCRETA. MÉTODOS RUNGE-KUTTA Y RUNGE-KUTTA-NYSTRÖM PARA PVI DE  
% 2º ORDEN:  
% 1. RK Euler explícito (sección X.X de la memoria)  
% 2. RK Heun de 3º orden (sección X.X de la memoria)  
% 3. RK explícito de 4º orden (sección X.X de la memoria)  
% 4. RKN explícito directo de 3º orden (sección X.X de la memoria)  
% 5. RKN explícito transformado de 3º orden (sección X.X de la memoria)  
% 6. RKN explícito de 4º orden (sección X.X de la memoria)  
%  
% El código, con la forma de proceder, toma la forma siguiente:  
%-----  
clear all  
format long  
tic  
% Valores numéricos considerados en la Tabla (7.2):  
m = 1; k = 5; c = 0.5; Q = 10; w = 3; x0 = 0; dx0 = 0;  
% Valores numéricos de las agrupaciones consideradas en la Tabla (7.2):  
w = sqrt(k/m); n = c/(2*m); q = Q/m; M = q*(w^2-w^2)/((w^2-w^2)^2+4*n^2*w^2);  
N = 2*q*n*w/((w^2-w^2)^2+4*n^2*w^2); wd = sqrt(w^2-n^2); C1=x0-M;  
C2=(dx0+C1*n-N*w)/wd;
```



```

% Construcción de los nodos t de las distintas mallas equiespaciadas en los
% que se pretende evaluar la solución del PVI (7.6).
a = 0; b = 3;
h = [0.1, 0.05, 0.025, 0.0125, 0.00625];
t1 = transpose(a:h(1):b); %Malla 1
t2 = transpose(a:h(2):b); %Malla 2
t3 = transpose(a:h(3):b); %Malla 3
t4 = transpose(a:h(4):b); %Malla 4
t5 = transpose(a:h(5):b); %Malla 5
% Solución discreta exacta (7.7), primera derivada y coeficientes del PVI
% (2.7) en forma normal:
x = @(t) exp(-n*t).*(C1*cos(wd*t)+C2*sin(wd*t))+M*cos(W*t)+N*sin(W*t);
dx = @(t) exp(-n*t).*((C2*wd-n*C1)*cos(wd*t)-
(C2+C1*wd)*sin(wd*t))+w*(N*cos(W*t)-M*sin(W*t));
p = -2*n; q = -w^2; r = @(t) q*cos(W*t);
%-----
% ORDEN EMPÍRICO DE MÉTODOS NUMÉRICOS PARA LA OBTENCIÓN DE SOLUCIONES
% EN MALLA DISCRETA. MÉTODOS RUNGE-KUTTA Y RUNGE-KUTTA-NYSTRÖM PARA PVI DE
% 2º ORDEN:
%-----
% Gráfica de la solución analítica del PVI lineal (7.6) x(t) empleando la
% malla t1 e interpolación lineal por defecto de MATLAB:
plot(t1,x(t1),'-o',t1,dx(t1),'-s');
xlabel('Nodos de la malla 1 (h=0.1)','Fontweight','bold');
legend('x(t)','dx(t1)','location','best');
grid on;
print -dpdf 'x(t1) y dx(t1).pdf'
% Construcción de la Tabla que almacena los valores de la solución
% discreta exacta en los nodos de la malla t1, para posteriormente
% compararlos con los valores obtenidos mediante métodos numéricos
% aproximados:
fileID = fopen('TablaSolucionExacta.txt','w');
fprintf(fileID,'%12s %12s\n','x(t1)','dx(t1)');
fprintf(fileID,'%6.3f\n',x(t1));
fprintf(fileID,'\n%6.3f\n',dx(t1));
fclose(fileID);
% Función f correspondiente al PVI lineal (7.6) expresado en forma normal:
f = @(t,x,dx) -0.5*dx-5*x+10*cos(3*t);
% MALLA t1. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos:
xERKt1 = ERK1s1o(t1,f,x0,dx0); %1
xHRKt1 = HRK3s1o(t1,f,x0,dx0); %2
xRK4t1 = RK4s1o(t1,f,x0,dx0); %3
xRKN3t1 = RKN3(t1,f,x0,dx0); %4
xRKN3tt1 = RKN3t(t1,f,x0,dx0); %5
xRKN4t1 = RKN4(t1,f,x0,dx0); %6
% Gráfica del |EGT| respecto de la solución exacta x(t) para el método de
% Euler y la malla t1:
plot(t1,abs(xERKt1-x(t1)),'-x','MarkerSize',10);
xlabel('Nodos de la malla 1 (h1 = 0.1)','Fontweight','bold');
ylabel('|EGT|','Fontweight','bold');
grid on;
print -dpdf 'Error_Euler_malla_t1.pdf'
% Gráfica del |EGT| respecto de la solución exacta x(t) para cada uno de
los métodos propuestos exclusivamente de 3º orden y la malla t1:

```

```
plot(t1,abs(xHRKt1-x(t1)),'-o', t1,abs(xRKN3t1-x(t1)),'-
s',t1,abs(xRKN3tt1-x(t1)),'-d');
legend({'|EGT| HRK3s1o','|EGT| RKN3','|EGT|
RKN3t'}, 'location','best','Fontweight','bold');
xlabel('Nodos de la malla 1 (h1 = 0.1)','Fontweight','bold');
ylabel('|EGT|','Fontweight','bold');
grid on;
print -dpdf 'Error_métodos_malla_t1_orden3.pdf'
% Gráfica del |EGT| respecto de la solución exacta x(t) para los métodos
% propuestos exclusivamente de 4º orden y la malla t1:
plot(t1,abs(xRK4t1-x(t1)),'-^',t1,abs(xRKN4t1-x(t1)),'-*');
legend({'|EGT| RK4s1o','|EGT| RKN4'}, 'location','best','Fontweight','bold');
xlabel('Nodos de la malla 1 (h1 = 0.1)','Fontweight','bold');
ylabel('|EGT|','Fontweight','bold');
grid on;
print -dpdf 'Error_metodos_malla_t1_orden4.pdf'
% MALLA t2. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos (Gráficas de la solución no obtenidas):
xERKt2 = ERK1s1o(t2,f,x0,dx0); %1
xHRKt2 = HRK3s1o(t2,f,x0,dx0); %2
xRK4t2 = RK4s1o(t2,f,x0,dx0); %3
xRKN3t2 = RKN3(t2,f,x0,dx0); %4
xRKN3tt2 = RKN3t(t2,f,x0,dx0); %5
xRKN4t2 = RKN4(t2,f,x0,dx0); %6
% MALLA t3. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos (Gráficas de la solución no obtenidas):
xERKt3 = ERK1s1o(t3,f,x0,dx0); %1
xHRKt3 = HRK3s1o(t3,f,x0,dx0); %2
xRK4t3 = RK4s1o(t3,f,x0,dx0); %3
xRKN3t3 = RKN3(t3,f,x0,dx0); %4
xRKN3tt3 = RKN3t(t3,f,x0,dx0); %5
xRKN4t3 = RKN4(t3,f,x0,dx0); %6
% MALLA t4. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos (Gráficas de la solución no obtenidas):
xERKt4 = ERK1s1o(t4,f,x0,dx0); %1
xHRKt4 = HRK3s1o(t4,f,x0,dx0); %2
xRK4t4 = RK4s1o(t4,f,x0,dx0); %3
xRKN3t4 = RKN3(t4,f,x0,dx0); %4
xRKN3tt4 = RKN3t(t4,f,x0,dx0); %5
xRKN4t4 = RKN4(t4,f,x0,dx0); %6
% MALLA t5. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos (Gráficas de la solución no obtenidas):
xERKt5 = ERK1s1o(t5,f,x0,dx0); %1
xHRKt5 = HRK3s1o(t5,f,x0,dx0); %2
xRK4t5 = RK4s1o(t5,f,x0,dx0); %3
xRKN3t5 = RKN3(t5,f,x0,dx0); %4
xRKN3tt5 = RKN3t(t5,f,x0,dx0); %5
xRKN4t5 = RKN4(t5,f,x0,dx0); %6
% Construcción de la Tabla que almacena los valores del |EGT| respecto de la
% solución exacta x(t) en el último nodo de cada una de las mallas y para cada
% uno
% de los métodos numéricos utilizados.
EGTend = abs([xERKt1(end)-x(t1(end)), xHRKt1(end)-x(t1(end)), xRK4t1(end)-
x(t1(end)), xRKN3t1(end)-x(t1(end)), xRKN3tt1(end)-x(t1(end)), xRKN4t1(end)-
x(t1(end));
xERKt2(end)-x(t2(end)), xHRKt2(end)-x(t2(end)), xRK4t2(end)-x(t2(end)),
```



```

xRKN3t2(end)-x(t2(end)), xRKN3tt2(end)-x(t2(end)), xRKN4t2(end)-x(t2(end));
  xERKt3(end)-x(t3(end)), xHRKt3(end)-x(t3(end)), xRK4t3(end)-x(t3(end)),
xRKN3t3(end)-x(t3(end)), xRKN3tt3(end)-x(t3(end)), xRKN4t3(end)-x(t3(end));
  xERKt4(end)-x(t4(end)), xHRKt4(end)-x(t4(end)), xRK4t4(end)-x(t4(end)),
xRKN3t4(end)-x(t4(end)), xRKN3tt4(end)-x(t4(end)), xRKN4t4(end)-x(t4(end));
  xERKt5(end)-x(t5(end)), xHRKt5(end)-x(t5(end)), xRK4t5(end)-x(t5(end)),
xRKN3t5(end)-x(t5(end)), xRKN3tt5(end)-x(t5(end)), xRKN4t5(end)-x(t5(end))]];
fileID = fopen('TablaEGTdiscretos.csv','w');
fprintf(fileID,'%12s %12s %12s %12s %12s
%12s\n','ERK1s1o','HRK3s1o','RK4s1o','RKN3','RKN3t','RKN4');
fprintf(fileID,'%6.3e %6.3e %6.3e %6.3e %6.3e %6.3e\n',transpose(EGTend));
fclose(fileID);
% Gráfica |log2(|EGT_end|)| vs. |log2(h)| para estimar empíricamente el orden
de % los métodos numéricos a partir de todas las mallas de ensayo:
log2h = abs(log2(h));
log2EGTend = abs(log2(EGTend));
plot(log2h,log2EGTend(:,1),'-x',log2h,log2EGTend(:,2),'-o', ...
      log2h,log2EGTend(:,4),'-s',log2h,log2EGTend(:,5),'-d', ...
      log2h,log2EGTend(:,3),'-^',log2h,log2EGTend(:,6),'-*');
legend({'EKR1s1o, p=0.974','HRK3s1o, p=2.945','RKN3, p=2.956','RKN3t,
p=2.954','RK4s1o, p=4.128','RKN4, p=4.014'}, ...
       'location','best','Fontweight','bold');
xlabel('|log_{2}(h)|','Fontweight','bold');
ylabel('|log_{2}|EGT||','Fontweight','bold');
grid on;
print -dpdf 'Orden_metodos.pdf'
% Zoom de la gráfica anterior para apreciar mejor las líneas correspondientes
% a los métodos HRK3s1o y RKN3t:Lin
plot(log2h,log2EGTend(:,4),'-s',log2h,log2EGTend(:,5),'-d');
legend({'HRK3s1o, p=2.945','RKN3t,
p=2.954'}, 'location','best','Fontweight','bold');
xlabel('|log_{2}(h)|','Fontweight','bold');
ylabel('|log_{2}|EGT||','Fontweight','bold');
grid on;
print -dpdf 'Orden_metodos_HRK3_y_RKN3t.pdf'
toc

```

Experimento numérico 2. Problemas de prueba. Órdenes de convergencia empíricos (sección 7.3 de la memoria).

```

%-----
% EXPERIMENTO NUMÉRICO 2: PROBLEMAS DE PRUEBA. ÓRDENES DE CONVERGENCIA
% EMPÍRICOS
%
% Este script recoge toda la información necesaria para computar las
% soluciones analíticas y numéricas de los PVI's propuestos en 7.2,
% correspondientes a sendos problemas que presentan soluciones analíticas
% polinómicas. El procedimiento de actuación se resume como sigue a
% continuación:
%
% ORDEN EMPÍRICO DE MÉTODOS NUMÉRICOS PARA LA OBTENCIÓN DE SOLUCIONES EN
% MALLA DISCRETA. MÉTODOS RUNGE-KUTTA Y RUNGE-KUTTA-NYSTRÖM PARA PVI DE 2º

```



```
% ORDEN:
% 1. RK de Gauss de orden 6 y tres etapas (sección 3.2.6 de la memoria)
% 2. RKN simple de orden 4 para problemas independientes de y' (sección
% 3.3.5.1 de la memoria)
% 3. RKN simple de orden 5 para problemas independientes de y' (sección
% 3.3.5.2 de la memoria)
% 4. RKN especial de orden 5 para problemas independientes de y'
% (sección 3.3.5.3 de la memoria)
%
% El código, con la forma de proceder, toma la forma siguiente:
%-----
clear all
format long
tic
% Construcción del PVI (7.10) completo con solución analítica polinómica
% quinta:
f1 = @(x,y,dy) x.*dy-5*y+x.^4+22*x.^3+15*x.^2+10*x+7;
y10 = 1; dy10 = 1;
y1 = @(x) x.^5+x.^4+x.^3+x.^2+x+1;
dy1 = @(x) 5*x.^4+4*x.^3+3*x.^2+2*x+1;
% Construcción del PVI (7.12) independiente de y' con solución analítica
% polinómica cuártica:
f2 = @(x,y,dy) y-x.^4-x.^3+11*x.^2+5*x+1;
y20 = 1; dy20 = 1;
y2 = @(x) x.^4+x.^3+x.^2+x+1;
dy2 = @(x) 4*x.^3+3*x.^2+2*x+1;
% Construcción de los nodos x de las distintas mallas equiespaciadas en los
% que se pretende evaluar la solución de los PVI anteriormente construidos:
a = 0; b = 2;
h = [0.4, 0.2, 0.1, 0.05, 0.025];
x1 = transpose(a:h(1):b); %Malla 1
x2 = transpose(a:h(2):b); %Malla 2
x3 = transpose(a:h(3):b); %Malla 3
x4 = transpose(a:h(4):b); %Malla 4
x5 = transpose(a:h(5):b); %Malla 5
%-----
% ORDEN EMPÍRICO DE MÉTODOS NUMÉRICOS PARA LA OBTENCIÓN DE SOLUCIONES
% EN MALLA DISCRETA. MÉTODOS RUNGE-KUTTA Y RUNGE-KUTTA-NYSTRÖM PARA PVI DE
% 2º ORDEN:
%-----
% Gráfica de las soluciones analíticas de los PVI empleando la
% malla x3 e interpolación lineal por defecto de MATLAB:
plot(x3,y1(x3),'-o',x3,y2(x3),'-s');
xlabel('Nodos de la malla x3 (h=0.1)','Fontweight','bold');
ylabel('y(x)','Fontweight','bold');
legend('y1(x3)','y2(x3)','Fontweight','bold','location','best');
grid on;
print -dpdf 'y1(x3) e y2(x3).pdf'
% Construcción de la Tabla que almacena los valores de la solución
% discreta exacta en los nodos de la malla x3, para posteriormente
% compararlos con los valores obtenidos mediante métodos numéricos
% aproximados:
fileID = fopen('TablaSolucionExactax3.txt','w');
fprintf(fileID,'%12s %12s\n','y1(x3)','y2(x3)');
fprintf(fileID,'%6.3f %6.3f\n',[transpose(y1(x3)); transpose(y2(x3))]);
```



```

fclose(fileID);
% 1. RK de Gauss de orden 6 y tres etapas. Soluciones discretas aproximadas
% que proporciona el método para el PVI completo con solución quintica y
% cada una de las mallas equiespaciadas de ensayo:
yGRK6x1 = GRK6s1o(x1,f1,y10,dy10); %Malla x1
yGRK6x2 = GRK6s1o(x2,f1,y10,dy10); %Malla x2
yGRK6x3 = GRK6s1o(x3,f1,y10,dy10); %Malla x3
yGRK6x4 = GRK6s1o(x4,f1,y10,dy10); %Malla x4
yGRK6x5 = GRK6s1o(x5,f1,y10,dy10); %Malla x5
% Gráfica del |EGT| respecto de la solución exacta y1(x) para el método
% GRK6s1o y la primera malla x1:
plot(x1,abs(yGRK6x1-y1(x1)),'-o');
xlabel('Nodos de la malla x1 (h=0.4)','FontWeight','bold');
ylabel('|EGT| malla x1','FontWeight','bold');
grid on;
print -dpdf 'EGT malla x1 GRK6.pdf'
% Gráfica del |EGT| respecto de la solución exacta y1(x) para el método
% GRK6s1o y la malla x2:
plot(x2,abs(yGRK6x2-y1(x2)),'-s');
xlabel('Nodos de la malla x2 (h=0.2)','FontWeight','bold');
ylabel('|EGT|','FontWeight','bold');
grid on;
print -dpdf 'EGT malla x2 GRK6.pdf'
% Gráfica del |EGT| respecto de la solución exacta y1(x) para el método
% GRK6s1o y el resto de mallas:
plot(x3,abs(yGRK6x3-y1(x3)),'-*',x4,abs(yGRK6x4-y1(x4)),'-
+',x5,abs(yGRK6x5-y1(x5)),'-^','MarkerSize',4);
xlabel('x','FontWeight','bold');
ylabel('|EGT|','FontWeight','bold');
legend('|EGT| malla x3 (h=0.1)', '|EGT| malla x4 (h=0.05)', 'EGT malla x5
(h=0.025)','FontWeight','bold','location','best');
grid on;
print -dpdf 'EGT multimalla GRK6.pdf'
% Construcción de la tabla que almacena los valores máximos del |EGT|
respecto de la solución exacta y1(x) en el nodo identificado automáticamente
de cada una de las mallas y exclusivamente para el método GRK6s1o:
EGTmaxG = abs([yGRK6x1(id_elemento_vector(abs(yGRK6x1-y1(x1)),max(abs(yGRK6x1-
y1(x1))))))-y1(x1(id_elemento_vector(abs(yGRK6x1-y1(x1)),max(abs(yGRK6x1-y1(x1))))));
yGRK6x2(id_elemento_vector(abs(yGRK6x2-y1(x2)),max(abs(yGRK6x2-y1(x2))))))-
y1(x2(id_elemento_vector(abs(yGRK6x2-y1(x2)),max(abs(yGRK6x2-y1(x2)))))); ...
yGRK6x3(id_elemento_vector(abs(yGRK6x3-y1(x3)),max(abs(yGRK6x3-y1(x3))))))-
y1(x3(id_elemento_vector(abs(yGRK6x3-y1(x3)),max(abs(yGRK6x3-y1(x3))))));
yGRK6x4(id_elemento_vector(abs(yGRK6x4-y1(x4)),max(abs(yGRK6x4-y1(x4))))))-
y1(x4(id_elemento_vector(abs(yGRK6x4-y1(x4)),max(abs(yGRK6x4-y1(x4))))));
yGRK6x5(id_elemento_vector(abs(yGRK6x5-y1(x5)),max(abs(yGRK6x5-y1(x5))))))-
y1(x5(id_elemento_vector(abs(yGRK6x5-y1(x5)),max(abs(yGRK6x5-y1(x5))))))]);
fileID = fopen('TablaEGTGRK6.csv','w');
fprintf(fileID,'%12s %12s %12s %12s %12s\n','Malla x1','Malla x2','Malla
x3','Malla x4','Malla x5');
fprintf(fileID,'%6.3e %6.3e %6.3e %6.3e %6.3e\n',EGTmaxG);
fclose(fileID);
% Gráfica |log2|EGT_end|| vs. |log2(h)| para estimar empíricamente el
orden del método GRK6s1o para las mallas en las que todavía no existe una
convergencia clara:
log2h = abs(log2(h(1:3)));
log2EGTmaxG = abs(log2(EGTmaxG(1:3)));
plot(log2h,log2EGTmaxG(1:3),'-*','MarkerSize',8);

```

```
legend('Orden empírico obtenido:
p=6.482', 'Fontweight', 'bold', 'location', 'northwest');
xlabel('|log_{2}(h)|', 'Fontweight', 'bold');
ylabel('|log_{2}|EGT||', 'Fontweight', 'bold');
grid on;
print -dpdf 'Orden empirico GRK6s1o.pdf'

% ORDEN EMPÍRICO DEL RESTO DE LOS MÉTODOS INDEPENDIENTES DE LA PRIMERA
% DERIVADA y' (2,3,4,5):
% Malla x1. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos:
yRKN4x1 = RKN4_no_dy(x1, f2, y20, dy20); %2
yRKN51x1 = RKN51_no_dy(x1, f2, y20, dy20); %3
yRKN52x1 = RKN52_no_dy(x1, f2, y20, dy20); %4
% Malla x2. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos:
yRKN4x2 = RKN4_no_dy(x2, f2, y20, dy20); %2
yRKN51x2 = RKN51_no_dy(x2, f2, y20, dy20); %3
yRKN52x2 = RKN52_no_dy(x2, f2, y20, dy20); %4
% Malla x3. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos:
yRKN4x3 = RKN4_no_dy(x3, f2, y20, dy20); %2
yRKN51x3 = RKN51_no_dy(x3, f2, y20, dy20); %3
yRKN52x3 = RKN52_no_dy(x3, f2, y20, dy20); %4
% Gráfica del |EGT| respecto de la solución exacta y2(x) para el método
% RKN4_no_dy y la malla x3:
plot(x3, abs(yRKN4x3-y2(x3)), '-x');
xlabel('Nodos de la malla x3 (h=0.1)', 'Fontweight', 'bold');
ylabel('|EGT|', 'Fontweight', 'bold');
grid on;
print -dpdf 'EGT RKN4_no_dy malla x3.pdf'
% Gráfica del |EGT| respecto de la solución exacta y2(x) para cada uno de
% los métodos propuestos exclusivamente de orden 5 y la malla x3:
plot(x3, abs(yRKN51x3-y2(x3)), '-o', x3, abs(yRKN52x3-y2(x3)), '-s');
legend('|EGT| RKN51', '|EGT| RKN52', 'Fontweight', 'bold', 'location', 'best');
xlabel('Nodos de la malla x3 (h=0.1)', 'Fontweight', 'bold');
ylabel('|EGT|', 'Fontweight', 'bold');
grid on;
print -dpdf 'EGT orden 5 malla x3.pdf'
% Malla x4. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos:
yRKN4x4 = RKN4_no_dy(x4, f2, y20, dy20); %2
yRKN51x4 = RKN51_no_dy(x4, f2, y20, dy20); %3
yRKN52x4 = RKN52_no_dy(x4, f2, y20, dy20); %4
% Malla x5. Soluciones discretas aproximadas para cada uno de los métodos
% propuestos:
yRKN4x5 = RKN4_no_dy(x5, f2, y20, dy20); %2
yRKN51x5 = RKN51_no_dy(x5, f2, y20, dy20); %3
yRKN52x5 = RKN52_no_dy(x5, f2, y20, dy20); %4
% Construcción de la tabla que almacena los valores del |EGT| respecto de la
% solución exacta en el último nodo de cada una de las mallas y para cada
% uno de los métodos numéricos utilizados en el problema independiente de y':
EGTend_no_dy = abs([yRKN4x1(end)-y2(x1(end)), yRKN51x1(end)-y2(x1(end)), yRKN52x1(end)-
y2(x1(end)); ...
yRKN4x2(end)-y2(x2(end)), yRKN51x2(end)-y2(x2(end)), yRKN52x2(end)-y2(x2(end)); ...
yRKN4x3(end)-y2(x3(end)), yRKN51x3(end)-y2(x3(end)), yRKN52x3(end)-y2(x3(end)); ...
yRKN4x4(end)-y2(x4(end)), yRKN51x4(end)-y2(x4(end)), yRKN52x4(end)-y2(x4(end)); ...
```



```

yRKN4x5(end)-y2(x5(end)),yRKN51x5(end)-y2(x5(end)),yRKN52x5(end)-y2(x5(end))]);
fileID = fopen('TablaEGTendRKNnody.csv','w');
fprintf(fileID,'%12s %12s %12s\n','RKN4','RKN51','RKN52');
fprintf(fileID,'%6.3e %6.3e %6.3e\n',transpose(EGTend_no_dy));
fclose(fileID);
% Gráfica |log2|EGT_end| vs. |log2(h)| para estimar empíricamente el orden
% de los métodos numéricos a partir de todas las mallas de ensayo:
log2ht = abs(log2(h));
log2EGTend_no_dy = abs(log2(EGTend_no_dy));
plot(log2ht,log2EGTend_no_dy(:,1),'-o',log2ht,log2EGTend_no_dy(:,2),'-
s',log2ht,log2EGTend_no_dy(:,3),'-d');
legend('RKN4-no-dy, p=3.931','RKN51-no-dy, p=4.962','RKN52-no-dy,
p=4.826','location','best','Fontweight','bold');
xlabel('|log_{2}(h)|','Fontweight','bold');
ylabel('|log_{2}|EGT||','Fontweight','bold');
grid on;
print -dpdf 'Orden_metodos_RKN4y5.pdf'
toc

```

Experimento numérico 3. Problema de prueba. Ensayos con *splines* cúbicos (sección 7.4 de la memoria).

```

%-----
% EXPERIMENTO NUMÉRICO 3: PROBLEMA DE PRUEBA. ENSAYOS CON SPLINES CÚBICOS
%
% Este script recoge toda la información necesaria para computar los
% ensayos numéricos realizados con los trazadores polinomiales cúbicos del
% Capítulo 4 y documentados en la sección 7.4 del Capítulo 7 empleando para
% ello el PVI lineal (7.14), con solución polinómica cúbica dada por (7.15).
%
% El código, con la forma de proceder, toma la forma siguiente:
%-----
clear all
format long
tic
% Construcción del PVI (7.14) con solución analítica polinómica cúbica:
f = @(x,y,dy) x.*dy-3*y+x.^2+8*x+5;
y0 = 1; dy0 = 1;
y = @(x) x.^3+x.^2+x+1;
dy = @(x) 3*x.^2+2*x+1;
% Creación de las mallas equiespaciadas y con control del densificado
% definidas en la Tabla 7.21:
a = 0; b = 1; %Extremos del intervalo [0,1]
n = [4, 8, 12, 16, 29, 32]; %Subintervalos de cada malla
h = [(b-a)/n(1), (b-a)/n(2), (b-a)/n(3), (b-a)/n(4), (b-a)/n(5), (b-a)/n(6)];
%Pasos uniformes
x1 = transpose(a:h(1):b); %Malla x1
x2 = transpose(a:h(2):b); %Malla x2
x3 = transpose(a:h(3):b); %Malla x3
x4 = transpose(a:h(4):b); %Malla x4
x5 = transpose(a:h(5):b); %Malla x5
x6 = transpose(a:h(6):b); %Malla x6
% Gráfica de la solución analítica del PVI (7.14) en los nodos de la malla x4:

```

```
plot(x4,y(x4),'-o');
xlabel('Nodos de la malla x4 (h=0.0625)','FontWeight','bold');
ylabel('y(x)','FontWeight','bold');
grid on;
print -dpdf 'y(x4).pdf'
% Creación de la malla densificada para las posteriores evaluaciones de los
% trazadores:
xdensa = transpose(a:h/250:b);
% EXPERIMENTACIÓN CON SPLINES CÚBICOS EMPLEANDO EL PVI (7.14)
% Cálculo de los coeficientes del spline cúbico que interpola a la
% solución discreta analítica del PVI (7.14) en los nodos de cada malla
% definida previamente:
s3int1 = spline3int(x1,y(x1),f(x1(1),y0,dy0),dy0); %Malla x1
s3int2 = spline3int(x2,y(x2),f(x2(1),y0,dy0),dy0); %Malla x2
s3int3 = spline3int(x3,y(x3),f(x3(1),y0,dy0),dy0); %Malla x3
s3int4 = spline3int(x4,y(x4),f(x4(1),y0,dy0),dy0); %Malla x4
s3int5 = spline3int(x5,y(x5),f(x5(1),y0,dy0),dy0); %Malla x5
s3int6 = spline3int(x6,y(x6),f(x6(1),y0,dy0),dy0); %Malla x6
% Construcción de la tabla que almacena los valores del coeficiente an3
% correspondiente al spline cubico computado para cada una de las
% mallas de ensayo y la solución analítica en los nodos del último
intervalo:
xends = [x1(end-1); x1(end);
         x2(end-1); x2(end);
         x3(end-1); x3(end);
         x4(end-1); x4(end);
         x5(end-1); x5(end);
         x6(end-1); x6(end)];
an3end = [s3int1(end); s3int2(end); s3int3(end);
s3int4(end); s3int5(end); s3int6(end)];
fileID = fopen('TablaAN3.txt','w');
fprintf(fileID,'%25s %12s\n','Nodos [tn-1,tn]','y(x)');
fprintf(fileID,'%6.15f %6.15f\n',[transpose(xends); y(transpose(xends))]);
fprintf(fileID,'%12s\n','an3');
fprintf(fileID,'%6.15f\n',an3end);
fclose(fileID);
% Gráfica de la solución interpolada con el trazador s3int3 y en el último
intervalo:
plot(x3,y(x3),'o',xdensa,ppval(mkpp(x3,flip(s3int3,2)),xdensa),xdensa,y(xdensa));
xlabel('Nodos de la malla x3 (h=0.083)','FontWeight','bold');
ylabel('y(x)','FontWeight','bold');
xlim([x3(end-1),x3(end)]);
legend('y(x3)','s3int3','y(x)','FontWeight','bold','location','best');
grid on;
print -dpdf 's3int3.pdf'
% Gráfica de la solución interpolada con el trazador s3int5 y en el último
intervalo:
plot(x5,y(x5),'o',xdensa,ppval(mkpp(x5,flip(s3int5,2)),xdensa),xdensa,y(xdensa));
xlabel('Nodos de la malla x5 (h=0.033)','FontWeight','bold');
ylabel('y(x)','FontWeight','bold');
xlim([x5(end-1),x5(end)]);
legend('y(x5)','s3int5','y(x)','FontWeight','bold','location','best');
grid on;
print -dpdf 's3int5.pdf'
toc
```



Experimento numérico 4. Deflexión de una viga biapoyada (sección 7.5 de la memoria).

```
%-----  
% EXPERIMENTO NUMÉRICO 4: DEFLEXIÓN DE UNA VIGA BIAPOYADA.  
%  
% Este script recoge toda la información necesaria para computar las  
% soluciones numéricas interpoladas del ejemplo propuesto en 7.5,  
% correspondiente a un problema de estructuras, en concreto la deflexión  
% de una viga biapoyada sometida a una determinada distribución de cargas.  
% Este problema se formula en términos de un PVF (7.26) de dos puntos lineal  
% del tipo (6.4) e independiente de la primera derivada. Se propone resolver  
% el problema mediante la aplicación del método del disparo lineal y  
% estrategias de interpolación de soluciones discretas basadas en:  
%  
% 1. Splines cuadráticos interpoladores (sección 4.3 de la memoria)  
% 2. Pseudo-splines cúbicos e interpolación de Hermite (sección 5.3.2 de  
% la memoria)  
% 3. Pseudo-splines quinticos interoladores obtenidos mediante polinomios  
% osculadores (sección 5.3.3 de la memoria)  
%  
% Las soluciones discretas aproximadas de los PVI en los que el PVF se  
% desdobra se obtienen mediante la utilización del método Runge-Kutta-Nyström  
% de orden 4 y 3 etapas (RKN4_no_dy, consultar 3.3.5.1) para problemas  
% independientes de la primera derivada, pues se trata de un problema de estas  
% características.  
%  
% El código, con la forma de proceder, toma la forma siguiente:  
%-----  
clear all  
format long  
tic  
% Nodos extremos del PVF y condiciones de contorno simples:  
a = 0; b = 8;  
v0 = 0; vL = -0.02;  
% Construcción del primer PVI (7.27). Problema no homogéneo:  
f1 = @(x,v1,dv1) 8./(210*(9*x+8)).*v1+256./(105.*pi^2*(9*x+8)).*sin(pi.*x/8);  
v10 = v0; dv10 = 0;  
% Construcción del segundo PVI (7.28). Problema homogéneo:  
f2 = @(x,v2,dv2) 8./(210*(9*x+8)).*v2;  
v20 = 0; dv20 = 1;  
%-----  
% 1. ESTRATEGIA DE RESOLUCIÓN MEDIANTE SPLINES CUADRÁTICOS INTERPOLADORES:  
%-----  
% Creación de la malla no equiespaciada. Interesa un densificado mayor en  
% las inmediaciones de los nodos inicial a=0 y extremo b=8 ya que se  
% sospecha de una deflexión más pronunciada al existir una configuración  
% biapoyada con libertad de giro:  
h11 = 0.5; x11t1 = transpose(a:h11:b/4);  
h12 = 1; x12t1 = transpose(a:h12:b/4);  
h21 = 1; x21t2 = transpose(b/4:h21:3*b/4);  
h22 = 0.25; x22t2 = transpose(b/4:h22:3*b/4);  
h31 = h11; x31t3 = transpose(3*b/4:h31:b);
```

```
h32 = h12; x32t3 = transpose(3*b/4:h32:b);
% Ensamblaje de todas las mallas en un único vector en orden creciente de nodos:
x1 = [x11t1(1:length(x11t1)-1); x21t2; x31t3(2:length(x31t3))]; %Malla x1
x2 = [x12t1(1:length(x12t1)-1); x22t2; x32t3(2:length(x32t3))]; %Malla x2
% Resolución del PVI (7.27):
[v11,dv11] = RKN4_no_dy(x1,f1,v10,dv10); %Solución discreta aprox. malla x1
s2int11 = flip(spline2int(x1,v11,dv10),2); %Coefs. del spline cuadrático
interp. malla x1
[v21,dv21] = RKN4_no_dy(x2,f1,v10,dv10); %Solución discreta aprox. malla x2
s2int21 = flip(spline2int(x2,v21,dv10),2); %Coefs del spline cuadrático
interp. malla x2
% Resolución del PVI (7.28):
[v12,dv12] = RKN4_no_dy(x1,f2,v20,dv20); %Solución discreta aprox. malla x1
s2int12 = flip(spline2int(x1,v12,dv20),2); %Coefs. del spline cuadrático
interp. malla x1
[v22,dv22] = RKN4_no_dy(x2,f2,v20,dv20); %Solución discreta aprox. malla x2
s2int22 = flip(spline2int(x2,v22,dv20),2); %Coefs. del spline cuadrático
interp. malla x2
% Obtención de la solución del PVF (7.26) mediante combinación de los
splines de cada PVI aplicando el método del disparo lineal:
xdensa = transpose(a:h22/200:b); %Malla densificada para evaluar todos los casos
v1 = v11 + (vL-v11(end))/v12(end)*v12; %Solución discreta del PVF malla x1
s2int1 = ppval(mkpp(x1,s2int11),xdensa)+(vL-
v11(end))/v12(end)*ppval(mkpp(x1,s2int12),xdensa); %Solución interp. del PVF
malla x1
v2 = v21 + (vL-v21(end))/v22(end)*v22; %Solución discreta del PVF malla x2
s2int2 = ppval(mkpp(x2,s2int21),xdensa)+(vL-
v21(end))/v22(end)*ppval(mkpp(x2,s2int22),xdensa); %Solución interp. del PVF
malla x2
% Gráfica de la solución interpolada malla x1
plot(x1,v1,'o',xdensa,s2int1);
legend('v(x1)', 's2int1', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 's2int1.pdf'
% Gráfica de la solución interpolada malla x2
plot(x2,v2,'o',xdensa,s2int2);
legend('v(x2)', 's2int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 's2int2.pdf'
% Gráfica comparativa entre mallas con zoom [3,5]:
plot(x1,v1,'o',x2,v2,'s',xdensa,s2int1,xdensa,s2int2);
legend('v(x1)', 'v(x2)', 's2int1', 's2int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
xlim([3,5]);
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 's2int1 y s2int2 [3,5].pdf'
% Gráfica comparativa entre mallas con zoom [7,8]:
plot(x1,v1,'o',x2,v2,'s',xdensa,s2int1,xdensa,s2int2);
legend('v(x1)', 'v(x2)', 's2int1', 's2int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
xlim([7,b]);
```




```
ylabel('v(x) (m)', 'FontWeight', 'bold');
grid on;
print -dpdf 's2int1 y s2int2 [7,8].pdf'
% Tabla de resultados numéricos para ambas mallas:
fileID = fopen('Tablasolnodales.txt', 'w');
fprintf(fileID, '%6s %6s %6s %6s\n', 'x11', 'v11', 'v12', 'v1');
fprintf(fileID, '%6.2f %6.3e %6.3e %6.3e\n', [transpose(x1);
transpose(v11); transpose(v12); transpose(v1)]);
fprintf(fileID, '%6s %6s %6s %6s\n', 'x12', 'v21', 'v22', 'v2');
fprintf(fileID, '%6.2f %6.3e %6.3e %6.3e\n', [transpose(x2);
transpose(v21); transpose(v22); transpose(v2)]);
fclose(fileID);

%-----
% 2. ESTRATEGIA DE RESOLUCIÓN MEDIANTE PSEUDO-SPLINES CÚBICOS E INTERPOLACIÓN
% DE HERMITE.
% 3. ESTRATEGIA DE RESOLUCIÓN MEDIANTE PSEUDO-SPLINES QUÍNICOS E INTERPOLACIÓN
% CON POLINOMIOS OSCULANTES.
% Ambas estrategias se aplican aprovechando el mismo código, pues se comparten
% mallas y sintaxis:
%-----
% Se consideran las mismas mallas que antes, para poder establecer criterios
comparativos
% Resolución del PVI 1 (solución discreta ya calculada):
ps3int11 = pspline3int(x1, v11, dv11); %Coefs. del pseudo-spline cúbico interp. malla x1
ps5int11 = pspline5int(x1, v11, dv11, f1(x1, v11, dv11)); %Coefs. del pseudo-spline
quintico interp. malla x1
ps3int21 = pspline3int(x2, v21, dv21); %Coefs. del pseudo-spline cúbico interp. malla x2
ps5int21 = pspline5int(x2, v21, dv21, f1(x2, v21, dv21)); %Coefs. del pseudo-spline
quintico interp. malla x2
% Resolución del PVI 2:
ps3int12 = pspline3int(x1, v12, dv12); %Coefs. del pseudo-spline cúbico interp. malla x1
ps5int12 = pspline5int(x1, v12, dv12, f2(x1, v12, dv12)); %Coefs. del pseudo-spline
cúbico interp. malla x1
ps3int22 = pspline3int(x2, v22, dv22); %Coefs. del pseudo-spline cúbico interp. malla x2
ps5int22 = pspline5int(x2, v22, dv22, f2(x2, v22, dv22)); %Coefs. del pseudo-spline
cúbico interp. malla x2
% Obtención de la solución del PVF mediante combinación de los pseudo-splines
de cada PVI aplicando el método del disparo lineal:
% Evaluación de PVIs para malla x1:
[n1, h1] = subint_pasos(x1); c = 100; %Control del densificado
xlocal1 = transpose(x1(1):h1(1)/c:x1(2)); %Malla local para evaluación inicializada
xdensa1 = xlocal1(1:length(xlocal1)-1); %Malla global densa inicializada
ps3int11_eval_k = eval_pspline3int(ps3int11(1,:), xlocal1, b); %PVI 1.
Primer intervalo evaluado
ps5int11_eval_k = eval_pspline5int(ps5int11(1,:), xlocal1, b);
ps3int11_eval = ps3int11_eval_k(1:end-1);
ps5int11_eval = ps5int11_eval_k(1:end-1);
ps3int12_eval_k = eval_pspline3int(ps3int12(1,:), xlocal1, b); %PVI 2.
Primer intervalo evaluado
ps5int12_eval_k = eval_pspline5int(ps5int12(1,:), xlocal1, b);
ps3int12_eval = ps3int12_eval_k(1:end-1);
ps5int12_eval = ps5int12_eval_k(1:end-1);
for i = 2:n1 %Evaluación del resto de intervalos para ambos PVI simultáneamente
xlocal1 = transpose(x1(i):h1(i)/c:x1(i+1));
ps3int11_eval_k = eval_pspline3int(ps3int11(i,:), xlocal1, b);
ps5int11_eval_k = eval_pspline5int(ps5int11(i,:), xlocal1, b);
```



```
ps3int12_eval_k = eval_pspline3int(ps3int12(i,:),xlocal1,b);
ps5int12_eval_k = eval_pspline5int(ps5int12(i,:),xlocal1,b);
if i == n1
    ps3int11_eval = [ps3int11_eval; ps3int11_eval_k(1:end)];
    ps5int11_eval = [ps5int11_eval; ps5int11_eval_k(1:end)];
    ps3int12_eval = [ps3int12_eval; ps3int12_eval_k(1:end)];
    ps5int12_eval = [ps5int12_eval; ps5int12_eval_k(1:end)];
    xdensa1 = [xdensa1; xlocal1(1:end)];
else
    ps3int11_eval = [ps3int11_eval; ps3int11_eval_k(1:end-1)];
    ps5int11_eval = [ps5int11_eval; ps5int11_eval_k(1:end-1)];
    ps3int12_eval = [ps3int12_eval; ps3int12_eval_k(1:end-1)];
    ps5int12_eval = [ps5int12_eval; ps5int12_eval_k(1:end-1)];
    xdensa1 = [xdensa1; xlocal1(1:end-1)];
end
end
% Evaluación de PVI para malla x2:
[n2,h2] = subint_pasos(x2);
xlocal2 = transpose(x2(1):h2(1)/c:x2(2)); %Malla local para evaluación inicializada
xdensa2 = xlocal2(1:length(xlocal2)-1); %Malla global densa inicializada
ps3int21_eval_k = eval_pspline3int(ps3int21(1,:),xlocal2,b); %PVI 1.
Primer intervalo evaluado
ps5int21_eval_k = eval_pspline5int(ps5int21(1,:),xlocal2,b);
ps3int21_eval = ps3int21_eval_k(1:end-1);
ps5int21_eval = ps5int21_eval_k(1:end-1);
ps3int22_eval_k = eval_pspline3int(ps3int22(1,:),xlocal2,b); %PVI 2.
Primer intervalo evaluado
ps5int22_eval_k = eval_pspline5int(ps5int22(1,:),xlocal2,b);
ps3int22_eval = ps3int22_eval_k(1:end-1);
ps5int22_eval = ps5int22_eval_k(1:end-1);
for i = 2:n2 %Evaluación del resto de intervalos para ambos PVI simultáneamente
    xlocal2 = transpose(x2(i):h2(i)/c:x2(i+1));
    ps3int21_eval_k = eval_pspline3int(ps3int21(i,:),xlocal2,b);
    ps5int21_eval_k = eval_pspline5int(ps5int21(i,:),xlocal2,b);
    ps3int22_eval_k = eval_pspline3int(ps3int22(i,:),xlocal2,b);
    ps5int22_eval_k = eval_pspline5int(ps5int22(i,:),xlocal2,b);
    if i == n2
        ps3int21_eval = [ps3int21_eval; ps3int21_eval_k(1:end)];
        ps5int21_eval = [ps5int21_eval; ps5int21_eval_k(1:end)];
        ps3int22_eval = [ps3int22_eval; ps3int22_eval_k(1:end)];
        ps5int22_eval = [ps5int22_eval; ps5int22_eval_k(1:end)];
        xdensa2 = [xdensa2; xlocal2(1:end)];
    else
        ps3int21_eval = [ps3int21_eval; ps3int21_eval_k(1:end-1)];
        ps5int21_eval = [ps5int21_eval; ps5int21_eval_k(1:end-1)];
        ps3int22_eval = [ps3int22_eval; ps3int22_eval_k(1:end-1)];
        ps5int22_eval = [ps5int22_eval; ps5int22_eval_k(1:end-1)];
        xdensa2 = [xdensa2; xlocal2(1:end-1)];
    end
end
end
% COMBINACIÓN FINAL DE PSEUDO-SPLINES PARA SOLUCIÓN DE PVF (7.26) POR DISPARO LINEAL:
ps3int1 = ps3int11_eval + (vL-v11(end))/v12(end)*ps3int12_eval; %Solución PVF malla x1 CÚBICOS
ps5int1 = ps5int11_eval + (vL-v11(end))/v12(end)*ps5int12_eval; %Solución PVF malla x1 QUÍNTICOS
```



```
ps3int2 = ps3int21_eval + (vL-v21(end))/v22(end)*ps3int22_eval; %Solución PVF
malla x2 CÚBICOS
ps5int2 = ps5int21_eval + (vL-v21(end))/v22(end)*ps5int22_eval; %Solución PVF
malla x2 QUÍNTICOS
% Gráfica de la solución interpolada malla x1 CÚBICOS:
plot(x1,v1,'o',xdensa1,ps3int1);
legend('v(x1)', 'ps3int1', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps3int1.pdf'
% Gráfica de la solución interpolada malla x1 QUÍNTICOS:
plot(x1,v1,'o',xdensa1,ps5int1);
legend('v(x1)', 'ps5int1', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps5int1.pdf'
% Gráfica de la solución interpolada malla x2 CÚBICOS:
plot(x2,v2,'o',xdensa2,ps3int2);
legend('v(x2)', 'ps3int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps3int2.pdf'
% Gráfica de la solución interpolada malla x2 QUÍNTICOS:
plot(x2,v2,'o',xdensa2,ps5int2);
legend('v(x2)', 'ps5int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps5int2.pdf'
% Gráfica comparativa entre mallas con zoom [3,5] CÚBICOS:
plot(x1,v1,'o',x2,v2,'s',xdensa1,ps3int1,xdensa2,ps3int2);
legend('v(x1)', 'v(x2)', 'ps3int1', 'ps3int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
xlim([3,5]);
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps3int1 y ps3int2 [3,5].pdf'
% Gráfica comparativa entre mallas con zoom [3,5] QUÍNTICOS:
plot(x1,v1,'o',x2,v2,'s',xdensa1,ps5int1,xdensa2,ps5int2);
legend('v(x1)', 'v(x2)', 'ps5int1', 'ps5int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
xlim([3,5]);
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps5int1 y ps5int2 [3,5].pdf'
% Gráfica comparativa entre mallas con zoom [7,8] CÚBICOS:
plot(x1,v1,'o',x2,v2,'s',xdensa1,ps3int1,xdensa2,ps3int2);
legend('v(x1)', 'v(x2)', 'ps3int1', 'ps3int2', 'Fontweight', 'bold', 'location', 'best');
xlabel('x (m)', 'Fontweight', 'bold');
xlim([7,b]);
ylabel('v(x) (m)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps3int1 y ps3int2 [7,8].pdf'
```

```
% Gráfica comparativa entre mallas con zoom [7,8] QUÍNTICOS:
plot(x1,v1,'o',x2,v2,'s',xdensa1,ps5int1,xdensa2,ps5int2);
legend('v(x1)', 'v(x2)', 'ps5int1', 'ps5int2', 'FontWeight', 'bold', 'location', 'best');
xlabel('x (m)', 'FontWeight', 'bold');
xlim([7,b]);
ylabel('v(x) (m)', 'FontWeight', 'bold');
grid on;
print -dpdf 'ps5int1 y ps5int2 [7,8].pdf'

%-----
% OBTENCIÓN DE GRÁFICAS COMBINADAS PARA COMPARAR GRÁFICAMENTE TODAS LAS
% ESTRATEGIAS ENSAYADAS:
%-----

% Gráfica comparativa entre estrategias para malla x1 con zoom [3.6,4.4]:
plot(x1,v1,'o',xdensa,s2int1,xdensa1,ps3int1,xdensa1,ps5int1);
legend('v(x1)', 's2int1', 'ps3int1', 'ps5int1', 'FontWeight', 'bold', 'location', 'best');
xlabel('x (m)', 'FontWeight', 'bold');
xlim([3.6,4.4]);
ylabel('v(x) (m)', 'FontWeight', 'bold');
grid on;
print -dpdf 's2int1, ps3int1 y ps5int1.pdf'

% Valores numéricos de los trazadores en nodos intermedios a la
interpolación comunes a todas las estrategias para establecer criterios
comparativos:
valor_nodal1 = 3.65; %Para identificar el nodo común claramente
valor_nodal2 = 4.35; %Para identificar el nodo común claramente
fileID = fopen('v_nodales_malla1.txt', 'w');
fprintf(fileID, '%12s\n', 'Valor en nodo 3.65');
fprintf(fileID, '%6.5e\n', s2int1(id_elemento_vector(xdensa, valor_nodal1)));
fprintf(fileID, '%6.5e\n', ps3int1(id_elemento_vector(xdensa1, valor_nodal1)));
fprintf(fileID, '%6.5e\n', ps5int1(id_elemento_vector(xdensa1, valor_nodal1)));
fprintf(fileID, '%12s\n', 'Valor en nodo 4.35');
fprintf(fileID, '%6.5e\n', s2int1(id_elemento_vector(xdensa, valor_nodal2)));
fprintf(fileID, '%6.5e\n', ps3int1(id_elemento_vector(xdensa1, valor_nodal2)));
fprintf(fileID, '%6.5e\n', ps5int1(id_elemento_vector(xdensa1, valor_nodal2)));
fclose(fileID);

% Gráfica comparativa entre estrategias para malla x2 con zoom [3.6, 4.4]:
plot(x2,v2,'o',xdensa,s2int2,xdensa2,ps3int2,xdensa2,ps5int2);
legend('v(x2)', 's2int2', 'ps3int2', 'ps5int2', 'FontWeight', 'bold', 'location', 'best');
xlabel('x (m)', 'FontWeight', 'bold');
xlim([3.6,4.4]);
ylabel('v(x) (m)', 'FontWeight', 'bold');
grid on;
print -dpdf 's2int2, ps3int2 y ps5int2.pdf'

% Valores numéricos de los trazadores en nodos intermedios a la
interpolación comunes a todas las estrategias para establecer criterios
comparativos:
fileID = fopen('v_nodales_malla2.txt', 'w');
fprintf(fileID, '%12s\n', 'Valor en nodo 3.65');
fprintf(fileID, '%6.5e\n', s2int2(id_elemento_vector(xdensa, valor_nodal1)));
fprintf(fileID, '%6.5e\n', ps3int2(id_elemento_vector(xdensa2, valor_nodal1)));
fprintf(fileID, '%6.5e\n', ps5int2(id_elemento_vector(xdensa2, valor_nodal1)));
fprintf(fileID, '%12s\n', 'Valor en nodo 4.35');
fprintf(fileID, '%6.5e\n', s2int2(id_elemento_vector(xdensa, valor_nodal2)));
fprintf(fileID, '%6.5e\n', ps3int2(id_elemento_vector(xdensa2, valor_nodal2)));
fprintf(fileID, '%6.5e\n', ps5int2(id_elemento_vector(xdensa2, valor_nodal2)));
fclose(fileID);

toc
```



Experimento numérico 5. Perfil de temperaturas en una aleta anular hiperbólica (sección 7.6 de la memoria).

```

%-----
% EXPERIMENTO NUMÉRICO 5: PERFIL DE TEMPERATURAS EN UNA ALETA ANULAR
% HIPERBÓLICA
%
% Este script recoge toda la información necesaria para computar las
% soluciones numéricas interpoladas del ejemplo propuesto en 7.6,
% correspondiente a un problema de transmisión de calor, en concreto la
% distribución de temperaturas en una aleta anular de perfil hiperbólico
% sometida a unas determinadas condiciones de contorno. Este problema se
% formula en términos de un PVF (7.34) lineal del tipo (6.5), ya que
% presenta condiciones de contorno generales. Además, es independiente de
% la primera derivada. Se propone resolver el problema mediante la
% aplicación del método del disparo lineal y estrategias de interpolación
% de soluciones discretas basadas en pseudo-splines cúbicos (interpolación
% de Hermite, sección 5.3.2 del Capítulo 5).
%
% Es importante matizar que posteriormente al cómputo de las soluciones
% numéricas se hará un estudio de error cometido, pues el problema tiene
% solución exacta dada por funciones de Bessel modificadas de primera
% especie (7.35).
%
% El código, con la forma de proceder, toma la forma siguiente:
%-----

clear all
format long
tic
% Valores del problema particularizado (Unidades de SI):
r0 = 0.015; %Radio base de la aleta (m)
re = 0.05; %Radioe extremo de la aleta (m)
a = r0/re;
b = 7e-3; %Altura mitad base de la aleta (m)
k = 50.2; %Conductividad térmica (acero) (W/mK)
hconv = 200; %Coeficiente de película convectivo (W/m^2K)
M = sqrt(hconv*re^3/(k*b*r0)); %Número adimensional de Biot
% Nodos extremos del PVF y condiciones de contorno generales. Variables
% adimensionalizadas:
R0 = r0/re; Rf = 1;
p11 = 1; p12 = 0; p13 =1; %1ª CC. general
p21 = 0; p22 = 1; p23 = 0; %2ª CC. general
% Solución analítica y primera derivada para posteriores comprobaciones y
% estudios de errores:
D = (besseli(1/3,2/3*M*a^(3/2))*besseli(2/3,2/3*M)-besseli(-
1/3,2/3*M*a^(3/2))*besseli(-2/3,2/3*M));
T = @(R) sqrt(R/a).*(besseli(1/3,2/3*M*R.^(3/2))*besseli(2/3,2/3*M)-besseli(-
1/3,2/3*M*R.^(3/2))*besseli(-2/3,2/3*M))/D;
dT = @(R)
(sqrt(a*R)./(2*a*R).*(besseli(1/3,2/3*M*R.^(3/2)).*besseli(2/3,2/3*M)-
besseli(-1/3,2/3*M*R.^(3/2)).*besseli(-
2/3,2/3*M))+M*R/(2*sqrt(a)).*(besseli(2/3,2/3*M)*(besseli(-
2/3,2/3*M*R.^(3/2))+besseli(4/3,2/3*M*R.^(3/2)))-besseli(-

```

```
2/3,2/3*M)*(besseli(-4/3,2/3*M*R.^3/2))+besseli(2/3,2/3*M*R.^3/2))))/D;
% Creación de las mallas comunes a todos los PVI en los que el PVF se
desdobra:
% Malla 1 equiespaciada:
n1 = 5; h1 = (Rf-R0)/n1;
R1 = transpose(R0:h1:Rf);
% Malla 2 equiespaciada y muy poco densificada.
n2 = 2; h2 = (Rf-R0)/n2;
R2 = transpose(R0:h2:Rf);
% Malla 3 no equiespaciada: densificado local final para aproximar
% mejor la derivada que tiende a cero:
R08 = 0.8;
n31 = 2; h31 = (R08-R0)/n31; %Primer tramo equiespaciado [0.3,0.8]
R31 = transpose(R0:h31:R08);
n32 = 4; h32 = (Rf-R08)/n32; %Segundo tramo equiespaciado [0.8,1]
R32 = transpose(R08:h32:Rf);
R3 = [R31; R32(2:length(R32))];
% Función común a todos los PVI:
f = @(R,T,DT) M^2*R.*T;
% RESOLUCIÓN DEL PVI (7.36): NO PROCEDE AL TENER SOLUCIÓN NULA EN TODOS LOS
% NODOS
% RESOLUCIÓN DEL PVI (7.37) y almacenado :
T20 = 0; dT20 = 1; %Condiciones iniciales
[T21,dT21] = RKN51_no_dy(R1,f,T20,dT20); %Malla R1
ps3int21 = pspline3int(R1,T21,dT21); %Coefs. del pseudo-spline cúbico interp.
malla R1
[T22,dT22] = RKN51_no_dy(R2,f,T20,dT20); %Malla R2
ps3int22 = pspline3int(R2,T22,dT22); %Coefs. del pseudo-spline cúbico interp.
malla R2
[T23,dT23] = RKN51_no_dy(R3,f,T20,dT20); %Malla R3
ps3int23 = pspline3int(R3,T23,dT23); %Coefs. del pseudo-spline cúbico interp.
malla R2
% RESOLUCIÓN DEL PVI (7.38):
T30 = 1; dT30 = 0; %Condiciones iniciales
[T31,dT31] = RKN51_no_dy(R1,f,T30,dT30); %Malla R1
ps3int31 = pspline3int(R1,T31,dT31); %Coefs. del pseudo-spline cúbico interp.
malla R1
[T32,dT32] = RKN51_no_dy(R2,f,T30,dT30); %Malla R2
ps3int32 = pspline3int(R2,T32,dT32); %Coefs. del pseudo-spline cúbico interp.
malla R2
[T33,dT33] = RKN51_no_dy(R3,f,T30,dT30); %Malla R3
ps3int33 = pspline3int(R3,T33,dT33); %Coefs. del pseudo-spline cúbico interp.
malla R3
% Gráfica de la solución exacta con los nodos de las mallas creadas
% señalados:
Rdensa = transpose(R0:0.0001:Rf); %Malla auxiliar de evaluación
plot(Rdensa,T(Rdensa),R1,T(R1),'o',R2,T(R2),'s',R3,T(R3),'^');
xlabel('R','Fontweight','bold');
ylabel('\theta(R)','Fontweight','bold');
legend('\theta(R)','\theta(R1)','\theta(R2)','\theta(R3)','Fontweight','bold',
'location','best');
grid on;
print -dpdf 'Sol_exacta.pdf'
% Obtención de la solución del PVF mediante combinación lineal de los pseudo-
% splines de cada PVI aplicando el método del disparo lineal:
```



```
% Definición de los coeficientes de la combinación lineal:
d2 = @(a,b,c,d) (p13*(p21*a+p22*b)-p11*p23)/(p12*(p21*a+p22*b)-
p11*(p21*c+p22*d));
d3 = @(a,b,c,d) (-p13*(p21*c+p22*d)+p12*p23)/(p12*(p21*a+p22*b)-
p11*(p21*c+p22*d));
% Evaluación de pseudo-splines para malla R1:
[n1,h1] = subint_pasos(R1); c = 50; %Control del densificado
Rlocal1 = transpose(R1(1):h1(1)/c:R1(2)); %Malla local para evaluación
inicializada
Rdensa1 = Rlocal1(1:end-1); %Malla global densa inicializada
ps3int21_eval_k = eval_pspline3int(ps3int21(1,:),Rlocal1,Rf); %PVI (7.37).
Primer intervalo evaluado
ps3int21_eval = ps3int21_eval_k(1:end-1);
ps3int31_eval_k = eval_pspline3int(ps3int31(1,:),Rlocal1,Rf); %PVI (7.38).
Primer intervalo evaluado
ps3int31_eval = ps3int31_eval_k(1:end-1);
for i = 2:n1
    Rlocal1 = transpose(R1(i):h1(i)/c:R1(i+1));
    ps3int21_eval_k = eval_pspline3int(ps3int21(i,:),Rlocal1,Rf);
    ps3int31_eval_k = eval_pspline3int(ps3int31(i,:),Rlocal1,Rf);
    if i == n1
        ps3int21_eval = [ps3int21_eval; ps3int21_eval_k(1:end)];
        ps3int31_eval = [ps3int31_eval; ps3int31_eval_k(1:end)];
        Rdensa1 = [Rdensa1; Rlocal1(1:end)];
    else
        ps3int21_eval = [ps3int21_eval; ps3int21_eval_k(1:end-1)];
        ps3int31_eval = [ps3int31_eval; ps3int31_eval_k(1:end-1)];
        Rdensa1 = [Rdensa1; Rlocal1(1:end-1)];
    end
end
d21 = d2(T31(end),dT31(end),T21(end),dT21(end));
d31 = d3(T31(end),dT31(end),T21(end),dT21(end));
ps3int1 = d21*ps3int21_eval+d31*ps3int31_eval; %SOLUCIÓN INTERPOLADA MALLA R1
% Evaluación de pseudo-splines para malla R2:
[n2,h2] = subint_pasos(R2);
Rlocal2 = transpose(R2(1):h2(1)/c:R2(2)); %Malla local para evaluación
inicializada
Rdensa2 = Rlocal2(1:length(Rlocal2)-1); %Malla global densa inicializada
ps3int22_eval_k = eval_pspline3int(ps3int22(1,:),Rlocal2,Rf); %PVI (7.37).
Primer intervalo evaluado
ps3int22_eval = ps3int22_eval_k(1:length(ps3int22_eval_k)-1);
ps3int32_eval_k = eval_pspline3int(ps3int32(1,:),Rlocal2,Rf); %PVI (7.38).
Primer intervalo evaluado
ps3int32_eval = ps3int32_eval_k(1:length(ps3int32_eval_k)-1);
for i = 2:n2
    Rlocal2 = transpose(R2(i):h2(i)/c:R2(i+1));
    ps3int22_eval_k = eval_pspline3int(ps3int22(i,:),Rlocal2,Rf);
    ps3int32_eval_k = eval_pspline3int(ps3int32(i,:),Rlocal2,Rf);
    if i == n2
        ps3int22_eval = [ps3int22_eval; ps3int22_eval_k(1:end)];
        ps3int32_eval = [ps3int32_eval; ps3int32_eval_k(1:end)];
        Rdensa2 = [Rdensa2; Rlocal2(1:end)];
    else
        ps3int22_eval = [ps3int22_eval; ps3int22_eval_k(1:end-1)];
        ps3int32_eval = [ps3int32_eval; ps3int32_eval_k(1:end-1)];
    end
end
```

```
Rdensa2 = [Rdensa2; Rlocal2(1:end-1)];
end
end
d22 = d2(T32(end),dT32(end),T22(end),dT22(end));
d32 = d3(T32(end),dT32(end),T22(end),dT22(end));
ps3int2 = d22*ps3int22_eval+d32*ps3int32_eval; %SOLUCIÓN INTERPOLADA MALLA R2
% Evaluación de pseudo-splines para malla R3:
[n3,h3] = subint_pasos(R3);
Rlocal3 = transpose(R3(1):h3(1)/c:R3(2)); %Malla local para evaluación
inicializada
Rdensa3 = Rlocal3(1:end-1); %Malla global densa inicializada
ps3int23_eval_k = eval_p spline3int(ps3int23(1,:),Rlocal3,Rf); %PVI (7.37).
Primer intervalo evaluado
ps3int23_eval = ps3int23_eval_k(1:end-1);
ps3int33_eval_k = eval_p spline3int(ps3int33(1,:),Rlocal3,Rf); %PVI (7.38).
Primer intervalo evaluado
ps3int33_eval = ps3int33_eval_k(1:end-1);
for i = 2:n3
    Rlocal3 = transpose(R3(i):h3(i)/c:R3(i+1));
    ps3int23_eval_k = eval_p spline3int(ps3int23(i,:),Rlocal3,Rf);
    ps3int33_eval_k = eval_p spline3int(ps3int33(i,:),Rlocal3,Rf);
    if i == n3
        ps3int23_eval = [ps3int23_eval; ps3int23_eval_k(1:end)];
        ps3int33_eval = [ps3int33_eval; ps3int33_eval_k(1:end)];
        Rdensa3 = [Rdensa3; Rlocal3(1:end)];
    else
        ps3int23_eval = [ps3int23_eval; ps3int23_eval_k(1:end-1)];
        ps3int33_eval = [ps3int33_eval; ps3int33_eval_k(1:end-1)];
        Rdensa3 = [Rdensa3; Rlocal3(1:end-1)];
    end
end
end
d23 = d2(T33(end),dT33(end),T23(end),dT23(end));
d33 = d3(T33(end),dT33(end),T23(end),dT23(end));
ps3int3 = d23*ps3int23_eval+d33*ps3int33_eval; %SOLUCIÓN INTERPOLADA MALLA R3
% Solución discreta del PVF y primera derivada para posterior verificación de
% la interpolación mediante aplicación del método del disparo lineal y para
% cada una de las mallas:
T1 = d21*T21+d31*T31; dT1 = d21*dT21+d31*dT31; %Malla R1
T2 = d22*T22+d32*T32; dT2 = d22*dT22+d32*dT32; %Malla R2
T3 = d23*T23+d33*T33; dT3 = d23*dT23+d33*dT33; %Malla R2
% Tablas que almacenan los valores de las soluciones discretas aproximadas
% de los PVIs (7.37) y (7.38), así como la del PVF (7.34) obtenida por
% combinación de soluciones aplicando disparo lineal. Se crean 3 tablas, una
% para cada malla creada anteriormente.
% Malla R1
fileID = fopen('SOL1.txt','w');
fprintf(fileID,'%12s %12s %12s %12s\n','R1','PVI (7.37)','PVI (7.38)','PVF (7.34)');
fprintf(fileID,'%6.6f %6.6f %6.6f %6.6f\n',[transpose(R1); transpose(T21);
transpose(T31); transpose(T1)]);
fclose(fileID);
% Malla R2
fileID = fopen('SOL2.txt','w');
fprintf(fileID,'%12s %12s %12s %12s\n','R2','PVI (7.37)','PVI (7.38)','PVF (7.34)');
fprintf(fileID,'%6.6f %6.6f %6.6f %6.6f\n',[transpose(R2); transpose(T22);
transpose(T32); transpose(T2)]);
fclose(fileID);
% Malla R3
```




```
fileID = fopen('SOL3.txt','w');
fprintf(fileID,'%12s %12s %12s %12s\n','R3','PVI (7.37)','PVI (7.38)','PVF (7.34)');
fprintf(fileID,'%6.6f %6.6f %6.6f %6.6f\n',[transpose(R3); transpose(T23)];
transpose(T33); transpose(T3));
fclose(fileID);
% Tablas del |EGT| respecto de la solución analítica y su primera derivada
% que produce el método RKN51_no_dy para cada una de las mallas.
EGTR1 = abs(T(R1)-T1); EGTdR1 = abs(dT(R1)-dT1); %Malla R1
fileID = fopen('EGTR1.txt','w');
fprintf(fileID,'%12s %12s %12s %12s %12s\n','R1','T(R1)','EGTR1','dT(R1)','EGTdR1');
fprintf(fileID,'%6.2f %6.3e %6.3e %6.3e %6.3e\n',[transpose(R1); transpose(T(R1));
transpose(EGTR1); transpose(dT(R1)); transpose(EGTdR1)]);
fclose(fileID);
EGTR2 = abs(T(R2)-T2); EGTdR2 = abs(dT(R2)-dT2); %Malla R2
fileID = fopen('EGTR2.txt','w');
fprintf(fileID,'%12s %12s %12s %12s %12s\n','R2','T(R2)','EGTR2','dT(R2)','EGTdR2');
fprintf(fileID,'%6.2f %6.3e %6.3e %6.3e %6.3e\n',[transpose(R2); transpose(T(R2));
transpose(EGTR2); transpose(dT(R2)); transpose(EGTdR2)]);
fclose(fileID);
EGTR3 = abs(T(R3)-T3); EGTdR3 = abs(dT(R3)-dT3); %Malla R3
fileID = fopen('EGTR3.txt','w');
fprintf(fileID,'%12s %12s %12s %12s %12s\n','R3','T(R3)','EGTR3','dT(R3)','EGTdR3');
fprintf(fileID,'%6.2f %6.3e %6.3e %6.3e %6.3e\n',[transpose(R3); transpose(T(R3));
transpose(EGTR3); transpose(dT(R3)); transpose(EGTdR3)]);
fclose(fileID);
% Gráfica de la solución interpolada malla R1:
plot(R1,T1,'o',Rdensa1,ps3int1);
legend('\theta_{R1}','ps3int1','FontWeight','bold','location','best');
xlabel('R','FontWeight','bold');
ylabel('\theta_{R}','FontWeight','bold');
grid on;
print -dpdf 'ps3int1.pdf'
% Gráfica del error de interpolación respecto de la solución analítica del
% pseudo-spline cúbico interpolador de dicha solución en la malla R1:
plot(R1,EGTR1,'o',Rdensa1,abs(T(Rdensa1)-ps3int1));
legend('|EGT| "RKN51-no-dy" en nodos de R1','Error de
interpolación','FontWeight','bold','location','best');
xlabel('R','FontWeight','bold');
ylabel('Error','FontWeight','bold');
grid on;
print -dpdf 'Error_sol_exacta_R1.pdf'
% Gráfica de la solución interpolada malla R2:
plot(R2,T2,'o',Rdensa2,ps3int2);
legend('\theta_{R2}','ps3int2','FontWeight','bold','location','best');
xlabel('R','FontWeight','bold');
ylabel('\theta_{R}','FontWeight','bold');
grid on;
print -dpdf 'ps3int2.pdf'
% Gráfica del error de interpolación respecto de la solución analítica del
% pseudo-spline cúbico interpolador de dicha solución en la malla R2:
plot(R2,EGTR2,'o',Rdensa2,abs(T(Rdensa2)-ps3int2));
legend('|EGT| "RKN51-no-dy" en nodos de R2','Error de
interpolación','FontWeight','bold','location','best');
xlabel('R','FontWeight','bold');
ylabel('Error','FontWeight','bold');
grid on;
print -dpdf 'Error_sol_exacta_R2.pdf'
% Gráfica de la solución interpolada malla R3:
```



```
plot(R3,T3,'o',Rdensa3,ps3int3);
legend('\theta_{R3}','ps3int3','FontWeight','bold','location','best');
xlabel('R','FontWeight','bold');
ylabel('\theta_{R}','FontWeight','bold');
grid on;
print -dpdf 'ps3int3.pdf'
% Gráfica del error de interpolación respecto de la solución analítica del
% pseudo-spline cúbico interpolador de dicha solución en la malla R3:
plot(R3,EGTR3,'o',Rdensa3,abs(T(Rdensa3)-ps3int3));
legend('|EGT| "RKN51-no-dy" en nodos de R3','Error de
interpolación','FontWeight','bold','location','best');
xlabel('R','FontWeight','bold');
ylabel('Error','FontWeight','bold');
grid on;
print -dpdf 'Error_sol_exacta_R3.pdf'
toc
```

Experimento numérico 6. Respuesta temporal de un péndulo no lineal (sección 7.7 de la memoria).

```
%-----
% EXPERIMENTO NUMÉRICO 6: RESPUESTA TEMPORAL DE UN PÉNDULO NO LINEAL
%
% Este script recoge toda la información necesaria para computar las
% soluciones numéricas del ejemplo propuesto en 7.7, correspondiente a un
% péndulo no lineal sometido a una fuerza externa periódica. Este problema
% se formula en términos de un PVI del tipo (2.8) dado por (7.45),
% independiente de la primera derivada. Se propone realizar un estudio basado
% en la utilización de distintos solvers de EDOS de MATLAB para problemas no
% rígidos (Tabla 3.3) y su coste computacional asociado en la determinación de
% soluciones discretas aproximadas en mallas de nodos muy densas, junto con
% estrategias de interpolación ensayadas posteriormente que permitan disminuir
% dicho coste manteniendo el error acotado en valores fijados por el usuario.
%
% La estructura del código se detalla a continuación, con referencias a las
% secciones en la memoria donde puede consultarse el marco teórico de cada
% apartado:
%
% 1. EMPLEO DE SOLVERS DE MATLAB QUE IMPLEMENTAN PARES ENCAJADOS DE MÉTODOS
% RUNGE-KUTTA PARA PROBLEMAS NO RÍGIDOS CON ESTUDIO DEL COSTE COMPUTACIONAL
% ASOCIADO:
% 1.1 Solución numérica con solvers sin ajuste de tolerancias y cómputo
% automático del paso. Conclusiones sobre la calidad de la aproximación
% 1.2 Solución numérica con ajuste de tolerancias y cómputo automático del
% paso. Conclusiones sobre el coste computacional.
%
% 2. EMPLEO DE ESTRATEGIAS DE INTERPOLACIÓN PARA LA APROXIMACIÓN DE LA
% SOLUCIÓN DEL PVI CON DISMINUCIÓN DEL COSTE COMPUTACIONAL Y MONITORIZACIÓN
% DEL ERROR.
% 2.1. Empleo de splines cuadráticos (sección 4.3 de la memoria)
% 2.2. Empleo de pseudo-splines quinticos obtenidos mediante polinomios
% osciladores (sección 5.3.3 de la memoria)
```



```
%
% El código, con la forma de proceder, toma la forma siguiente:
%-----
clear all
format long
tic
% Datos del problema particularizado (Tabla 7.39):
l = 1; %Longitud de la varilla
g = 9.8; %Aceleración de la gravedad
w = 2; %Frecuencia de excitación
A = 0.7; %Amplitud de la excitación
u0 = 179*pi/180; %Condición inicial desplazamiento
du0 = 0; %Condición inicial velocidad
v0 = [u0,du0]; %Vector de condiciones iniciales
tf = 30; %Instante de tiempo final
tspan = [0,tf]; %Intervalo de tiempo del PVI (7.45)
% PVI (7.45) expresado como un sistema de EDOS de 1º orden del tipo (3.15):
f = @(t,u) [u(2); -g/l*sin(u(1))+A*cos(w*t)]; %u(2) es la primera derivada de
la solución y u(1) la solución
% 1. EMPLEO DE SOLVERS DE MATLAB QUE IMPLEMENTAN PARES ENCAJADOS DE MÉTODOS
% RUNGE-KUTTA PARA PROBLEMAS NO RÍGIDOS CON ESTUDIO DEL COSTE COMPUTACIONAL
% ASOCIADO:
% No se precisa de la creación de ninguna malla para obtener soluciones
% discretas como en anteriores experimentos al tener la capacidad los
% solvers de computarla automáticamente con las tolerancias fijas según lo
% indicado en la Figura 3.3.
% 1.1 Solución numérica con solvers sin ajuste de tolerancias y cómputo
% automático del paso. Conclusiones sobre la calidad de la aproximación.
% Por defecto, se emplea 'RelTol'=1e-3 y 'AbsTol'=1e-6 para computar el
% error en cada paso |e(i)| <= max(RelTol*y(i),AbsTol(i)).
[t23d,u23d] = ode23(f,tspan,v0); %Precisión baja. Malla t23d autocomputada
[t45d,u45d] = ode45(f,tspan,v0); %Precisión media. Malla t45d autocomputada
[t78d,u78d] = ode78(f,tspan,v0); %Precisión alta. Malla t78d autocomputada
[t89d,u89d] = ode89(f,tspan,v0); %Precisión alta. Malla t89d autocomputada
% Gráfica de resultados para cada uno de los solvers y mallas
% auto-computadas con los pares encajados:
plot(t23d,u23d(:,1),t45d,u45d(:,1),t78d,u78d(:,1),t89d,u89d(:,1));
legend('\theta(t) con ode23','\theta(t) con ode45','\theta(t) con
ode78','\theta(t) con ode89','Fontweight','bold','location','best');
xlabel('Tiempo t (s)','Fontweight','bold');
ylabel('\theta(t)','Fontweight','bold');
grid on;
print -dpdf 'solvers_tol_default_todo.pdf'
% Tabla que almacena el número de nodos no equiespaciados que cada solver
% ha computado automáticamente para satisfacer los criterios de tolerancia
por defecto de MATLAB:
TOLD = [length(t23d); length(t45d); length(t78d); length(t89d)];
fileID = fopen('Nodos_solvers_tol_default.txt','w');
fprintf(fileID,'%12s\n','Nodos');
fprintf(fileID,'%6.0f\n',TOLD);
fclose(fileID);
% 1.2 solución numérica con ajuste de tolerancias y cómputo automático del
% paso. Conclusiones sobre el coste computacional y análisis del error
% respecto al solver de mayor orden asumido como solución exacta.
options = odeset('RelTol',1e-6,'AbsTol',1e-6);
```

```
[t23m,u23m] = ode23(f,tspan,v0,options); %Precisión baja. Malla t23m autocomputada
[t45m,u45m] = ode45(f,tspan,v0,options); %Precisión media. Malla t45m autocomputada
[t78m,u78m] = ode78(f,tspan,v0,options); %Precisión alta. Malla t78m autocomputada
[t89m,u89m] = ode89(f,tspan,v0,options); %Precisión alta. Malla t89m autocomputada
% Gráfica de resultados para cada uno de los solvers y mallas
% auto-computadas con los pares encajados:
plot(t23m,u23m(:,1),t45m,u45m(:,1),t78m,u78m(:,1),t89m,u89m(:,1));
legend('\theta(t) con ode23','\theta(t) con ode45','\theta(t) con
ode78','\theta(t) con ode89','Fontweight','bold','location','best');
xlabel('Tiempo t (s)','Fontweight','bold');
ylabel('\theta(t)','Fontweight','bold');
grid on;
print -dpdf 'solvers_tol_mejoradas_todo.pdf'
% Gráficas del paso autocomputado por los pares encajados para
% satisfacer las tolerancias indicadas y determinar que las
% aproximaciones son mejores que las computadas con tolerancias por
% defecto:
[n23m,h23m] = subint_pasos(t23m); %Datos de malla t23m
plot(1:1:n23m,h23m);
xlabel('Subintervalos hasta n23m','Fontweight','bold');
ylabel('Paso h23m','Fontweight','bold');
xlim([1,n23m]);
xtickangle(45);
grid on;
print -dpdf 'pasosode23m.pdf'
[n45m,h45m] = subint_pasos(t45m); %Datos de malla t45m
plot(1:1:n45m,h45m);
xlabel('Subintervalos hasta n45m','Fontweight','bold');
ylabel('Paso h45m','Fontweight','bold');
xlim([1,n45m]);
xtickangle(45);
grid on;
print -dpdf 'pasosode45m.pdf'
[n78m,h78m] = subint_pasos(t78m); %Datos de malla t78m
plot(1:1:n78m,h78m);
xlabel('Subintervalos hasta n78m','Fontweight','bold');
ylabel('Paso h78m','Fontweight','bold');
xlim([1,n78m]);
xtickangle(45);
grid on;
print -dpdf 'pasosode78m.pdf'
[n89m,h89m] = subint_pasos(t89m); %Datos de malla t89m
plot(1:1:n89m,h89m);
xlim([1,n89m]);
xtickangle(45);
xlabel('Subintervalos hasta n89m','Fontweight','bold');
ylabel('Paso h89m','Fontweight','bold');
grid on;
print -dpdf 'pasosode89m.pdf'
% Tras asumir que ode89 proporciona la aproximación más exacta, se
% asume esta como si fuese la solución real del problema, parae obtener
% los errores cometidos respecto a ella por parte de los otros solvers
% utilizados. El Cómputo del error se realiza teniendo en cuenta la
% malla de nodos autocomputada para satisfacer el criterio con ode23,
% que es el solver que peor se comporta cualitativamente. Computando
```



```
% todas las soluciones nuevamente en esta malla conocida y las tolerancias
% impuestas manualmente, se tiene:
[t89m,u23e] = ode23(f,t89m,v0,options); %Precisión baja
[t89m,u45e] = ode45(f,t89m,v0,options); %Precisión media.
[t89m,u78e] = ode78(f,t89m,v0,options); %Precisión alta.
% Gráfica de las soluciones proporcionadas por los solvers en los nodos
% de malla t89m para posteriormente obtener los errores cometidos:
plot(t89m,u23e(:,1),t89m,u45e(:,1),t89m,u78e(:,1),t89m,u89m(:,1));
legend('\theta(t) con ode23','\theta(t) con ode45','\theta(t) con
ode78','\theta(t) con ode89','Fontweight','bold','location','best');
xlabel('Nodos de la malla t89m','Fontweight','bold');
ylabel('Error','Fontweight','bold');
grid on;
print -dpdf 'Solvers_malla_t89m_todo.pdf'
% Gráfica del error cometido respecto de la solución proporcionada por
% ode89 para cada uno de los solvers ensayados en la malla t89m:
plot(t89m,abs(u23e(:,1)-u89m(:,1)),t89m,abs(u45e(:,1)-
u89m(:,1)),t89m,abs(u78e(:,1)-u89m(:,1)));
legend('Error con ode23','Error con ode45','Error con
ode78','Fontweight','bold','location','best');
xlabel('Nodos de la malla t89m','Fontweight','bold');
ylabel('Error','Fontweight','bold');
grid on;
print -dpdf 'Errores_ode89_todos.pdf'
% Tabla que almacena las estadísticas más relevantes de cada malla
% autocomputada anteriormente:
TOLM = [length(t23m); length(t45m); length(t78m); length(t89m)]; %Número
de nodos de cada malla
minh = [min(h23m); min(h45m); min(h78m); min(h89m)]; %Paso mínimo de cada malla
posminh = [t23m(id_elemento_vector(h23m,min(h23m)));
t45m(id_elemento_vector(h45m,min(h45m)));
t78m(id_elemento_vector(h78m,min(h78m)));
t89m(id_elemento_vector(h89m,min(h89m)))]]; %Nodo extremo del subintervalo
en el que se alcanza el mínimo paso calculado previamente para cada malla
emax = [max(abs(u23e(:,1)-u89m(:,1)));
max(abs(u45e(:,1)-u89m(:,1)));
max(abs(u78e(:,1)-u89m(:,1))); 0]; %Errores máximos cometidos respecto a la
solución proporcionada por ode89 y la misma malla t89m
temax = [t89m(id_elemento_vector((abs(u23e(:,1)-u89m(:,1))),max(abs(u23e(:,1)-u89m(:,1)))));
t89m(id_elemento_vector((abs(u45e(:,1)-u89m(:,1))),max(abs(u45e(:,1)-u89m(:,1)))));
t89m(id_elemento_vector((abs(u78e(:,1)-u89m(:,1))),max(abs(u78e(:,1)-u89m(:,1))))); 0];
%Instante en el que el error se maximiza en la malla t89m
fileID = fopen('Estadisticas_mallas_solvers_mejoradas_malla_t89m.txt','w');
fprintf(fileID,'%12s %12s %12s %12s %12s\n','Nodos totales','Mínimo
h','t(minh)','emax_t89m','temax_t89m');
fprintf(fileID,'%6.0f %6.4f %6.4f %6.3e %6.4f\n',[transpose(TOLM); transpose(minh);
transpose(posminh); transpose(emax); transpose(temax)]);
fclose(fileID);
% 2. EMPLEO DE ESTRATEGIAS DE INTERPOLACIÓN PARA LA APROXIMACIÓN DE LA
% SOLUCIÓN DEL PVI CON DISMINUCIÓN DEL COSTE COMPUTACIONAL Y MONITORIZACIÓN
% DEL ERROR.
% Asumiendo que ode89 proporciona la solución discreta aproximada más
% exacta de todas, se sigue utilizando este solver para computar las
% soluciones que ahora se interpolarán empleando mallas mucho menos densas
% que las anteriores y trazadores polinomiales.
% 2.1. EMPLEO DE SPLINES CUADRÁTICOS INTERPOLADORES (sección 4.3 de la memoria):
% MALLA t1 equiespaciada y solución discreta en dichos nodos:
```

```
n1 = 30; h1 = tf/n1; t1 = transpose(0:h1:tf); c1 = 1000;
tdensa1 = 0:h1/c1:tf; %Densificado para posteriores evaluaciones
[tdensa1,u89densa1] = ode89(f,tdensa1,v0,options); %Solución discreta
aproximada muy densificada a evitar
u891 = zeros(length(t1),1); u89densa1_sol = u89densa1(:,1);
du891 = zeros(length(t1),1); du89densa1 = u89densa1(:,2);
for i = 1:n1+1
    u891(i) = u89densa1_sol(id_elemento_vector(tdensa1,t1(i))); %Solución
discreta aproximada en nodos de t1
    du891(i) = du89densa1(id_elemento_vector(tdensa1,t1(i))); %Primera
derivada aproximada en nodos de t1
end
s2int1 = flip(spline2int(t1,u891,du0),2); %Coeficientes del spline
cuadrático para malla t1
s2int1ev = ppval(mkpp(t1,s2int1),tdensa1); %Spline cuadrático para malla
t1 evaluado
% Gráfica de la solución interpolada y la proporcionada por ode89:
plot(t1,u891,'o',tdensa1,s2int1ev,tdensa1,u89densa1(:,1))
legend('\theta(t1)', 's2int1', '\theta(t)', 'FontWeight', 'bold', 'location', 'best');
xlabel('t (s)', 'FontWeight', 'bold');
ylabel('\theta(t)', 'FontWeight', 'bold');
grid on;
print -dpdf 's2int1.pdf'
% Gráfica del error respecto de la solución proporcionada por ode89:
plot(tdensa1,abs(s2int1ev-u89densa1(:,1)));
xlabel('t (s)', 'FontWeight', 'bold');
ylabel('Error', 'FontWeight', 'bold');
grid on;
print -dpdf 'error_s2int1.pdf'
% Tabla que almacena los valores u891 en los nodos de la malla t1
% que han sido utilizados para interpolar:
fileID = fopen('u891.txt', 'w');
fprintf(fileID, '%12s %12s\n', 'Nodos t1', 'u891');
fprintf(fileID, '%6.2f %6.4f\n', [transpose(t1); transpose(u891)]);
fclose(fileID);
% Malla t2 equiespaciada y solución discreta en dichos nodos:
n2 = 60; h2 = tf/n2; t2 = transpose(0:h2:tf); c2 = 1000;
tdensa2 = 0:h2/c2:tf; %Densificado para posteriores evaluaciones
[tdensa2,u89densa2] = ode89(f,tdensa2,v0,options); %Solución discreta
aproximada muy densificada a evitar
u892 = zeros(length(t2),1); u89densa2_sol = u89densa2(:,1);
du892 = zeros(length(t2),1); du89densa2 = u89densa2(:,2);
for i = 1:n2+1
    u892(i) = u89densa2_sol(id_elemento_vector(tdensa2,t2(i))); %Solución
discreta aproximada en nodos de t2
    du892(i) = du89densa2(id_elemento_vector(tdensa2,t2(i))); %Primera
derivada aproximada en nodos de t2
end
s2int2 = flip(spline2int(t2,u892,du0),2); %Coeficientes del spline
cuadrático para malla t2
s2int2ev = ppval(mkpp(t2,s2int2),tdensa2); %Spline cuadrático para malla
t2 evaluado
% Gráfica de la solución interpolada y la proporcionada por ode89:
plot(t2,u892,'o',tdensa2,s2int2ev,tdensa2,u89densa2(:,1))
legend('\theta(t2)', 's2int2', '\theta(t)', 'FontWeight', 'bold', 'location', 'best');
xlabel('t (s)', 'FontWeight', 'bold');
```



```
ylabel('\theta(t)', 'Fontweight', 'bold');
grid on;
print -dpdf 's2int2.pdf'
% Gráfica del error respecto de la solución proporcionada por ode89:
plot(tdensa2,abs(s2int2ev-u89densa2(:,1)));
xlabel('t (s)', 'Fontweight', 'bold');
ylabel('Error', 'Fontweight', 'bold');
grid on;
print -dpdf 'error_s2int2.pdf'
% Tabla que almacena los valores u891 en los nodos de la malla t1
% que han sido utilizados para interpolar:
fileID = fopen('u892.txt', 'w');
fprintf(fileID, '%12s %12s\n', 'Nodos t2', 'u892');
fprintf(fileID, '%6.2f %6.4f\n', [transpose(t2); transpose(u892)]);
fclose(fileID);
% Malla t3 EQUIESPACIADA y solución discreta en dichos nodos:
n3 = 120; h3 = tf/n3; t3 = transpose(0:h3:tf); c3 = 1000;
tdensa3 = 0:h3/c3:tf; %Densificado para posteriores evaluaciones
[tdensa3, u89densa3] = ode89(f, tdensa3, v0, options); %Solución discreta
aproximada muy densificada a evitar
u893 = zeros(length(t3), 1); u89densa3_sol = u89densa3(:, 1);
du893 = zeros(length(t3), 1); du89densa3 = u89densa3(:, 2);
for i = 1:n3+1
    u893(i) = u89densa3_sol(id_elemento_vector(tdensa3, t3(i))); %Solución
discreta aproximada en nodos de t3
    du893(i) = du89densa3(id_elemento_vector(tdensa3, t3(i))); %Primera
derivada aproximada en nodos de t3
end
s2int3 = flip(spline2int(t3, u893, du0), 2); %Coeficientes del spline
cuadrático para malla t3
s2int3ev = ppval(mkpp(t3, s2int3), tdensa3); %Spline cuadrático para malla
t3 evaluado
% Gráfica de la solución interpolada y la proporcionada por ode89:
plot(t3, u893, 'o', tdensa3, s2int3ev, tdensa3, u89densa3(:, 1))
legend('\theta(t3)', 's2int3', '\theta(t)', 'Fontweight', 'bold', 'location', 'best');
xlabel('t (s)', 'Fontweight', 'bold');
ylabel('\theta(t)', 'Fontweight', 'bold');
grid on;
print -dpdf 's2int3.pdf'
% Gráfica del error respecto de la solución proporcionada por ode89:
plot(tdensa3, abs(s2int3ev-u89densa3(:, 1)));
xlabel('t (s)', 'Fontweight', 'bold');
ylabel('Error', 'Fontweight', 'bold');
grid on;
print -dpdf 'error_s2int3.pdf'
% Tabla que almacena los valores u891 en los nodos de la malla t1
% que han sido utilizados para interpolar:
fileID = fopen('u893.txt', 'w');
fprintf(fileID, '%12s %12s\n', 'Nodos t3', 'u893');
fprintf(fileID, '%6.2f %6.4f\n', [transpose(t3); transpose(u893)]);
fclose(fileID);
% 2.1. EMPLEO DE PSEUDO-SPLINES QUÍNTICOS INTERPOLADORES (sección 5.3.3 de
la memoria). Se conservan las mallas anteriores y se realiza el mismo estudio
de error para observar las diferencias y las ventajas si es que existen:
% Función para aplicar colocación en segunda derivada:
fcol = @(t, u, du) -g/l*sin(u)+A*cos(W*t);
```

```
% MALLA t1: Creación y evaluación del pseudo-spline quintico interpolador
% de la solución discreta obtenida por ode89:
ps5int1 = pspline5int(t1,u891,du891,fcol(t1,u891,du891)); %Coeficientes
del pseudo-spline quintico en nodos de t1
ps5int1_eval_k = eval_pspline5int(ps5int1(1,:),tdensa1(1:c1+1),tf);
%Primer intervalo evaluado
ps5int1_eval = ps5int1_eval_k(1:end-1);
for i = 2:n1
    ps5int1_eval_k = eval_pspline5int(ps5int1(i,:),tdensa1(c1*(i-1)+1:c1*i+1),tf);
    if i == n1
        ps5int1_eval = [ps5int1_eval; ps5int1_eval_k];
    else
        ps5int1_eval = [ps5int1_eval; ps5int1_eval_k(1:end-1)];
    end
end

% Gráfica de la solución interpolada y la proporcionada por ode89:
plot(t1,u891,'o',tdensa1,ps5int1_eval,tdensa1,u89densa1(:,1));
Legend('\theta(t1)','ps5int1','\theta(t)','FontWeight','bold','location','best');
xlabel('t (s)','FontWeight','bold');
ylabel('\theta(t)','FontWeight','bold');
grid on;
print -dpdf 'ps5int1.pdf'
% Gráfica del error respecto de la solución proporcionada por ode89:
plot(tdensa1,abs(ps5int1_eval-u89densa1(:,1)));
xlabel('t (s)','FontWeight','bold');
ylabel('Error','FontWeight','bold');
grid on;
print -dpdf 'error_ps5int1.pdf'

% MALLA t2: Creación y evaluación del pseudo-spline quintico interpolador
ps5int2 = pspline5int(t2,u892,du892,fcol(t2,u892,du892)); %Coeficientes
del pseudo-spline quintico en nodos de t2
ps5int2_eval_k = eval_pspline5int(ps5int2(1,:),tdensa2(1:c2+1),tf);
%Primer intervalo evaluado
ps5int2_eval = ps5int2_eval_k(1:end-1);
for i = 2:n2
    ps5int2_eval_k = eval_pspline5int(ps5int2(i,:),tdensa2(c2*(i-1)+1:c2*i+1),tf);
    if i == n2
        ps5int2_eval = [ps5int2_eval; ps5int2_eval_k];
    else
        ps5int2_eval = [ps5int2_eval; ps5int2_eval_k(1:end-1)];
    end
end

% Gráfica de la solución interpolada y la proporcionada por ode89:
plot(t2,u892,'o',tdensa2,ps5int2_eval,tdensa2,u89densa2(:,1));
Legend('\theta(t2)','ps5int2','\theta(t)','FontWeight','bold','location','best');
xlabel('t (s)','FontWeight','bold');
ylabel('\theta(t)','FontWeight','bold');
grid on;
print -dpdf 'ps5int2.pdf'
% Gráfica del error respecto de la solución proporcionada por ode89:
plot(tdensa2,abs(ps5int2_eval-u89densa2(:,1)));
xlabel('t (s)','FontWeight','bold');
ylabel('Error','FontWeight','bold');
grid on;
print -dpdf 'error_ps5int2.pdf'

% MALLA t3: Creación y evaluación del pseudo-spline quintico interpolador
```




```
ps5int3 = pspline5int(t3,u893,du893,fc0l(t3,u893,du893)); %Coeficientes
del pseudo-spline quíntico en nodos de t3
ps5int3_eval_k = eval_pspline5int(ps5int3(1,:),tdensa3(1:c3+1),tf);
%primer intervalo evaluado
ps5int3_eval = ps5int3_eval_k(1:end-1);
for i = 2:n3
    ps5int3_eval_k = eval_pspline5int(ps5int3(i,:),tdensa3(c3*(i-1)+1:c3*i+1),tf);
    if i == n3
        ps5int3_eval = [ps5int3_eval; ps5int3_eval_k];
    else
        ps5int3_eval = [ps5int3_eval; ps5int3_eval_k(1:end-1)];
    end
end
% Gráfica de la solución interpolada y la proporcionada por ode89:
plot(t3,u893,'o',tdensa3,ps5int3_eval,tdensa3,u89densa3(:,1));
legend('\theta(t3)', 'ps5int3', '\theta(t)', 'Fontweight', 'bold', 'location', 'best');
xlabel('t (s)', 'Fontweight', 'bold');
ylabel('\theta(t)', 'Fontweight', 'bold');
grid on;
print -dpdf 'ps5int3.pdf'
% Gráfica del error respecto de la solución proporcionada por ode89:
plot(tdensa3,abs(ps5int3_eval-u89densa3(:,1)));
xlabel('t (s)', 'Fontweight', 'bold');
ylabel('Error', 'Fontweight', 'bold');
grid on;
print -dpdf 'error_ps5int3.pdf'
% Tabla que almacena las estadísticas más relevantes para cada una de
% las mallas y los trazadores polinomiales, tanto cuadráticos como
quínticos:
ntotal = [n1+1; n2+1; n3+1]; h = [h1; h2; h3]; %Nodos totales y pasos uniformes
emaxint = [max(abs(ps5int1_eval-u89densa1(:,1)));
    max(abs(ps5int2_eval-u89densa2(:,1)));
    max(abs(ps5int3_eval-u89densa3(:,1)))]]; %Error de interpolación máximo
respecto ode89
temaxint = [tdensa1(id_elemento_vector(abs(ps5int1_eval-
u89densa1(:,1)),max(abs(ps5int1_eval-u89densa1(:,1)))));
    tdensa2(id_elemento_vector(abs(ps5int2_eval-
u89densa2(:,1)),max(abs(ps5int2_eval-u89densa2(:,1)))));
    tdensa3(id_elemento_vector(abs(ps5int3_eval-
u89densa3(:,1)),max(abs(ps5int3_eval-u89densa3(:,1)))))]]; %Instante en el que
se alcanza el error máximo respecto de ode89
fileID = fopen('Estadísticas_mallas_interpolacion.txt','w');
fprintf(fileID,'%16s %16s %16s %16s\n', 'Nodos totales', 'Paso uniforme
h', 'emaxint', 'temaxint');
fprintf(fileID,'%6.0f %6.4f %6.3e %6.4f\n', [transpose(ntotal);
transpose(h); transpose(emaxint); transpose(temaxint)]);
fclose(fileID);
toc
```


Experimento numérico 7. PVI rígido (sección 7.8 de la memoria).

```
%-----  
% EXPERIMENTO NUMÉRICO 7: PROBLEMA DE VALORES INICIALES RÍGIDO  
%  
% Este script recoge toda la información necesaria para realizar el estudio  
% de reducción del coste computacional propuesto en la sección 7.8 del  
% Capítulo 7, tomando como punto de partida el PVI lineal rígido dado por  
% (7.48), del tipo (2.7) y homogéneo.  
%  
% Primero, se computan las soluciones discretas aproximadas utilizando el  
% solver "ode23s" indicado en la Tabla 3.4 del Capítulo 3, ya que es ideal  
% para problemas rígidos (stiff). Posteriormente, se calcula el |EGT| que  
% este método, con su malla y pasos auto computados al tratarse de un par  
% encajado de métodos numéricos Runge-Kutta, presenta respecto de la  
% solución exacta (conocida y dada por (7.49)) de (7.48).  
%  
% Después, habiendo fijado una tolerancia a cumplir por el |EGT|, se computa  
% la solución discreta con el mismo solver "ode23s" pero en una malla de  
% nodos mucho menos densa, para su posterior interpolación y determinación  
% de aproximación continua empleando pseudo-splines cúbicos basados en los  
% polinomios interpoladores de Hermite (sección 5.3.2 del Capítulo 5).  
%  
% Finalmente, se observa si la tolerancia fijada para el |EGT| de la solución  
% discreta proporcionada por el solver se cumple también para el error de  
% interpolación, llegando a conclusiones sobre la reducción del coste  
% computacional, algo que también se ha abordado en el Experimento numérico  
% 6 (sección 7.7).  
%  
% El código, con la forma de proceder, toma la forma siguiente:  
%-----  
clear all  
format long  
tic  
% Construcción del PVI (7.48) y de su solución:  
a = 0; b = 7; %Extremos del intervalo en el que el problema se define  
c = -1001; d = -1000.5;  
f = @(x,y) [y(2); c*y(2)+d*y(1)];  
y0 = 2; %1ª condición inicial  
dy0 = 0; %2ª condición inicial  
s1 = c/2+1/2*sqrt(c^2+4*d);  
s2 = c/2-1/2*sqrt(c^2+4*d);  
c1 = -2*s2/(s1-s2); %Constante c1 dependiente de las condiciones iniciales  
c2 = 2*s1/(s1-s2); %Constante c2 dependiente de las condiciones iniciales  
yex = @(x) c1*exp(s1*x)+c2*exp(s2*x); %Solución exacta de (7.48)  
dyex = @(x) s1*c1*exp(s1*x)+c2*s2*exp(s2*x); %Primera derivada de la solución  
% Gráfica de la solución exacta en un número suficiente de puntos para  
% poder apreciarse con nitidez:  
xex = transpose(a:0.01:b);  
plot(xex,yex(xex));  
xlabel('x','FontWeight','bold');  
ylabel('y(x)','FontWeight','bold');
```



```
grid on;
print -dpdf 'sol_stiff_exacta.pdf'
% Gráfica de la primera derivada exacta en un número suficiente de
% puntos para poder apreciarse con nitidez:
plot(xex,dyex(xex));
xlabel('x','Fontweight','bold');
ylabel("y'(x)",'Fontweight','bold');
grid on;
print -dpdf 'deriv_sol_stiff_exacta.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot(xex,dyex(xex));
xlabel('x','Fontweight','bold');
ylabel("y'(x)",'Fontweight','bold');
xlim([0,0.02]);
xtickangle(45);
grid on;
print -dpdf 'deriv_sol_stiff_exacta_zoom.pdf'
% Cómputo de las soluciones discretas aproximadas utilizando el solver
% "ode23s" con tolerancias por defecto "RelTol" = 1e-3 y "AbsTol" = 1e-6.
% Se computa también la malla discreta del tipo (2.11) en la que se
% almacenan los valores de la solución aproximada calculados.
[x23s,y23s] = ode23s(f,[a,b],[y0,dy0]);
% Gráfica de la solución discreta aproximada en los nodos de la malla
% auto computada por el solver:
plot(x23s,y23s(:,1),'-o');
xlabel('Nodos de la malla x23s','Fontweight','bold');
ylabel('y_{x23s}','Fontweight','bold');
grid on;
print -dpdf 'sol_stiff_ode23s.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot(x23s,y23s(:,1),'-o');
xlabel('Nodos de la malla x23s','Fontweight','bold');
ylabel('y_{x23s}','Fontweight','bold');
xlim([0,0.01]);
xtickangle(45);
grid on;
print -dpdf 'sol_stiff_ode23s_zoom.pdf'
% Gráfica de la evolución del paso de la malla "x23s":
[n23s,h23s] = subint_pasos(x23s);
plot([1:n23s],h23s,'-o');
xlabel('Subintervalos hasta n23s','Fontweight','bold');
ylabel('h23s','Fontweight','bold');
xlim([1,n23s]);
grid on;
print -dpdf 'h23s.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot([1:n23s],h23s,'-o');
xlabel('Subintervalos hasta n23s','Fontweight','bold');
ylabel('h23s','Fontweight','bold');
xlim([1,10]);
grid on;
print -dpdf 'h23s_zoom.pdf'
% Gráfica del |EGT| que proporciona el solver utilizado:
plot(x23s,abs(yex(x23s)-y23s(:,1)),'-o');
xlabel('Nodos de la malla x23s','Fontweight','bold');
```

```
ylabel('|EGT|','FontWeight','bold');
grid on;
print -dpdf 'EGT_ode23s.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot(x23s,abs(yex(x23s)-y23s(:,1)),'-o');
xlabel('Nodos de la malla x23s','FontWeight','bold');
ylabel('|EGT|','FontWeight','bold');
xlim([0,0.01]);
xtickangle(45);
grid on;
print -dpdf 'EGT_ode23s_zoom.pdf'
% Tabla que almacena los valores del |EGT| en cada uno de los nodos de la
malla auto computada x23s:
EGT23smax = max(abs(yex(x23s)-y23s(:,1))); %|EGT| máximo
x23smax = x23s(id_elemento_vector(abs(yex(x23s)-y23s(:,1)),EGT23smax));
%Nodo para |EGT| máximo
fileID = fopen('EGT23s.txt','w');
fprintf(fileID,'%12s %12s\n','Nodos x23s','EGT23s');
fprintf(fileID,'%6.3e %6.3e\n',[transpose(x23s); transpose(abs(yex(x23s)-
y23s(:,1)))]);
fprintf(fileID,'\n%12s %12s\n','Nodo','EGT max');
fprintf(fileID,'%6.3e %6.3e\n',[x23smax; EGT23smax]);
fclose(fileID);
% Interpolación de las soluciones discretas aproximadas proporcionadas por
% "ode23s" en malla indicada por el usuario explícitamente mediante pseudo-
splines cúbicos.
% Malla xint no equiespaciada con densificado en el entorno de x=0;
n1 = 10; a1 = 0.05; h1 = (a1-a)/n1;
x1 = transpose(a:h1:a1); %Primer tramo equiespaciado
n2 = 10; a2 = 2; h2 = (a2-a1)/n2;
x2 = transpose(a1:h2:a2); %Segundo tramo equiespaciado
n3 = 5; h3 = (b-a2)/n3;
x3 = transpose(a2:h3:b); %Tercer tramo equiespaciado
xint = [x1; x2(2:end-1); x3]; %Malla ensamblada
[xint,yx] = ode23s(f,xint,[y0,dy0]); %Soluciones discretas aproximadas en
malla xint
ps3int = pspline3int(xint,yx(:,1),yx(:,2)); %Coeficientes del pseudo-spline
cúbico interpolador
[nint,hint] = subint_pasos(xint); e = 100; %Control del densificado
xlocal = transpose(xint(1):hint(1)/e:xint(2)); %Malla local para evaluación
inicializada
xdensa = xlocal(1:end-1); %Malla global densa inicializada
ps3int_eval_k = eval_p spline3int(ps3int(1,:),xlocal,b); %Primer intervalo
evaluado
ps3int_eval = ps3int_eval_k(1:end-1);
for i = 2:nint
    xlocal = transpose(xint(i):hint(i)/e:xint(i+1));
    ps3int_eval_k = eval_p spline3int(ps3int(i,:),xlocal,b);
    if i == nint
        ps3int_eval = [ps3int_eval; ps3int_eval_k];
        xdensa = [xdensa; xlocal(1:end)];
    else
        ps3int_eval = [ps3int_eval; ps3int_eval_k(1:end-1)];
        xdensa = [xdensa; xlocal(1:end-1)];
    end
end
```



```
end
% Gráfica de la solución discreta aproximada e interpolada con el
% pseudo-spline cúbico ps3int:
plot(xint,yx(:,1),'o',xdensa,ps3int_eval);
legend('ode23s','ps3int','Fontweight','bold','location','best');
xlabel('x','Fontweight','bold');
ylabel('y(x)','Fontweight','bold');
grid on;
print -dpdf 'ps3int.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot(xint,yx(:,1),'o',xdensa,ps3int_eval);
legend('ode23s','ps3int','Fontweight','bold','location','best');
xlabel('x','Fontweight','bold');
ylabel('y(x)','Fontweight','bold');
xlim([0,0.01]);
xtickangle(45);
grid on;
print -dpdf 'ps3int_zoom.pdf'
% Gráfica del error cometido en la interpolación de soluciones
% discretas del PVI (7.48):
plot(xint,abs(yx(:,1)-yex(xint)),'o',xdensa,abs(ps3int_eval-yex(xdensa)));
legend('|EGT| ode23s','Error de
interpolación','Fontweight','bold','location','best');
xlabel('x','Fontweight','bold');
ylabel('Error','Fontweight','bold');
grid on;
print -dpdf 'error_ps3int.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot(xint,abs(yx(:,1)-yex(xint)),'o',xdensa,abs(ps3int_eval-yex(xdensa)));
legend('|EGT| ode23s','Error de
interpolación','Fontweight','bold','location','best');
xlabel('x','Fontweight','bold');
ylabel('Error','Fontweight','bold');
xlim([0,0.005]);
grid on;
print -dpdf 'error_ps3int_zoom1.pdf'
% Zoom de la gráfica anterior en el entorno de x=0:
plot(xint,abs(yx(:,1)-yex(xint)),'o',xdensa,abs(ps3int_eval-yex(xdensa)));
legend('|EGT| ode23s','Error de
interpolación','Fontweight','bold','location','best');
xlabel('x','Fontweight','bold');
ylabel('Error','Fontweight','bold');
xlim([0.005,0.01]);
grid on;
print -dpdf 'error_ps3int_zoom2.pdf'
% Tabla que almacena los valores del |EGT| en cada uno de los nodos de la
% malla xint:
EGTxdensamax = max(abs(ps3int_eval-yex(xdensa))); %Error de interpolación
máximo
xdensamax = xdensa(id_elemento_vector(abs(ps3int_eval-
yex(xdensa)),EGTxdensamax));
fileID = fopen('EGTxint.txt','w');
fprintf(fileID,'%12s %12s\n','Nodos xint','Error');
fprintf(fileID,'%6.3e %6.3e\n',[transpose(xint); transpose(abs(yx(:,1)-
yex(xint)))]);
```



```
fprintf(fileID, '\n%12s %12s\n', 'Nodo', 'EGTxintmax');  
fprintf(fileID, '%6.3e %6.3e\n', [transpose(xdensamax);  
transpose(EGTxdensamax)]);  
fclose(fileID);  
toc
```



ANEXO V. Funciones auxiliares.

Este último anexo contiene funciones auxiliares necesarias para dar mayor robustez a todos los códigos expuestos anteriormente y que pueden encontrarse llamadas en numerosas ocasiones.

- **Calculadora de pasos y número de subintervalos a partir de una malla de nodos determinada.**

```
function [n,h] = subint_pasos(x)
%-----
% La función "subint_pasos" obtiene el número de subintervalos contenidos
% en [a,b] a partir del vector de nodos de malla x. A su vez, también
% construye el paso variable h y lo almacena en un vector. Es de interés el
% empleo de esta función ya que su algoritmo se repite en numerosas ocasiones
%
% ENTRADAS DE LA FUNCIÓN:
%
% x = Vector columna que contiene los n+1 nodos distintos t y ordenados
% crecientemente, cumpliendo x_1=a y x_{n+1}=b (n lo calcula la propia
% función, devolviendo su valor)
%
% SALIDAS DE LA FUNCIÓN:
%
% n = Número de subintervalos contenidos en [a,b] a partir de x
% h = Vector columna con n filas que almacena el paso correspondiente a
% cada uno de los n subintervalos contenidos en [a,b]
%-----
format long
% Cálculo del número de subintervalos contenidos en [a,b]:
n = size(x,1)-1;
% Declaración e inicialización del vector columna de pasos h:
h = zeros(n,1);
% Construcción del paso variable hk y almacenado en el vector h:
for i = 1:n
    h(i) = x(i+1)-x(i);
end
end
```

- **Identificador de un determinado elemento de un vector partiendo del valor almacenado.**

```
function [id] = id_elemento_vector(x,valor)
%-----
% La función "id_elemento_vector" identifica el elemento de un vector en el
% que se almacena un determinado valor indicado por el usuario como
% entrada. Esto permite encontrar rápidamente y sin ningún esfuerzo
% adicional nodos comunes en mallas densificadas distintas.
%
% ENTRADAS DE LA FUNCIÓN:
%
% x = Vector en el que se desea identificar un determinado valor
```



```
% valor = Valor que se desea identificar en el vector x
%
% SALIDAS DE LA FUNCIÓN:
%
% id = Posición del elemento en el que se almacena el valor pedido por
% el usuario en el vector x. Si el vector no contiene ningún valor que
% cumpla las especificaciones, la función falla
%-----
% Tamaño del vector x:
n = length(x);
% Búsqueda del elemento pedido:
for i = 1:n
    if x(i) == valor
        id = i;
    end
end
end
end
```

- Cálculo de los parámetros (coeficientes) comunes a todas las estrategias de interpolación que emplean pseudo-splines cuárticos (sección 5.2 de la memoria).

```
function [A_com] = pspline4int_coefs_com(A_prev,h)
%-----
% La función "pspline4int_coefs_com" obtiene los coeficientes comunes a
% todas las estrategias de interpolación de las soluciones discretas de PVI de
% 2º orden genérico (2.6), lineal (2.7) e independiente de y' (2.8) con
% pseudo-splines cuárticos sistematizables y expuestas en la sección 5.2 para
% los intervalos genéricos k=2,3,...,n. Para minimizar el error de redondeo,
% las evaluaciones de los coeficientes se llevan a cabo mediante el algoritmo
% de Horner, el cual permite evaluar polinomios de grado n mediante n sumas y
% n productos estrictamente monomiales.
%
% ENTRADAS DE LA FUNCIÓN:
%
% A_prev = vector fila que contiene los coeficientes a(k-1)i, con
% i=0,1,...,4 correspondiente a la fila (k-1)-ésima de la matriz que
% almacena los coeficientes
% del pseudo-spline cuártico interpolador en el dominio [a,b].
% h = Paso correspondiente al intervalo inmediatamente anterior k
%
% SALIDAS DE LA FUNCIÓN:
%
% A_com = Vector fila que contiene los coeficientes ak0, ak1 y ak2
% calculados comúnmente a todas la estrategias y posteriormente
% almacenados convenientemente en la matriz de coeficientes habilitada
% mediante concatenación.
%-----
format long
% Declaración e inicialización del vector fila A_com:
A_com = zeros(1,3);
% Cálculo de los coeficientes ak0, ak1 y ak2:
A_com(1) = A_prev(1)+h*(A_prev(2)+h*(A_prev(3)+h*(A_prev(4)+h*A_prev(5)))));
%ak0
```




```
A_com(2) = A_prev(2)+h*(2*A_prev(3)+h*(3*A_prev(4)+4*h*A_prev(5))); %ak1
A_com(3) = A_prev(3)+h*(3*A_prev(4)+6*h*A_prev(5)); %ak2
end
```