



Métodos numéricos para ecuaciones diferenciales rígidas.
Aplicación a la semidiscretización del Método de Elementos Finitos.

Elisabete Alberdi Celaya

dirigida por:

Dr. Juan José Anza
Departamento de Matemática Aplicada UPV/EHU

*Para la obtención del grado de Doctor
bajo el programa de doctorado de la Universidad del País Vasco:
Ingeniería Mecánica de Sistemas Mecánicos y Estructurales.*

*Nire ingurune hurbilekoei,
bereziki hurbilenekoei:
zuri aita, zuri ama,
zuei Alaz eta Imanol, zuri Joseba,
zuei, nire bihotz eta memoria zaudenten
nire bi amamei
eta zuri Aitor.*

Eskertza

Hasitako lanak bukatzea ohitura ona izaten da, batzuetan kostatzen bada ere. Behar beste denbora izan ez dugulako, bide gogor eta luzeek ibiltaria nekatu eta frustatzeko gaitasun handia dutelako edo biderako *bota onik* eduki ez dugulako, batzuetan ez da izaten posible hasitako hori bukatzea, ordea.

Tesia egitea nire betiko ametsa izan da eta azken lau urteotan, buru-belarri eta bihotz, batez ere honexetan aritu naiz. Umetatik liluratu izan naute Matematikek eta epealdi honetan bergogorazi didate zeinen zorrotzak eta batera ederrak diren. Berriz ohartu naiz, beste gauza batzuetan bezalaxe, matematiketan ere, lehenengoan ezen gauza gutxi ateratzen direla eta beraiek ulertzeko eta ikasteko egin beharreko esfortzua ez dela eskutakada baten sartzen den horietakoa. Baina gauza bat ulertzea edo hobetzea lortutakoan sentitzen duzun barne-sariak -hau ere eskutakadaz gaindi-, egindako esfortzua aise gainditzen duela ere frogatu didate.

8000korik uste dut sekula ez dudala igoko nahiz eta esperantzarik galdu behar ez den baina, niretzat, pausoz-pauso, artikuluz-artikulu, ekaitz, denbora gaizto eta neke uneetan eguraldi eta moralaren araberrako entrenamenduak eginez eta beti ere helmuga amestuz, tontorra egin dudalakoa daukat. Ametsa egi bihurtu dut, *bota onek* zerikusia izan dute nonbait.

Lea Artibai ikastetxeko hamahiru urteko ibilbidearen ondoren, iaz aldi berria hasi nuen Euskal Herriko Unibertsitatean. Eskakizun berriak dituen aldia da berau: tesia bukatzea da bat eta unibertsitateko irakasle ziurtagiria lortzea beste bat. Beraz, lan erdiak bete baditut ere, beste erdiak ditut zain.

Tesi honek ez du olaturik harrotuko, ez dio munduari zirkinik edo ikararik eragingo, tesien unibertso zabalean izar txiki bat baino ez baita; nahiz eta niretzat berezia den, beharbada neuk sortutako izarra delako. Bereziak zareten bezalaxe bide honetan animatu eta lagundu didazuen guztiok. Nire eskerrik beroenak konszienteki edo inkonszienteki lagun egin didazuen *sherpoi*: Aitorri, bere mugarik gabeko pazientziagatik (infinitua existitzen denaren adibide eta frogak); etxekoei, asko geratzen zitzaidanean ere gutxi geratzen zela esateagatik, hau da, kalkulatuak *hain ondo* egiteagatik; txiki-txikitatik mantentzen ditudan lagunei eta nagusitu ahala egin ditudan berriei, zelan nindoan eta nengoen uneoro -hau da beti- arduratzeagatik; nire lankide-ohi eta lagun Arantza, Dolo, Jose Ramon eta Rikardori, besteak beste zuekin talde-lanean aritutako urteetan irakatsi dizkidazuen *teoremengatik*; *etxe-ondoko eskolari*, hau da, Lea Artibai ikastetxeari, eman dizkidan erraztasun guztiengatik eta ingelesez esaten den bezalaxe, *last but not least* nire tutore Juanjo Anzari, gidaririk gabe nekezago aurkituko bainuen gorako bidea. Eta izen gehiago aipatzea luze joango litzatekeenez -eta Matematikei erreferentzi eginez-, 2 eta 2 lau diren bezalaxe ($2 + 2 = 4$) bukatuko nuenaren sinismena berbaz eta animoz adierazi didazuen guztioi.

Mila esker bihotz-bihotzez

Agradecimientos

Es bueno terminar los trabajos que se han empezado, aunque a veces cuesta. El hecho de no disponer de suficiente tiempo, el cansancio y la frustración que genera cualquier proceso largo, o simplemente, no tener unas *buenas botas*, pueden dificultar o incluso impedir la consecución de los mismos.

Siempre he tenido el sueño de terminar la tesis y estos cuatro últimos años, me he dedicado en cuerpo y alma a ello. Desde pequeña me han sorprendido las Matemáticas y en este periodo me han vuelto a recordar el rigor que se necesita al trabajar con ellas y que su esencia se encuentra en su belleza. Me he dado cuenta de nuevo que, como en otras tantas disciplinas, tampoco en ésta salen las cosas a la primera; que la constancia, la tenacidad y el esfuerzo que se requieren para entenderlas, demostrarlas, etc. son enormes. Pero que la satisfacción que un@ siente en la meta es directamente proporcional al esfuerzo requerido.

Creo que en mi vida nunca subiré ningún 8000 -aunque no haya que perder la esperanza-. Pero para mí, paso a paso, artículo tras artículo, realizando entrenamientos acordes al tiempo y a la moral y siempre sin perder de vista la meta, me parece que he hecho cumbre en un 8000. He cumplido mi sueño -ese sueño que de momento sólo era real en mi dimensión imaginativa- y parece que las *buenas botas* han tenido algo que ver.

Tras dar fin a un recorrido de trece años en Lea Artibai ikastetxea, el año pasado empecé una nueva etapa en la Universidad del País Vasco. Es una etapa con nuevas exigencias: terminar la tesis es una de ellas y conseguir la acreditación de profesor universitario es otra. Por lo que, todavía tengo trabajo por hacer.

Esta tesis no hará que las olas azoten más fuertemente, o que notemos que la tierra gira “y sin embargo se mueve...”, ya que tan sólo es una estrella más en el universo infinito de las tesis; aunque para mí es especial, quizá únicamente porque es mi estrella. Como sois especiales tod@s l@s que me habéis animado y apoyado durante este proceso. Mi más sincero agradecimiento a los *sherpas* que consciente o inconscientemente me habéis ayudado: a Aitor, por su ilimitada paciencia (prueba de que el infinito existe); a mi familia, por decirme que ya quedaba poco, incluso cuando faltaba mucho, es decir, *por lo bien que hacéis los cálculos*; a mis amig@s, a aquéll@s que mantengo desde mi niñez y a l@s que he ido haciendo según crecía, por vuestro constante interés de cómo iba la tesis y cómo me encontraba yo; a mis excompañeros y amigos Arantza, Dolo, Jose Ramon y Rikardo, entre otras cosas por los *teoremas* que me habéis enseñado en el tiempo que hemos trabajado juntos; a esa escuela que está *junto a nuestra casa*, es decir, a Lea Artibai ikastetxea, por todas las facilidades que me ha dado y como se dice en inglés, *last but not least* a mi tutor Juanjo Anza, porque sin guía hubiese sido más difícil encontrar el camino hacia la cima. Y haciendo referencia a las Matemáticas, a todos los que me habéis animado y dicho que lo iba a terminar como 2 y 2 son cuatro ($2 + 2 = 4$).

Gracias de todo corazón

Índice general

Índice general	IX
1. Introducción, objetivos y organización	1
1.1. Introducción	1
1.2. Algunos ejemplos MEF en Matlab	10
1.2.1. Ejemplo 1: Solución exacta y solución MEF modal	10
1.2.1.1. Solución del continuo. Desarrollo en serie, D'Alembert	11
1.2.1.2. Semidiscretización MEF. Superposición modal	14
1.2.2. Ejemplo 2: Solución MEF numérica	21
1.2.2.1. Difusión	21
1.2.2.2. Onda	25
1.2.3. Ejemplo 3: Una versión no lineal de la ecuación de onda	34
1.3. Objetivos y organización del trabajo	39
2. Métodos para EDOs de orden 1	45
2.1. Error, estabilidad y rigidez	46
2.1.1. Error y orden de precisión	46
2.1.2. Estabilidad	47
2.1.3. Rigidez	50
2.2. Métodos multipaso lineales	51
2.2.1. Interpolación polinómica. Diferencias regresivas	53
2.2.2. Métodos de Adams Bashforth	54
2.2.3. Métodos de Adams Moulton	55
2.2.4. Métodos BDF	57
2.2.5. Métodos NDF	60
2.2.6. Métodos multipaso lineales mejorados	61
2.2.6.1. Métodos multipaso lineales que utilizan la 2ª derivada	61
2.2.6.2. Métodos multipaso extendidos	63
2.2.6.3. Combinación de métodos multipaso	64
2.3. Métodos Runge-Kutta	66
2.3.1. Introducción al método	67
2.3.2. Condiciones de orden	68

2.3.3.	Características de estabilidad	71
2.3.4.	Métodos explícitos anidados	73
3.	Métodos lineales multipaso para EDOs de orden 2	77
3.1.	Métodos lineales multipaso para EDOs de orden 2	78
3.2.	Método alfa-generalizado	80
3.2.1.	Método de Newmark	88
3.2.2.	Método alfa	92
3.3.	Método de Houbolt	93
3.4.	Método Collocation	94
3.5.	Métodos que disipan la energía y conservan el momento	95
3.6.	Otros parámetros	96
4.	ode45, ode15s	99
4.1.	ode45	99
4.1.1.	Estimación del error en la ode45	100
4.1.2.	Estimación del error local vs. regiones de estabilidad ode45	103
4.1.3.	Control del tamaño de paso	106
4.1.3.1.	Primer tamaño de paso	107
4.1.3.2.	Nuevo tamaño de paso tras h bueno	107
4.1.3.3.	Nuevo tamaño de paso tras h fallido	108
4.2.	ode15s	109
4.2.1.	Aspectos computacionales ode15s	109
4.2.1.1.	La matriz de diferencias	112
4.2.2.	Estimación del error en la ode15s	113
4.2.3.	Estimador del error local vs. regiones de estabilidad ode15s	115
4.2.4.	Control del tamaño de paso	115
4.2.4.1.	Primer tamaño de paso	115
4.2.4.2.	Nuevo tamaño de paso tras h bueno	115
4.2.4.3.	Nuevo tamaño de paso tras h fallido	117
4.2.4.4.	Actualización de la matriz de diferencias	118
4.3.	Aplicación de ode45 y ode15s a ejemplos sencillos	120
4.3.1.	Resolución de EDOs de orden 1	120
4.3.2.	Resolución de EDOs de orden 2	125
4.3.3.	Estimación del error en la ode15s	132
4.3.4.	Resolución de la ecuación de onda	135
4.3.5.	Conclusiones	138
5.	Métodos resultantes de cambiar los predictores en EBDF y MEBDF	139
5.1.	Introducción	139
5.2.	Características de los métodos BDF, NDF, EBDF, MEBDF	140
5.2.1.	Esquema de los métodos BDF y NDF	141
5.2.2.	Esquema de los métodos EBDF y MEBDF	141
5.3.	Esquema de los nuevos métodos	143

5.3.1.	ENDF y MENDF	143
5.3.2.	ENBDF y MENBDF, EBNDf y MEBNDf	144
5.4.	Análisis de estabilidad	145
5.4.1.	Estabilidad de los métodos ENDF y MENDF	145
5.4.2.	Estabilidad de los métodos ENBDF y MENBDF	148
5.4.3.	Estabilidad de los métodos EBNDf y MEBNDf	149
5.4.4.	Conclusiones sobre las regiones de estabilidad	150
5.5.	Resultados numéricos	151
6.	BDF-α: mét. multipaso con control del amortiguamiento algorítmico	155
6.1.	Radios espectrales de los métodos BDF	155
6.2.	Método BDF- α	160
6.2.1.	Orden	161
6.2.2.	Regiones de estabilidad	163
6.2.3.	Factor de amplificación	164
6.2.4.	Error relativo en el periodo y amortiguamiento algorítmico	165
7.	Metodología y resultados	169
7.1.	Problema lineal. El objeto malla	170
7.1.1.	Resumen de la formulación	170
7.1.1.1.	Laplace-Poisson	171
7.1.1.2.	Formato matricial (2D)	172
7.1.1.3.	Elasticidad lineal	173
7.1.1.4.	Postproceso. Alisado de flujos y tensiones	173
7.1.2.	Objetos y resolución	174
7.1.2.1.	Datos del problema y resolución	174
7.1.2.2.	Objeto método elementos finitos	175
7.1.2.3.	Objeto sistema lineal de ecuaciones	175
7.1.2.4.	Objeto problema	176
7.1.2.5.	Objeto Restricciones y Condiciones de Contorno	176
7.1.2.6.	Objeto malla	176
7.1.2.7.	Objeto elemento finito	177
7.1.2.8.	Objeto elemento patrón	177
7.1.2.9.	Objeto punto de integración	178
7.1.2.10.	Objeto material	179
7.1.3.	Postproceso	180
7.1.4.	Conclusiones	180
7.2.	Problema no lineal	181
7.2.1.	Resumen de la formulación	181
7.2.1.1.	Elasticidad	181
7.2.1.2.	Formato matricial 2D	182
7.2.1.3.	Laplace-Poisson	184
7.2.2.	Objetos y resolución	185
7.2.2.1.	Objeto malla	185

7.2.2.2.	Objeto elemento finito	186
7.2.2.3.	Objeto elemento patrón	186
7.2.2.4.	Objeto punto de integración	186
7.2.2.5.	Objeto deformación	187
7.2.2.6.	Objeto material	188
7.2.2.7.	Objeto problema	189
7.2.2.8.	Objeto método Newton Raphson	189
7.2.2.9.	Objeto Jacobiano	190
7.2.2.10.	Objeto método elemento finito	190
7.2.3.	Conclusiones	191
7.3.	El problema dinámico	192
7.3.1.	Resumen de la formulación	192
7.3.2.	Objetos	194
7.3.2.1.	Objeto método ode (1), objeto base_met	195
7.3.2.2.	Objeto fty	196
7.3.2.3.	Objeto método ode (2)	197
7.3.2.4.	Objeto problema	198
7.3.3.	Método cuasi-estático	198
7.3.3.1.	Objeto cuasi	199
7.3.4.	Método trapezoidal	199
7.3.4.1.	Objeto trap	200
7.3.4.2.	Objeto traplinc	201
7.3.5.	Métodos multipaso	202
7.3.5.1.	Método BDF	203
7.3.5.2.	Método NDF	206
7.3.5.3.	Método BDF- α	209
7.3.5.4.	Método EBDF	211
7.3.5.5.	Método MEBDF	213
7.3.5.6.	El resto de métodos de las familias EBDF y MEBDF	215
7.3.6.	Método odesuite	216
7.3.6.1.	Objeto odesuite	217
7.3.7.	Método alfa	218
7.3.7.1.	Objeto alfa	221
7.3.7.2.	Objeto alfalinc	222
7.3.8.	Conclusiones	222
7.4.	Resultados	223
7.4.1.	Ejemplo 1: Ecuación de onda lineal	223
7.4.2.	Ejemplo 2: Una versión no lineal de la ecuación de onda	230
7.4.3.	Ejemplo 3: Elasticidad dinámica lineal	234
7.4.3.1.	Solución modal y paso a paso	235
7.4.3.2.	Atenuación, amortiguamiento modal	237
7.4.4.	Ejemplo 4: Tracción elastoplástica de bloque axisimétrico	241
7.4.4.1.	Caso estático	242

7.4.4.2. Caso dinámico	243
7.4.5. Ejemplo 5: Flexión de viga hiperelástica incompresible	244
7.4.5.1. Caso estático en tensión plana	245
7.4.5.2. Caso dinámico en tensión plana	245
7.4.5.3. Caso dinámico en deformación plana	247
7.4.6. Ejemplo 6: Movimiento de rotación de muelle no lineal	247
8. Conclusiones y futuras líneas de investigación	253
8.1. Conclusiones y aportaciones de la tesis	254
8.2. Futuras líneas de investigación	257
A. Ecuación de onda estándar	259
B. Autovalores de las EDP lineales de difusión y onda	261
B.1. Difusión	261
B.1.1. Rigidez en función del n° de elementos de la semidiscretización	261
B.1.2. Rigidez en función de la longitud de la varilla L	263
B.1.3. Rigidez en función de la difusividad térmica	263
B.2. Onda	265
B.2.1. Relación entre los autovalores del problema de difusión y onda	266
B.2.2. Rigidez en función del n° de elementos de la semidiscretización	268
B.2.3. Rigidez en función de la longitud de la cuerda L	269
B.2.4. Rigidez en función de la velocidad de propagación	269
C. Datos de las funciones de los objetos	271
D. Programación Objeto	275
D.1. Objeto métodos multipaso no lineales	275
D.1.1. Objeto bdf	276
D.1.2. Objeto ndf	277
D.2. Objeto métodos multipaso lineales	278
D.2.1. Objeto bdfinc	279
D.2.2. Objeto ndfinc	280
E. Publicaciones y participación en Congresos	281
E.1. Publicaciones científicas	281
E.2. Participación en Congresos	281
E.3. Publicaciones de divulgación	282
Índice de figuras	283
Índice de tablas	289
Bibliografía	291

Capítulo 1

Introducción, objetivos y organización

Cuando creáis haber encontrado un hecho científico importante y os apremie el deseo de publicarlo, esperad unos días o unas semanas, o años; es preciso luchar, comprobar e incluso destruir los experimentos propios, es preciso agotar todas las hipótesis contrarias antes de proclamar el descubrimiento. Pero luego, al cabo de esfuerzos tan arduos, cuando la certeza llega, vuestra alegría será una de las más grandes que puede experimentar el alma humana.

LOUIS PASTEUR

1.1. Introducción

La matemática aplicada a la ingeniería tiene una importante incidencia en el análisis y el diseño de procesos y productos. Numerosos fenómenos de la ciencia y de la ingeniería quedan modelados matemáticamente mediante sistemas de Ecuaciones en Derivadas Parciales (EDPs). Así, los balances de masa, cantidad de movimiento y energía, con leyes constitutivas apropiadas, son la base de una amplia clase de problemas de Condiciones de Contorno (CC) que modelan el movimiento macroscópico de sólidos, fluidos y gases con sus fuerzas correspondientes, o el transporte de calor y de especies materiales, o los problemas de interacción entre distintos medios: mecánicos, químicos, electromagnéticos, térmicos, etc.

En general, la resolución analítica de las EDPs en dominios de forma cualquiera es imposible y es preciso recurrir a métodos numéricos, entre los que sobresalen las diferencias finitas (MDF), los volúmenes finitos y el método de los elementos finitos (MEF), siendo este último el más sencillo y capaz para tratar dominios de forma cualquiera. El método de los elementos finitos (MEF) ha transformado el modo de trabajo del ingeniero de diseño en los últimos 50 años. De los tableros de dibujo, los manuales de

fórmulas de la resistencia de materiales o de la mecánica de fluidos, la gran experiencia práctica de los proyectistas,... se ha pasado a la utilización de potentes herramientas computacionales CAD-CAE que reducen drásticamente el ciclo de diseño y el número de prototipos experimentales a fabricar y ensayar.

La causa principal de esta transformación reside en el incremento exponencial de la potencia computacional junto con la disminución espectacular de su coste. Ello ha impulsado el trabajo de numerosos investigadores en este campo y la creación de importantes compañías de software que desarrollan paquetes comerciales. Nastran, ANSYS, Abaqus, LS-DYNA, COMSOL, etc. son frutos del conocimiento desarrollado y de la investigación publicada.

Se puede decir que el análisis lineal está consolidado como herramienta de diseño mecánico y estructural, donde lo habitual es que las especificaciones de diseño no admitan deformaciones anelásticas y requieran suficiente rigidez como para evitar la formulación en grandes deformaciones. La interacción con el CAD y el mallado pueden ser pasos críticos antes de obtener los valores nodales (desplazamientos en el análisis de tensiones, concentraciones en los sistemas químicos, potencial en los problemas electromagnéticos, etc.) mediante la resolución de grandes sistemas de ecuaciones lineales, desde mil hasta un millón -o más- de ecuaciones.

No ocurre así con los problemas no lineales (CFD, impacto, conformado), ni con los de interacción multifísica, donde la disponibilidad de *soft* científico robusto y eficiente es todavía un factor de limitación. En este contexto es relevante la discusión de cómo conceptos modernos de las ciencias computacionales son útiles para el desarrollo de *soft* científico más fácil de extender, mantener y utilizar que los conceptos tradicionales.

Tradicionalmente, el desarrollo de software numérico se ha basado en el uso de lenguajes procedimentales como Fortran o C, pero en los últimos años ha habido un interés creciente en aplicar los paradigmas de la Programación Orientada a Objeto (POO), que permite reutilizar los códigos, extenderlos y mantenerlos de forma más eficiente que la secuencial [53].

En la programación procedimental (secuencial), las dependencias pueden ser difíciles de detectar y pequeños cambios en las estructuras de datos pueden desencadenar efectos impredecibles. Por contra, en la POO los métodos están asociados a los datos que gestionan mediante la definición de clases abstractas que se concretan en objetos. Entonces, el polimorfismo (o sobrecarga) permite que la misma función pueda responder de distinta manera según el tipo de objeto sobre el que actúe. La encapsulación acrecienta la seguridad y robustez del código, ya que los datos del objeto sólo son accesibles desde sus métodos, y la herencia permite organizar las clases en jerarquías, facilitando la reutilización del código sin necesidad de alterarlo, ver [3].

La modelización matemática del medio continuo mediante ecuaciones y sistemas de Ecuaciones Diferenciales aparece en principio organizado en numerosas disciplinas cuasi-independientes: elasticidad y resistencia de materiales, mecánica de fluidos, electromag-

netismo, transferencia de calor, reactores químicos, etc. Es un planteamiento natural desde otro punto de vista, pero oculta la estrecha relación existente entre aplicaciones con similares fundamentos físicos y matemáticos que conducen a similares problemas de valores de contorno. Por ejemplo, la EDP generalizada de Laplace:

$$\nabla \cdot \left[\underbrace{-C\nabla u}_{\text{Flujo conductivo}} + \underbrace{\mathbf{D}u}_{\text{Flujo convectivo}} \right] + \underbrace{Eu}_{\text{Absorción}} = \underbrace{f}_{\text{Fuente}} \quad (1.1)$$

representa el comportamiento de varios problemas estacionarios de interés.

Si la función incógnita $u(x, y, z)$ representa la temperatura, la constante física C la conductividad térmica y el vector \mathbf{D} es $\mathbf{D} = \rho \mathbf{v} c_p$, siendo \mathbf{v} la velocidad del fluido, ρ la densidad y c_p el calor específico, entonces la ecuación (1.1) gobierna el transporte de calor y representa en cada punto del medio continuo el balance de energía a partir de la ecuación constitutiva de Fourier que relaciona el flujo conductivo con el gradiente de temperaturas. Pero si la función incógnita $u(x, y, z)$ representa la concentración de una especie, la constante física C la difusividad y el vector $\mathbf{D} = \mathbf{v}$ siendo \mathbf{v} la velocidad del fluido, entonces (1.1) gobierna el transporte másico de una especie (soluto) en un medio (disolvente) y representa el balance de masa en cada punto a partir de la ecuación constitutiva de Fick, que relaciona el flujo difusivo con el gradiente de la concentración.

Cualquiera de los dos casos corresponde a un modelo convectivo difusivo advectivo lineal. Los coeficientes pueden ser constantes o variables en cada punto $\mathbf{x} = (x, y, z)$, pero si alguno de ellos depende de la función incógnita $u(\mathbf{x})$ el problema será no lineal. Por ejemplo, si en el segundo caso el término fuente depende de u , el problema se denomina reactivo y gobierna los procesos de los reactores químicos, donde pueden existir n especies y acoplamiento con la temperatura, resultando en un sistema de $(n + 1)$ EDPs del tipo (1.1).

Si consideramos nulos el término convectivo y el término de absorción ($\mathbf{D} = 0, E = 0$) obtenemos la EDP de Laplace-Poisson:

$$\nabla \cdot (-C\nabla u) = f \quad (1.2)$$

que gobierna el problema del potencial, y tiene aplicaciones variadas como el flujo de un fluido incompresible y no viscoso, el campo electrostático, la torsión de un perfil de sección cualquiera, la conducción simple de calor, la difusión simple de masa sin convección, etc.

Todos estos casos son estacionarios, pero basta con añadir un término fuente adicional relacionado con la inercia al cambio temporal para extender cualquiera de los modelos anteriores al caso más general transitorio, que queda gobernado por EDPs con Condiciones Iniciales (CI) y Condiciones de Contorno (CC). Los casos representativos más simples son la ecuación de difusión y la ecuación de onda, que se obtienen directamente de (1.2) para $f = 0$.

- Ecuación de difusión: Cogiendo $f = -G \cdot u_t(\mathbf{x}, t)$, la ecuación (1.2) queda:

$$\nabla \cdot C \nabla u(\mathbf{x}, t) = G \cdot u_t(\mathbf{x}, t) \rightarrow \Delta u = \frac{G}{C} u_t \quad (1.3)$$

- Ecuación de onda: Cogiendo $f = -G \cdot u_{tt}(\mathbf{x}, t)$, la ecuación (1.2) queda:

$$\nabla \cdot C \nabla u(\mathbf{x}, t) = G \cdot u_{tt}(\mathbf{x}, t) \rightarrow \Delta u = \frac{G}{C} u_{tt} \quad (1.4)$$

En el primer caso, si la función incógnita u representa la temperatura, la constante física C la conductividad térmica y $G = \rho c_p$, la EDP es la ecuación de calor que queda definida con una única constante si C y G son valores constantes en todo el dominio.

En el segundo caso, si la función incógnita $u(\mathbf{x})$ representa el desplazamiento transversal de una cuerda tensionada de densidad ρ y traccionada con una fuerza de tiro T , siendo $C = T$ y $G = \rho$, la EDP gobierna las vibraciones transversales de la cuerda. Si los parámetros físicos son constantes en ella, $c = \sqrt{T/\rho}$ es la velocidad de propagación de la onda.

También en el segundo caso, si consideramos un medio elástico con coeficientes de Lamé λ y μ , siendo \mathbf{u} el vector de desplazamientos que deforma sin distorsión y $C = \lambda + 2\mu$, $G = \rho$, la EDP gobierna las ondas elásticas dilatacionales del medio. La función incógnita $u(\mathbf{x})$ representa la dilatación volumétrica (que es un escalar): $\nabla \cdot \mathbf{u}$.

Sin embargo, si el medio elástico deforma sin dilatación volumétrica y si $C = \mu$ y $G = \rho$, la EDP (1.4) es vectorial y gobierna las ondas equivolumétricas del medio, siendo la incógnita una función vectorial $\mathbf{w}(\mathbf{x})$ que representa la rotación elástica:

$$\mathbf{w}(\mathbf{x}) = \frac{1}{2} (\nabla \times \mathbf{u}) \quad (1.5)$$

El caso general de deformación elástica dinámica con fuerzas de volumen \mathbf{f} queda modelado por las ecuaciones dinámicas de Navier [34]:

$$(\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} + \mathbf{f} = \rho \mathbf{u}_{tt} \quad (1.6)$$

El sistema de 3 EDPs (1.6) representa en cada punto del medio elástico el balance de la cantidad de movimiento, a partir de las ecuaciones constitutivas de Lamé:

$$\sigma = \lambda \nabla \cdot \mathbf{u} + 2\mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \quad (1.7)$$

Teniendo en cuenta (1.7), (1.6) se transforma en el sistema de ecuaciones de equilibrio dinámico:

$$\nabla \cdot \sigma + \mathbf{f} = \rho \mathbf{u}_{tt} \quad (1.8)$$

que es el punto de partida para trabajar con ecuaciones constitutivas de tipo no lineal, como plasticidad e hiperelasticidad, y con grandes deformaciones.

En (1.8) $\rho \mathbf{u}_{tt}(\mathbf{X}, t)$ representa la fuerza de inercia por unidad de masa que afecta a la partícula que ocupa la posición \mathbf{X} en una configuración inicial de referencia. Es el planteamiento lagrangiano propio del estudio de los sólidos con grandes deformaciones. Sin embargo en los fluidos no existe configuración de referencia, el planteamiento es euleriano y estamos interesados en conocer en cada punto \mathbf{x} del flujo, el campo de velocidades, $\mathbf{u}(\mathbf{x}, t)$ y la presión $p(\mathbf{x}, t)$. Entonces la aceleración de la partícula que ocupa la posición \mathbf{x} en un instante t será:

$$\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial t} = \frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \nabla \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t) \quad (1.9)$$

Apareciendo un nuevo término convectivo no lineal.

Un paso más nos conduce desde las ecuaciones de Navier (1.8) a las ecuaciones de Navier-Stokes que gobiernan el comportamiento de fluidos incompresibles newtonianos. Basta para ello considerar que el vector \mathbf{u} representa el campo de velocidades, incluir en (1.6) el término convectivo y considerar como ecuación constitutiva:

$$\sigma = -p\mathbf{I} + \eta \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \quad (1.10)$$

donde η es el coeficiente de viscosidad de Newton y p la presión. Entonces se obtiene el sistema no lineal de tres EDPs [35]:

$$\begin{aligned} \rho \mathbf{u}_t + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} &= \nabla \cdot \sigma + \mathbf{f} \Rightarrow \\ \rho \mathbf{u}_t + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} &= \nabla \cdot \left[-p\mathbf{I} + \eta \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] + \mathbf{f} \end{aligned}$$

que debe de ser complementada con la condición de incompresibilidad: $\nabla \cdot \mathbf{u} = 0$, para determinar las cuatro funciones incógnitas (\mathbf{u} y p). En el caso más general de compresibilidad será preciso otra ecuación constitutiva para la presión.

La secuencia anterior muestra las similitudes entre las EDPs y su progresión constructiva, ya que comportamientos más complejos quedan modelados con la adición de términos sobre las EDPs que gobiernan comportamientos más simples, ver [39].

Este hecho tiene importantes consecuencias prácticas, ya que facilita el desarrollo de bloques constructivos cuya reutilización reduce el esfuerzo requerido para desarrollar software en un área particular de aplicación. Así, los ingredientes básicos de la aproximación MEF quedan ya reflejados en el problema más simple: la EDO lineal de Laplace-Poisson y adiciones posteriores añadirán complejidad hasta llegar a la aproximación MEF del sistema de EDPs no lineal de Navier-Stokes.

En este sentido, la aproximación MEF de un problema estacionario de conducción de calor de conductividad k constante en el dominio Ω y con CC Dirichlet homogéneas

viene dada por:

$$\left\{ \begin{array}{l} \boxed{-\nabla \cdot [k(\mathbf{x}) \nabla u(\mathbf{x})] = f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega} \\ \text{CC. Dirichlet : } u(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \Gamma_1 \\ \text{CC. Neumann : } k(\mathbf{x}) \frac{\partial u(\mathbf{x})}{\partial n} = h(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_2 \end{array} \right. \quad (1.11)$$

El proceso para hallar $u(\mathbf{x})$ consiste en discretizar el dominio en elementos y nodos, definir las funciones MEF de pequeño soporte $N_j(\mathbf{x})$ asociadas a los nodos de índice j , tales que verifican $N_i(\mathbf{x}_j) = \delta_{ij}$, y definir la aproximación:

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{j \in \eta_d} d_j N_j(\mathbf{x}) \quad (1.12)$$

donde η_d es el conjunto de los índices de los nodos que no pertenecen al contorno Dirichlet Γ_1 . De esta forma los coeficientes de la aproximación verifican:

$$d_j = u^h(\mathbf{x}_j) \approx u(\mathbf{x}_j)$$

La función $u^h(\mathbf{x})$ verifica por construcción las CC Dirichlet homogéneas. Para aproximar la EDP y la CC de contorno Neumann anulamos el residuo ponderado:

$$\int_{\Omega} N_i \left(-\nabla \cdot [k \nabla u^h] - f \right) d\Omega + \int_{\Gamma_2} N_i \left(k \frac{\partial u}{\partial n} - h \right) d\Gamma_2 = 0, \quad \forall i \in \eta_d \quad (1.13)$$

que obtenemos al insertar la aproximación (1.12) en el problema de CC (1.11), utilizando como funciones de peso las funciones de base $\{N_i(\mathbf{x}), i \in \eta_d\}$ de la aproximación.

Integrando por partes el término con la derivada segunda, se anula un término de la integral sobre el contorno Γ_2 , y se reduce el orden de derivación de la función incógnita $u(\mathbf{x})$. Por tanto, también se reduce la exigencia sobre las funciones de base MEF de la aproximación, cuyas primeras derivadas pueden ser continuas a trozos:

$$\int_{\Omega} \left(k \nabla N_i \cdot \nabla u^h \right) d\Omega = \int_{\Omega} N_i f d\Omega + \int_{\Gamma_2} N_i h d\Gamma_2, \quad \forall i \in \eta_d \quad (1.14)$$

Sustituyendo ahora la aproximación (1.12) en la expresión (1.14) obtenemos el sistema lineal de ecuaciones cuya resolución proporciona los coeficientes incógnita $d_j \approx u(\mathbf{x}_j)$ de la aproximación:

$$\sum_{j \in \eta_d} \left(\int_{\Omega} k \nabla N_i \cdot \nabla N_j d\Omega \right) d_j = \int_{\Omega} N_i f d\Omega + \int_{\Gamma_2} N_i h d\Gamma_2, \quad \forall i \in \eta_d \quad (1.15)$$

La expresión (1.15) se puede expresar de esta forma en formato matricial:

$$\sum_{j \in \eta_d} k_{ij} d_j = F_i, \quad \forall i \in \eta_d \Rightarrow \mathbf{K} \mathbf{d} = \mathbf{F}$$

Estos son los pasos básicos, ya que añadir complejidad a la EDP, por ejemplo los términos convectivo y de absorción de (1.1) sólo repercute en la matriz de coeficientes del sistema, que queda:

$$k_{ij} = \int_{\Omega} [k \nabla N_i \cdot \nabla N_j + \rho c_p N_i \mathbf{v} \cdot \nabla N_j + a N_i N_j] d\Omega, \quad \forall i, j \in \eta_d$$

siendo a el coeficiente físico de absorción.

Si la conductividad k depende de la temperatura, la EDP será no lineal y la matriz de coeficientes K dependerá de la solución:

$$k_{ij}(\mathbf{d}) = \int_{\Omega} \left[k \left(\underbrace{\sum_{j \in \eta_d} d_j N_j(\mathbf{x})}_{u^h(\mathbf{x})} \right) \nabla N_i \cdot \nabla N_j + \rho c_p N_i \mathbf{v} \cdot \nabla N_j + a N_i N_j \right] d\Omega, \quad \forall i, j \in \eta_d$$

Y el sistema de ecuaciones $K(\mathbf{d})\mathbf{d} = \mathbf{F}$ es no lineal y se resuelve iterando por Newton Raphson, para lo cual es preciso calcular tanto el residuo $R = \mathbf{F} - K(\mathbf{d})\mathbf{d}$, como la matriz jacobiana del mismo $J = \frac{\partial R}{\partial \mathbf{d}}$.

Otra forma de añadir complejidad es pasando del caso estacionario al transitorio. Lo haremos para los casos más sencillos de difusión (1.3) y onda (1.4):

$$\left\{ \begin{array}{l} \boxed{\rho c_p \frac{\partial u(\mathbf{x}, t)}{\partial t} = \nabla \cdot [k(\mathbf{x}) \nabla u(\mathbf{x}, t)], \quad \forall \mathbf{x} \in \Omega, t \in [0, \infty)} \\ \text{CC. Dirichlet :} \quad u(\mathbf{x}, t) = 0, \quad \forall \mathbf{x} \in \Gamma_1, t \in [0, \infty) \\ \text{CC. Neumann :} \quad k(\mathbf{x}) \frac{\partial u(\mathbf{x}, t)}{\partial n} = h(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Gamma_2, t \in [0, \infty) \\ \text{Condiciones Iniciales :} \quad u(\mathbf{x}, 0) = g(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \end{array} \right.$$

En ambos casos, se define la aproximación MEF mediante coeficientes incógnita que son funciones de t :

$$u(\mathbf{x}, t) \approx u^h(\mathbf{x}, t) = \sum_{j \in \eta_d} d_j(t) N_j(\mathbf{x}) \quad (1.16)$$

En lugar de obtener un sistema algebraico tras la semidiscretización MEF, obtendremos un sistema de Ecuaciones Diferenciales Ordinarias (EDOs). Siguiendo los pasos anteriores y para el caso más sencillo cuando $h = 0$, obtenemos:

$$\int_{\Omega} N_i \rho c_p u_t^h d\Omega = - \int_{\Omega} [k \nabla N_i \cdot \nabla u^h] d\Omega, \quad \forall i \in \eta_d \quad (1.17)$$

Derivando la aproximación (1.16) con respecto al tiempo y al espacio:

$$\frac{\partial u^h(\mathbf{x}, t)}{\partial t} = \sum_{j \in \eta_d} d_j'(t) N_j(\mathbf{x}); \quad \nabla u^h(\mathbf{x}, t) = \sum_{j \in \eta_d} d_j(t) \nabla N_j(\mathbf{x}) \quad (1.18)$$

Y sustituyendo (1.18) en (1.17) se tiene:

$$\sum_{\forall j \in \eta_d} \left(\underbrace{\int_{\Omega} \rho c_p N_i N_j d\Omega}_{m_{ij}} \right) d'_j(t) = - \sum_{\forall j \in \eta_d} \left(\underbrace{\int_{\Omega} k \nabla N_i \cdot \nabla N_j d\Omega}_{k_{ij}} \right) d_j(t), \quad \forall i \in \eta_d \quad (1.19)$$

Escribimos (1.19) en formato matricial:

$$\sum_{\forall j \in \eta_d} m_{ij} d'_j(t) = - \sum_{\forall j \in \eta_d} k_{ij} d_j(t), \quad \forall i \in \eta_d \Rightarrow \boxed{M \mathbf{d}'(t) = -K \mathbf{d}(t)} \quad (1.20)$$

En (1.20) hemos obtenido un sistema de EDOs de orden 1 que deberá integrarse a partir del vector de Condiciones Iniciales (CI) cuyas componentes son: $\{d_i^0 = g(\mathbf{x}_i), \forall i \in \eta_d\}$.

En el caso de la ecuación de onda dada por:

$$\left\{ \begin{array}{l} \boxed{\rho \frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} = \nabla \cdot [T(\mathbf{x}) \nabla u(\mathbf{x}, t)], \quad \forall \mathbf{x} \in \Omega, t \in [0, \infty)} \\ \text{CC. Dirichlet :} \quad u(\mathbf{x}, t) = 0, \quad \forall \mathbf{x} \in \Gamma_1, t \in [0, \infty) \\ \text{CC. Neumann :} \quad T(\mathbf{x}) \frac{\partial u(\mathbf{x}, t)}{\partial n} = h(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Gamma_2, t \in [0, \infty) \\ \text{Condiciones Iniciales :} \quad \begin{cases} u(\mathbf{x}, 0) = g_1(\mathbf{x}) \\ u_t(\mathbf{x}, 0) = g_2(\mathbf{x}) \end{cases} \quad \forall \mathbf{x} \in \Omega \end{array} \right.$$

se llega a un sistema de EDOs de orden 2 ($M \mathbf{d}''(t) = -K \mathbf{d}(t)$), que deberá integrarse a partir de los dos vectores de Condiciones Iniciales cuyas componentes son: $\{d_i^0 = g_1(\mathbf{x}_i), \forall i \in \eta_d\}, \{(d_i^1)^0 = g_2(\mathbf{x}_i), \forall i \in \eta_d\}$.

Observamos por tanto, que la aplicación del MEF al problema de CI y CC, desemboca en la resolución de un sistema de EDOs que queda definido por las matrices de rigidez K y masa M , que se pueden obtener por ensamblaje de matrices elementales. Cualquiera de los elementos de las matrices involucradas responde a la forma:

$$\int_{\Omega} f(\mathbf{x}) \mathcal{L} N_i \cdot \mathcal{M} N_j d\Omega, \quad \forall i, j \in \eta_d \quad (1.21)$$

siendo $f(\mathbf{x})$ una función asociada a la definición de una constante física, \mathcal{L} y \mathcal{M} el operador identidad en la matriz M y el operador gradiente en la matriz K respectivamente en los casos de difusión y onda. En otros casos, como Navier en elasticidad o Navier-Stokes en fluidos, estos operadores son algo más complejos pero actuando también sobre las funciones MEF N_j .

Por tanto, disponiendo de procedimientos comunes para la evaluación numérica de integrales del tipo (1.21), podremos formular y construir los sistemas de ecuaciones

asociados a la aproximación MEF de muchos problemas de interés. Si además disponemos de procedimientos para resolver sistemas de ecuaciones lineales, podremos resolver los problemas estacionarios lineales. Si además disponemos de procedimientos iterativos para resolver sistemas de ecuaciones no lineales, podremos resolver los problemas estacionarios no lineales. Y si a esto le añadimos que disponemos de procedimientos para resolver sistemas de ecuaciones diferenciales ordinarias, también podremos resolver los problemas evolutivos temporales haciendo uso de los procedimientos anteriores paso a paso en el tiempo: resolviendo un sistema lineal en cada paso si la EDP es lineal o un sistema no lineal en caso contrario.

El punto inicial de este proceso constructivo, es el cálculo numérico de integrales elementales en elementos normalizados (patrón) donde se conocen las posiciones de muestreo del integrando y los pesos correspondientes. El integrando queda definido por las funciones MEF elementales utilizadas, el determinante jacobiano de la transformación geométrica entre el elemento y el elemento patrón, y las características de la EDP o sistema de EDPs tratado. Este esquema es el mismo para 1D, 2D, o 3D y una vez implementado puede ser reutilizado en cualquier problema lineal o no lineal, estacionario o transitorio, a expensas de suministrar la forma del integrando que depara la forma débil de la EDP y las funciones MEF a utilizar.

En este contexto la reutilización de código puede ser fundamental, ya que ahorrará esfuerzo de programación y mejorará la fiabilidad de los nuevos desarrollos que descansarán en componentes muy utilizados y comprobados. Convendrá además añadir mecanismos que faciliten las contribuciones de distintas personas, y su posterior disponibilidad. Con este fin la alternativa utilizada en este trabajo es Matlab, entorno matemático para el cálculo interactivo y la programación.

Matlab proporciona herramientas para resolver sistemas de ecuaciones, calcular integrales, manipular polinomios e integrar sistemas de ecuaciones diferenciales. Es fácilmente extensible y personalizable a través de funciones definidas por el usuario, escritas en lenguaje Matlab o utilizando módulos escritos en C++, Java, C o Fortran. Es fácil de usar e instalar y no precisa compilar ni encadenar con las librerías para montar los programas ejecutables. Es inmediato experimentar de manera interactiva y también es inmediato programar en ficheros de texto siguiendo unas reglas mínimas. No hace falta declarar ni dimensionar y las funciones de cálculo que proporciona están muy comprobadas y avaladas por millones de usuarios. La programación puede hacerse en la forma procedimental clásica o también siguiendo las metodologías de la Programación Orientada a Objeto (POO). Es un lenguaje de alto nivel que exime de los detalles, facilitando el diseño rápido, la prueba y la puesta a punto de los prototipos. Matlab es un intérprete y no compete en tiempo de ejecución con largos ejecutables de C o Fortran, pero no obstante es muy rápido para numerosas aplicaciones. No proporciona recursos directos para la resolución de las EDPs mediante el MEF, pero sí proporciona potentes recursos para resolver numéricamente sistemas de EDOs que se obtienen mediante semidiscretización MEF, por ejemplo de las ecuaciones de difusión y onda anteriormente descritas. También resultan muy útiles los recursos que proporciona para trabajar con los sistemas de

ecuaciones dispersos, característicos del método MEF, liberando de los algoritmos de reenumeración y de los sistemas especiales de almacenamiento de las matrices.

1.2. Algunos ejemplos MEF en Matlab

A continuación se presentan tres ejemplos sencillos unidimensionales que reflejan muchos de los aspectos mencionados y algunos otros que son motivación para este trabajo. Los tres tratan las EDPs de difusión y onda, en un dominio de longitud L y con Condiciones de Contorno Dirichlet homogéneas (nulas). Se considera también nula la 2ª Condición Inicial de la onda, la velocidad inicial. En los dos primeros ejemplos trabajaremos con la versión lineal:

$$\blacksquare \text{ Difusión: } \begin{cases} u_t = \alpha^2 u_{xx} \\ CC : u(0, t) = 0 = u(L, t) \\ CI : u(x, 0) = g(x) \end{cases} \quad (1.22)$$

siendo $\alpha^2 = \frac{k}{c\rho}$ la difusividad térmica, k la conductividad termal, c la capacidad termal y ρ la densidad.

$$\blacksquare \text{ Onda: } \begin{cases} u_{tt} = \alpha^2 u_{xx} \\ CC : u(0, t) = 0 = u(L, t) \\ CI : u(x, 0) = g(x), u_t(x, 0) = 0 \end{cases} \quad (1.23)$$

siendo $\alpha = \sqrt{\frac{T}{\rho}}$ la velocidad de propagación de la onda, T el tiro en la cuerda y ρ la densidad lineal de masa.

En el tercer ejemplo trabajaremos con una versión no lineal de onda. En los tres casos observaremos que con pocas líneas de código Matlab podemos resolver las EDPs mediante el método de las líneas con semidiscretización MEF [47, 72] y con distintos métodos de integración temporal: superposición modal e integración paso a paso. Observaremos también que el deseo de mejorar la precisión afinando el tamaño de malla, conduce a sistemas de EDOs cuya rigidez aumenta con el número de elementos (ver Apéndice B), lo cual en principio supone una dificultad importante para la integración temporal y reclama la utilización de métodos especializados para sistemas rígidos (stiff).

1.2.1. Ejemplo 1: Solución exacta y solución MEF modal

En este ejemplo se resuelven los problemas de difusión y onda mencionados, primero de manera exacta por el método de separación de variables [25] y después de manera aproximada mediante semidiscretización MEF, resolviendo el sistema de EDOs mediante superposición modal. Las soluciones se concretan para tres casos particulares de CI: senoidal, triangular y pulso rectangular.

1.2.1.1. Solución del continuo. Desarrollo en serie y solución de D'Alembert

Utilizando el método de separación de variables podemos representar la solución de los problemas de CC y CI definidos por (1.22) y (1.23), mediante desarrollo en serie de autofunciones $\{\sin(\frac{k\pi x}{L}), k = 1, 2, \dots\}$:

$$\blacksquare \text{ Difusión: } u(x, t) = \sum_{k=1}^{\infty} A_k \sin\left(\frac{k\pi x}{L}\right) e^{-(\alpha k\pi)^2 t} \quad (1.24)$$

$$\blacksquare \text{ Onda: } u(x, t) = \sum_{k=1}^{\infty} A_k \sin\left(\frac{k\pi x}{L}\right) \cos\left(\frac{\alpha k\pi t}{L}\right) \quad (1.25)$$

siendo en ambos casos: $A_k = \frac{2}{L} \int_0^L g(x) \sin\left(\frac{k\pi x}{L}\right) dx$

Si la condición inicial coincide con una de las autofunciones, es decir, $g(x) = \sin\left(\frac{i\pi x}{L}\right)$, entonces: $A_k = \begin{cases} 0, & \text{si } i \neq k \\ 1, & \text{si } i = k \end{cases}$. Y la solución se reduce a uno de los términos de la serie:

$$\blacksquare \text{ Difusión: } u(x, t) = \sin\left(\frac{i\pi x}{L}\right) e^{-(\alpha i\pi)^2 t}$$

$$\blacksquare \text{ Onda: } u(x, t) = \sin\left(\frac{i\pi x}{L}\right) \cos\left(\frac{\alpha i\pi t}{L}\right)$$

En ambos casos la CI mantiene su forma senoidal pero decayendo exponencialmente en el primero y vibrando armónicamente en el segundo. En el caso de la onda, los autovalores son las frecuencias naturales y las autofunciones son los modos naturales de vibración.

Para una cuerda de 8 unidades de longitud ($L = 8$), con $\alpha^2 = 1$, se definen a continuación los tres casos de condiciones iniciales que vamos a tratar:

$$\blacksquare \text{ Senoidal: } g(x) = \sin\left(\frac{\pi x}{8}\right), \quad \forall x \in [0, 8]$$

$$\blacksquare \text{ Triangular: } g(x) = \begin{cases} \frac{x}{6}, & \forall x \in [0, 6] \\ \frac{8-x}{2}, & \forall x \in [6, 8] \end{cases}$$

$$\blacksquare \text{ Pulso: } g(x) = \begin{cases} 0, & \forall x \in [0, 3] \\ 1, & \forall x \in [3, 5] \\ 0, & \forall x \in (5, 8) \end{cases}$$

Cuyas soluciones son:

- Senoidal:

$$\text{Difusión: } u(x, t) = \sin\left(\frac{\pi x}{8}\right) e^{-\pi^2 t} \quad \text{Onda: } u(x, t) = \sin\left(\frac{\pi x}{8}\right) \cos\left(\frac{\pi t}{8}\right)$$

- Triangular y pulso:

$$\text{Difusión: } u(x, t) = \sum_{k=1}^{\infty} A_k \sin\left(\frac{k\pi x}{8}\right) e^{-\pi^2 t}$$

$$\text{Onda: } u(x, t) = \sum_{k=1}^{\infty} A_k \sin\left(\frac{k\pi x}{8}\right) \cos\left(\frac{k\pi t}{8}\right)$$

siendo los coeficientes A_k (factores de amplificación modal) los siguientes:

$$\text{Caso triangular: } A_k = \frac{32}{3k^2\pi^2} \sin\left(\frac{3k\pi}{4}\right)$$

$$\text{Caso pulso: } A_k = \frac{2}{k\pi} \left[\cos\left(\frac{3k\pi}{8}\right) - \cos\left(\frac{5k\pi}{8}\right) \right]$$

En la Figura 1.1 se representa la condición inicial ($t = 0$) correspondiente a 99, 399, 999 y 1599 términos (modos) del sumatorio en (1.25) con la condición inicial pulso. Se observa la dificultad para reproducir adecuadamente las esquinas del pulso, lo cual se consigue con 1599 modos. En la Figura 1.2 se representa la solución para $t = 2$, cuando el pulso ya se ha separado en sus dos partes de amplitud mitad.

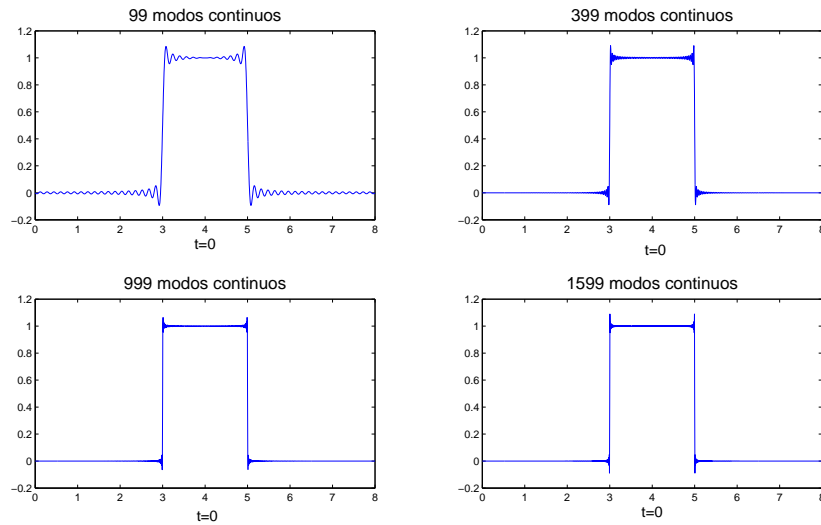


Figura 1.1: Condición inicial del caso del pulso ($t = 0$).

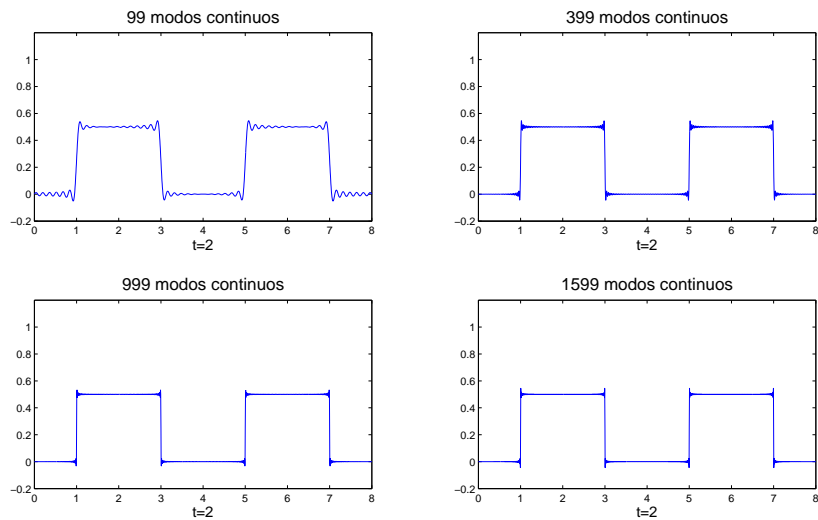


Figura 1.2: Solución del caso del pulso para $t = 2$.

Se observa que esta dificultad también se manifiesta durante el proceso de propagación y que se aminora a medida que la CI es más lisa, ver Figura 1.3, donde para la CI triangular y tiempos $t = 0, 1, 2$, 99 modos son más que suficientes. En este sentido para la CI senoidal (primer modo natural) ya sabemos que la solución exacta tiene un único modo de vibración (que coincide con la CI).

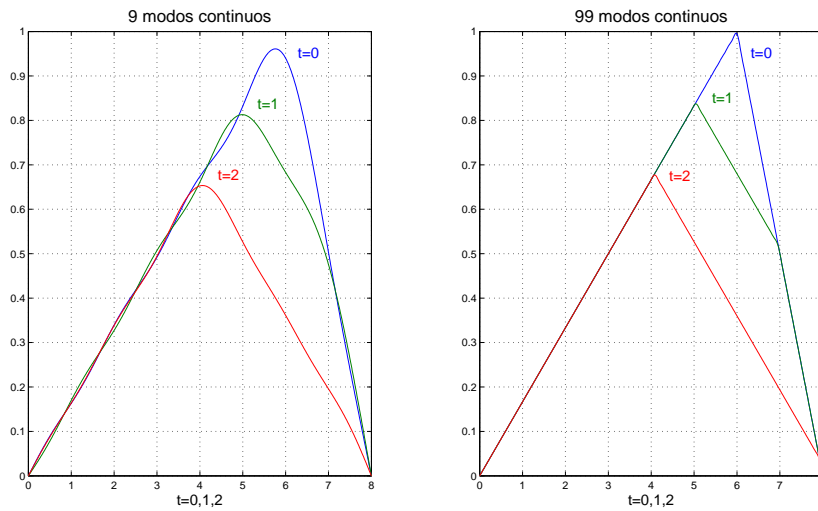


Figura 1.3: Solución del caso triangular para $t = 0, 1, 2$.

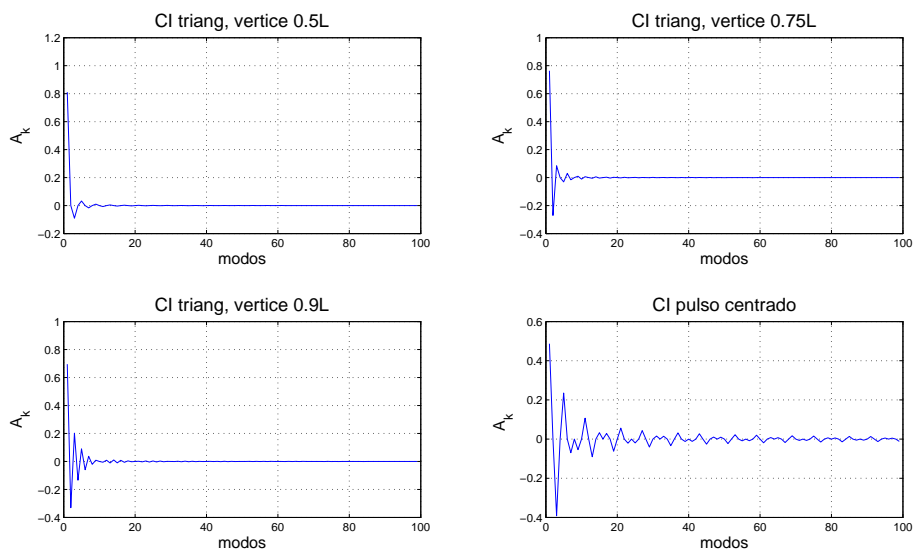


Figura 1.4: Factor de amplificación modal (A_k).

Estos resultados indican que a medida que la CI es más abrupta, se excitan mayor número de modos que participan en la solución, lo cual queda reflejado en la Figura 1.4 para los coeficientes A_k de participación modal. Los tres primeros gráficos de la Figura 1.4 corresponden a condiciones iniciales triangulares, de altura unidad y con el vértice situado en las abscisas $0,5L$, $0,75L$ y $0,9L$ respectivamente. La cuarta corresponde al pulso rectangular centrado de altura unidad que hemos utilizado como tercera CI.

1.2.1.2. Semidiscretización MEF. Solución por superposición modal

La particularización del sistema de EDOs (1.18) a este problema unidimensional mediante semidiscretización MEF con n nodos y $(n - 1)$ elementos es la siguiente:

$$\sum_{j=2}^{n-1} \left[\int_0^L N_i(x) \rho N_j(x) dx \right] d_j''(t) = - \sum_{j=2}^{n-1} \left[\int_0^L N_i'(x) T N_j'(x) dx \right] d_j(t), \quad i = 2, \dots, n-1 \quad (1.26)$$

y en formato matricial viene dado por:

$$M \mathbf{d}''(t) + K \mathbf{d}(t) = 0, \quad \begin{cases} \mathbf{d}(0) = \mathbf{d}_0 = (g(x_2), \dots, g(x_{n-1}))^T \\ \mathbf{d}'(0) = \mathbf{d}'_0 = (0, \dots, 0)^T \end{cases} \quad (1.27)$$

Este sistema admite soluciones armónicas $\mathbf{d}(t) = \hat{\mathbf{d}} \sin(\hat{\omega}t + \varphi)$ donde $\hat{\mathbf{d}}$ es un vector cualquiera de $m = n - 2$ valores nodales. Derivando esta solución se obtiene:

$$\mathbf{d}''(t) = -\hat{\omega}^2 \underbrace{\hat{\mathbf{d}} \sin(\hat{\omega}t + \varphi)}_{\mathbf{d}(t)} = -\hat{\omega}^2 \mathbf{d}(t) \quad (1.28)$$

Sustituyendo (1.28) en (1.27), simplificando y reordenando obtenemos el siguiente problema generalizado de autovalores y autovectores $(K - \hat{\omega}^2 M) \mathbf{d} = 0$, cuya solución proporciona $m = n - 2$ frecuencias propias $\hat{\omega}_i$ y m modos de vibración $\phi_i(x)$ que una vez normalizados expresaremos matricialmente en la forma:

$$\hat{\omega} = \begin{pmatrix} \hat{\omega}_1 \\ \vdots \\ \hat{\omega}_m \end{pmatrix}, \quad \Phi = (\phi_1, \dots, \phi_m) = \begin{pmatrix} \phi_{11} & \cdots & \phi_{1m} \\ \vdots & \ddots & \vdots \\ \phi_{m1} & \cdots & \phi_{mm} \end{pmatrix}$$

verificándose las propiedades de ortogonalidad:

$$\begin{cases} \phi_i^T M \phi_j = 0 \\ \phi_i^T K \phi_j = 0 \end{cases} \quad \forall i \neq j, \quad i, j = 1, \dots, m \quad (1.29)$$

Representamos el vector solución de la discretización MEF mediante superposición modal:

$$\mathbf{d}(t) = \phi_1(x)Y_1(t) + \dots + \phi_m(x)Y_m(t) = \Phi \mathbf{Y} \quad (1.30)$$

Y utilizando (1.30) el sistema de EDOs (1.27) junto con sus CI, se transforma en:

$$M\Phi \mathbf{Y}''(t) + K\Phi \mathbf{Y}(t) = 0, \quad \begin{cases} \mathbf{Y}(0) = \Phi^{-1} \mathbf{d}_0 \\ \mathbf{Y}'(0) = \Phi^{-1} \mathbf{d}'_0 \end{cases} \quad (1.31)$$

donde la nueva incógnita es el vector de coordenadas normales $\mathbf{Y}(t)$. Premultiplicando por el modo genérico ϕ_i y utilizando las propiedades de ortogonalidad (1.29), obtenemos un sistema de m EDOs desacopladas:

$$M_i Y_i''(t) + K_i Y_i(t) = 0, \quad i = 1, \dots, m \quad (1.32)$$

siendo: $\begin{cases} M_i = \phi_i^T M \phi_i \\ K_i = \phi_i^T K \phi_i \end{cases}$ y donde los valores iniciales son:

$$\begin{cases} Y_i(0) = \frac{\phi_i^T M \mathbf{d}_0}{M_i} \\ Y_i'(0) = \frac{\phi_i^T M \mathbf{d}'_0}{M_i} \end{cases} \quad i = 1, \dots, m \quad (1.33)$$

La solución de cualquiera de las m EDOs de (1.32) a partir de las CI (1.33) es:

$$Y_i(t) = \frac{Y_i'(0)}{\hat{\omega}_i} \sin(\hat{\omega}_i t) + Y_i(0) \cos(\hat{\omega}_i t), \quad i = 1, \dots, m$$

Y para el caso particular de velocidad inicial nula:

$$Y_i(t) = Y_i(0) \cos(\hat{\omega}_i t), \quad i = 1, \dots, m$$

Resueltas las m EDOs podemos recuperar la solución (1.30) en las coordenadas geométricas originales.

Observando la similitud en la forma de la solución continua (1.25) y la solución discreta MEF (1.30), nos interesa comparar las frecuencias y modos de vibración del modelo discreto con el modelo continuo. Para la cuerda de longitud $L = 8$ discretizada con 100 elementos tendremos 99 frecuencias discretas $\hat{\omega}_i$ que comparamos en la Figura 1.5 con las primeras 99 frecuencias continuas: $\omega_k = \frac{k\pi}{8}$.

Las frecuencias naturales discretas (MEF) son mayores que las continuas según crece el número del modo (modos más altos), es decir, el error de la discretización MEF afecta

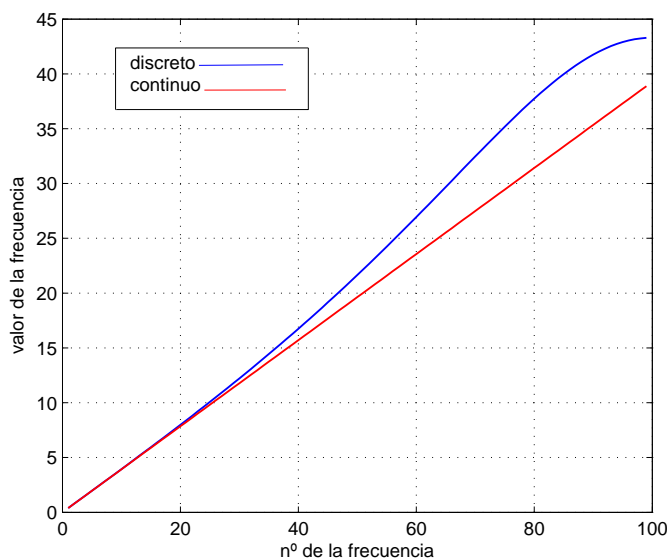


Figura 1.5: Frecuencias del modelo continuo y discreto (masa consistente).

a las frecuencias altas que son las de menor participación modal afectando muy poco a la solución, salvo que la condición inicial tenga contenido en las frecuencias altas.

Efectivamente, la discretización MEF de n grados de libertad es capaz de captar con mucha precisión los primeros modos de vibración del continuo, pero captará muy mal los modos del continuo cercanos a n , ya que el número de ciclos de estos modos altos es muy elevado y resulta imposible captarlos con elementos de longitud $\frac{L}{n}$. En la Figura 1.6 se observan los modos exactos 1, 2 y 10, y los discretos 1, 2, 10.

Se observa la superposición perfecta de todos ellos, excepto del segundo (dibujado en color verde) por una cuestión de signo que es indiferente. Sin embargo, esta similitud se va deteriorando, y los últimos modos discretos difieren de los continuos, como se observa para el modo 99 en las Figuras 1.7 y 1.8.

En las Figuras 1.9 y 1.10 se representan gráficamente la participación modal continua (A_k) y discreta ($Y_i(0)$) de la CI pulso (la de mayor contenido en frecuencias altas de las tres consideradas). Se ve que la participación disminuye significativamente con el aumento de la frecuencia modal y que los modos antisimétricos respecto al centro de la cuerda no participan.

La repercusión de esta diferencia entre los modos discretos y los continuos no es muy grande, ya que en general es muy escasa la participación de los modos altos en la solución. Por tanto, contribuirán erróneamente pero poco, en forma de ruido que acompaña a la solución, como se observa en la Figura 1.11, obtenida para 400 elementos, 399 modos. De nuevo, se reproduce la dificultad para representar las esquinas que ya

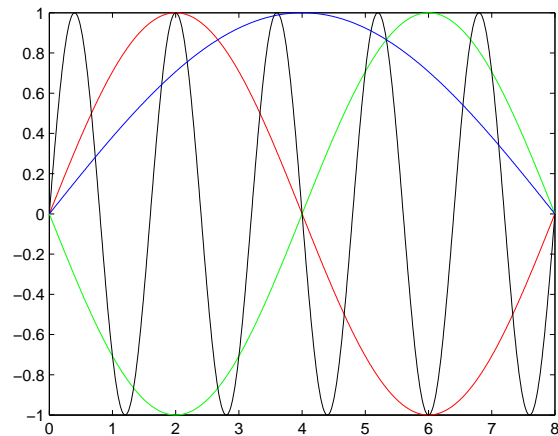


Figura 1.6: Modos 1, 2 y 10 (continuos y discretos).

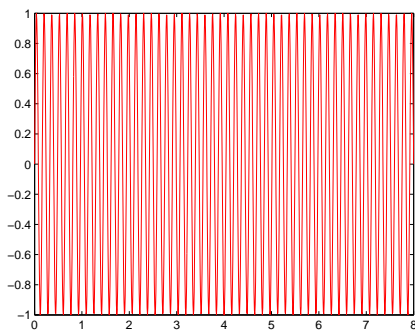


Figura 1.7: Modo 99 del continuo.

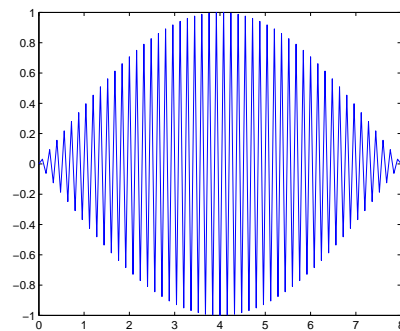


Figura 1.8: Modo 99 del discreto.

tiene el modelo continuo, como pudo verse en la Figura 1.2 para 399 modos continuos. Además, se observan las oscilaciones de ruido achacables a la diferencia entre el modelo discreto MEF y el continuo.

Aumentando la discretización MEF a 1600 elementos y utilizando 1599 modos, mejora el ajuste al pulso pero se mantiene un ruido importante como se observa en la Figura 1.12. A costa de perder algo de precisión este ruido se va reduciendo a medida que quitamos los modos altos pasando por ejemplo de todos los modos (1599) a 399 en la Figura 1.13, a 99 modos en la Figura 1.14 y a 25 modos en la Figura 1.15.

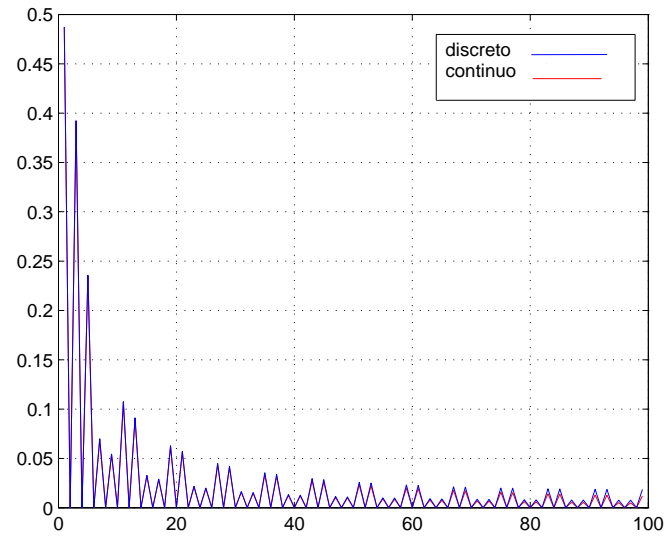


Figura 1.9: Participación modal $|A_k|$, $|Y_i(0)|$ para la CI pulso.

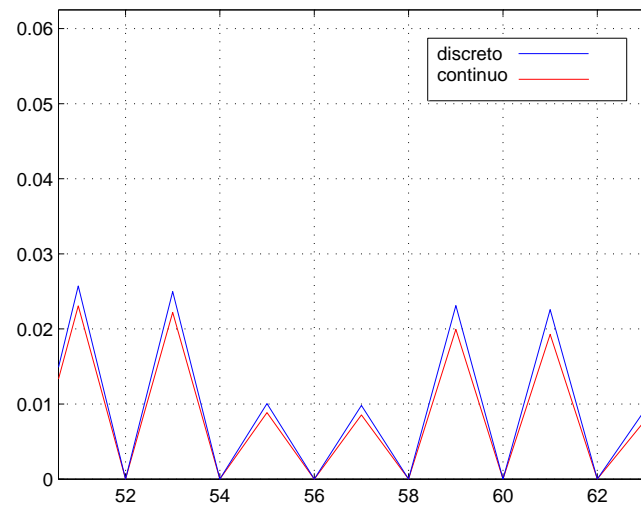


Figura 1.10: Participación modal para la CI pulso (detalle de la Figura 1.9).

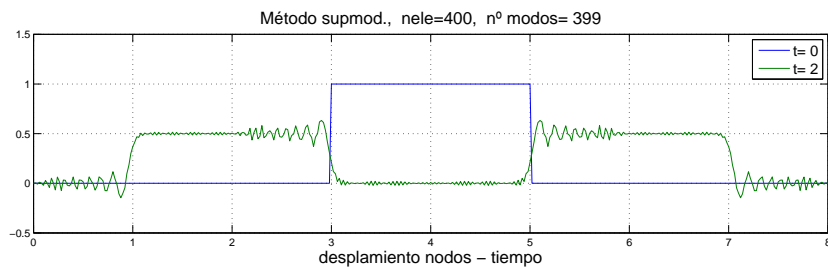


Figura 1.11: Solución del pulso por superposición modal (400 elem., 399 modos).

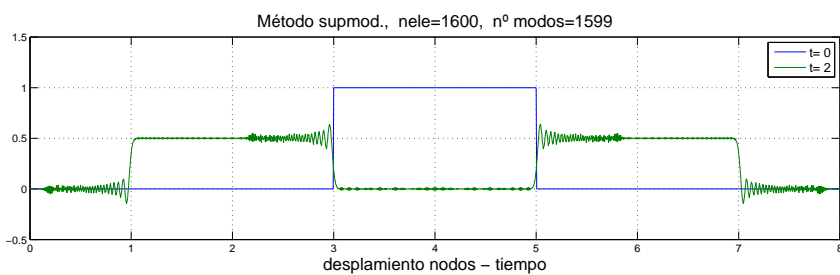


Figura 1.12: Solución del pulso por superposición modal (1600 elem., 1599 modos).

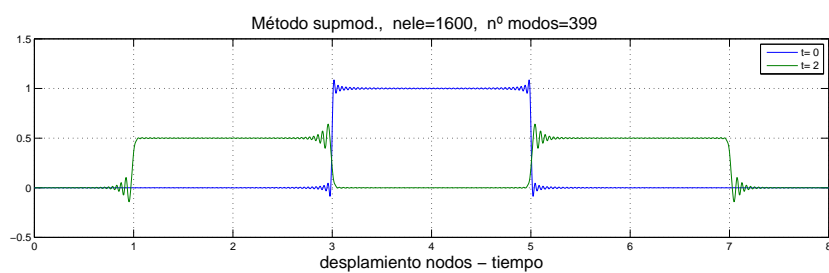


Figura 1.13: Solución del pulso por superposición modal (1600 elem., 399 modos).

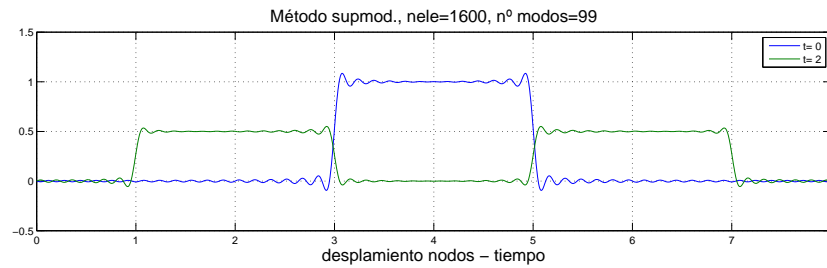


Figura 1.14: Solución del pulso por superposición modal (1600 elem., 99 modos).

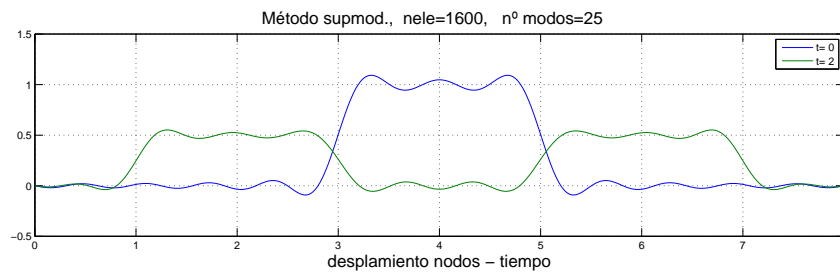


Figura 1.15: Solución del pulso por superposición modal (1600 elem., 25 modos).

Comparamos el resultado de la Figura 1.15 con la solución continua tomando sólo los 25 primeros modos continuos (Figura 1.16), esperando que sean muy parecidas y resulta que efectivamente es así. Ambas soluciones reflejan la dificultad de reproducir las esquinas pero no aparecen efectos de ruidos, ya que los 25 modos MEF utilizados aproximan muy bien los 25 modos continuos.

Lo mismo ocurre, aunque en menor medida, cuando se comparan las superposiciones modales de 99 y 399 modos con las del continuo obtenidas anteriormente (Figura 1.2). La mayor diferencia se observa entre la solución completa de 1599 modos (Figura 1.12) y la continua equivalente de 1599 (Figura 1.2). En este caso la diferencia es achacable en su totalidad al ruido de los modos MEF altos ya que el efecto esquina no se manifiesta con 1999 modos continuos. Este ruido puede ser molesto y puede dificultar la convergencia de los casos no lineales, por lo que puede ser conveniente eliminarlo o reducirlo.

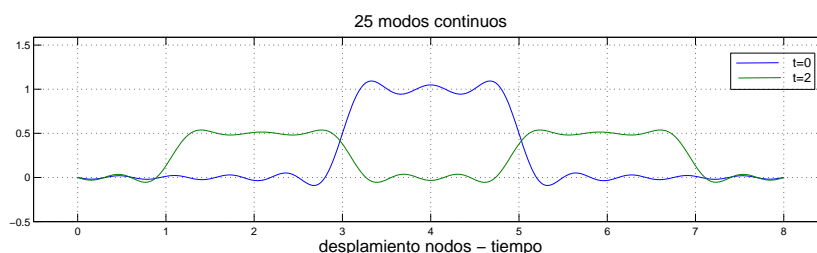


Figura 1.16: Solución del pulso por superposición modal (25 modos continuos).

Una primera conclusión de todo lo anterior es que la aproximación MEF asociada a una discretización no puede representar adecuadamente las formas de los modos altos de vibración. Otra conclusión relacionada con la anterior, es que la semidiscretización MEF conduce a un sistema de EDOs perturbado y su integración exacta mediante superposición modal contiene ruido en la solución.

1.2.2. Ejemplo 2: Solución MEF numérica

En este ejemplo seguimos con las EDPs de difusión y onda y las tres condiciones iniciales: una perturbación senoidal (primer modo natural), una perturbación lineal triangular y un pulso rectangular. Los seis casos se calculan con cuatro métodos distintos: Runge-Kutta de orden 4 (RK4), trapezoidal, *ode45* y *ode15s*. Los dos primeros de paso constante son muy sencillos de programar. El primero es explícito de orden 4 y el segundo es implícito de orden 2 e incondicionalmente estable. Los dos últimos son de paso variable y se encuentran en la *odesuite* de Matlab. La *ode45* es el método más popular: explícito de orden 5, es una implementación de un método Runge-Kutta anidado de órdenes (5, 4) basado en el par Dormand-Prince [23]. La *ode15s* está diseñada especialmente para resolver problemas rígidos (stiff). Basada en los métodos BDF de Gear [30] es un método implícito de orden variable (hasta 5).

1.2.2.1. Difusión

El ejemplo de difusión se estudia para una longitud $L = 8$, un intervalo temporal de $T = 16$ y $\alpha^2 = 1$. La semidiscretización MEF de la EDP $u_t = u_{xx}$ con n nodos y $(n - 1)$ elementos conduce al siguiente sistema de EDOs:

$$\sum_{j=2}^{n-1} \left[\int_0^L N_i(x) \cdot 1 \cdot N_j(x) dx \right] d_j'(t) = - \sum_{j=2}^{n-1} \left[\int_0^L N_i'(x) \cdot 1 \cdot N_j'(x) dx \right] d_j(t), \quad i = 2, \dots, n-1 \quad (1.34)$$

que en formato matricial se escribe:

$$M\mathbf{d}'(t) + K\mathbf{d}(t) = 0, \quad \mathbf{d}(0) = \mathbf{d}_0 = (g(x_2), \dots, g(x_{n-1}))^T \quad (1.35)$$

Escribimos (1.35) en la forma $y' = f(t, y)$:

$$\mathbf{d}'(t) = -M^{-1}K\mathbf{d}(t) = 0, \quad \mathbf{d}(0) = \mathbf{d}_0 = (g(x_2), \dots, g(x_{n-1}))^T \quad (1.36)$$

Con una discretización de 100 elementos los autovalores máximo y mínimo (en valor absoluto) de la matriz jacobiana $\frac{\partial \mathbf{f}}{\partial \mathbf{d}} = -M^{-1}K$ son respectivamente $\lambda_{max} = -1875$ y $\lambda_{min} = -0,1542$. Con 1000 elementos resultan $\lambda_{max} = -187500$ y $\lambda_{min} = -0,1542$, observándose cómo la rigidez del sistema aumenta con la discretización.

1. CI Senoidal

Aplicamos esta condición inicial con una discretización muy elevada de 1000 elementos y observamos la precisión y los tiempos requeridos por los cuatro métodos. En la Figura 1.17 observamos que cualquiera de los 4 métodos proporciona la misma solución que coincide con la exacta (1.24).

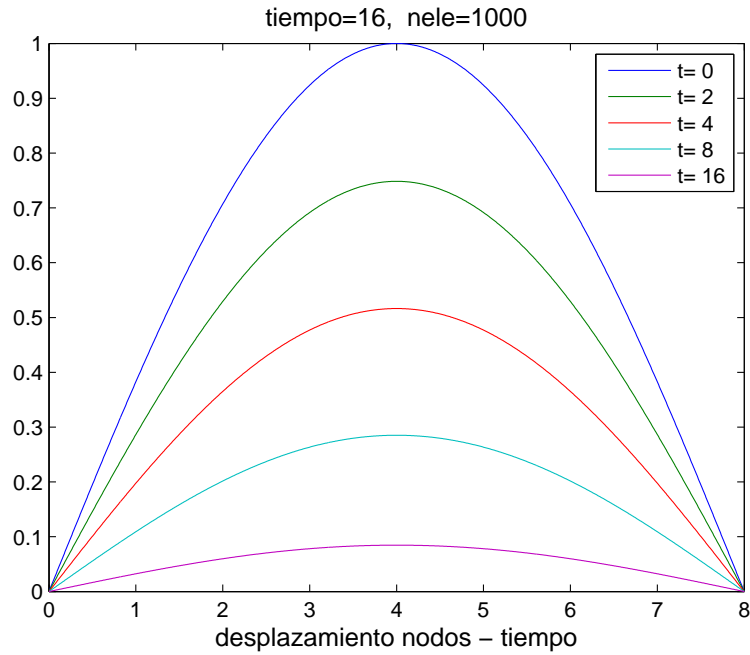


Figura 1.17: Solución de la ecuación de difus. CI senoidal (*ode15s*, *ode45*, RK4, trap.).

La máquina que se ha utilizado para realizar estos cálculos es un PC con un procesador Intel(R) Core(TM)2 Quad CPU Q6600 @2.4GHz con RAM de 4,00GB. Si empezamos por las dos funciones de Matlab, la *ode15s* lo hace muy rápidamente en tan sólo 16 pasos y en un tiempo de 0,18 segundos. Sin embargo, la *ode45* no ha obtenido resultados transcurridos 15 minutos. Ello no es sino manifestación de la rigidez, que crece con la discretización y es muy elevada para 1000 elementos.

El método trapezoidal lo hace perfectamente en 100 pasos constantes y 5,19 segundos, tardando más tiempo que la *ode15s*, aunque también lo puede hacer perfectamente en 10 pasos (3,5 segundos) ya que es incondicionalmente estable y 10 pasos son suficientes para captar la evolución exponencial de la solución exacta (1.24). Se puede observar que el tiempo requerido para 10 pasos es prácticamente el mismo que para 100, debido a que siendo el problema lineal y de coeficientes constantes sólo es preciso resolver una vez al inicio el sistema lineal de ecuaciones, actualizándose en cada paso únicamente el término independiente.

Para que el método RK4 de paso constante sea estable, en este problema es necesario utilizar un gran número de pasos, ya que siendo el intervalo de estabilidad $[-2,78, 0]$ y siendo el mayor autovalor $\lambda_{max} = -187500$, el tamaño de paso máximo es $h_{max} = -2,78/(-187500) = 1,4827 \cdot 10^{-5}$. Por tanto, en el intervalo temporal $[0, 16]$, el número mínimo de pasos estables será más de un millón, $16/h_{max} = 1,0791 \cdot 10^6$ pasos.

Para un número de elementos más moderado $n_{ele} = 100$, $\lambda_{max} = -1875$ y $h_{max} = -2,78/(-1875) = 1,4827 \cdot 10^{-3}$ y el número mínimo de pasos será $16/h_{max} = 10792$. Los resultados obtenidos con 20000 pasos (3,16 segundos) son estables y también muy precisos ya que los 100 elementos son suficientes para representar la CI senoidal. Se obtienen similares resultados con los otros métodos y 100 elementos. En particular la *ode45* responde ahora en 8995 pasos variables y en un tiempo aceptable de 4,70 segundos. No obstante, la *ode15s* sigue siendo muy ventajosa (del orden de 83 veces más rápida que la *ode45*).

2. CI Triangular

Nos centramos en comparar la *ode45* con la *ode15s* con 100 elementos de discretización MEF. Los resultados que se muestran en la Figura 1.18, son prácticamente idénticos con ambos métodos, pero en tiempos de cálculo muy diferentes. La *ode45* precisa 9033 pasos y 4,53 segundos, mientras que a la *ode15s* le bastan 58 pasos y 0,04 segundos. Es decir, la *ode15s* sigue siendo mucho más ventajosa (del orden de 114 veces más rápida que la *ode45*).

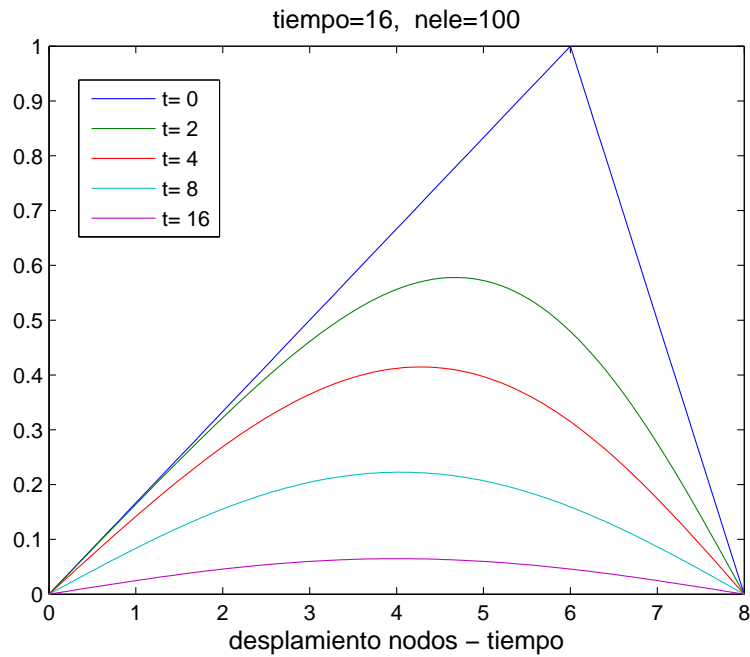


Figura 1.18: Solución de la ecuación de difusión CI triangular (métodos *ode45*, *ode15s*).

3. CI Pulso

También en este caso nos centramos en comparar la *ode45* con la *ode15s* con 100 elementos de discretización MEF, obteniendo resultados que son prácticamente idénticos con ambos métodos, pero en tiempos de cálculo muy diferentes. La *ode45* precisa 9045 pasos y 4,53 segundos, mientras que a la *ode15s* le bastan 142 pasos y 0,12 segundos (Figura 1.19). La *ode15s* sigue siendo mucho más ventajosa (del orden de 38 veces más rápida que la *ode45*) pero no tanto como en los casos anteriores, con CI más lisas. En ambos casos se observa la difusión, es decir el alisamiento de la CI en el tiempo.

4. Conclusiones

De acuerdo con lo esperado, la función especializada para resolver sistemas rígidos (*ode15s*) es mucho más eficiente que la función estándar (*ode45*). Se observa la dificultad creciente con el número de elementos de discretización para resolver el problema con métodos explícitos (RK4) de paso constante, dificultad que no presenta el método trapezoidal que es incondicionalmente estable. Todo ello muestra el carácter rígido del sistema de EDOs proveniente de la semidiscretización MEF, carácter que se acentúa con el número de elementos.

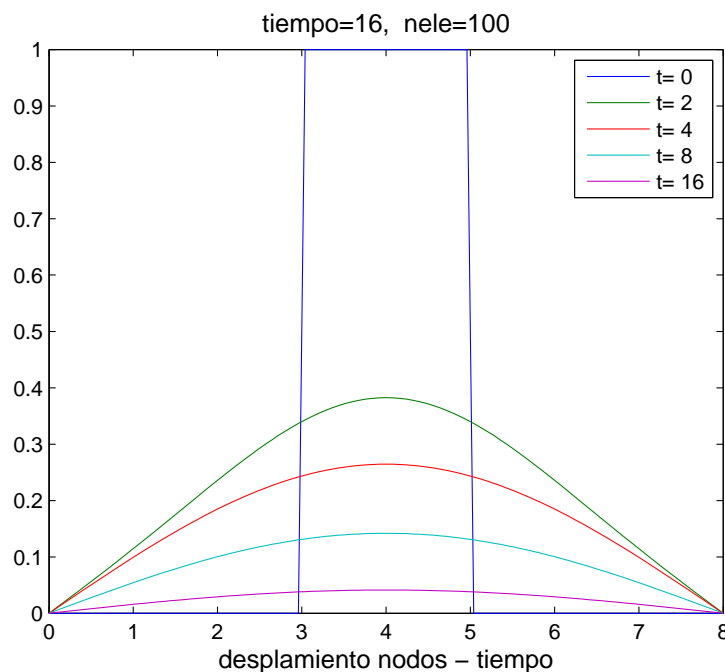


Figura 1.19: Solución de la ecuación de difusión CI pulso (métodos *ode45*, *ode15s*).

1.2.2.2. Onda

El ejemplo de onda se estudia para una cuerda de longitud $L = 8$, un intervalo temporal de $T = 16$ y $\alpha^2 = 1$. La semidiscretización MEF de la EDP $u_{tt} = u_{xx}$ con n nodos y $(n - 1)$ elementos conduce al siguiente sistema de EDOs:

$$\sum_{j=2}^{n-1} \left[\int_0^L N_i(x) \cdot 1 \cdot N_j(x) dx \right] d_j''(t) = - \sum_{j=2}^{n-1} \left[\int_0^L N_i'(x) \cdot 1 \cdot N_j'(x) dx \right] d_j(t), \quad i = 2, \dots, n-1 \quad (1.37)$$

que en formato matricial es:

$$M \mathbf{d}''(t) + K \mathbf{d}(t) = 0, \quad \begin{cases} \mathbf{d}(0) = \mathbf{d}_0 = (g(x_2), \dots, g(x_{n-1}))^T \\ \mathbf{d}'(0) = \mathbf{d}'_0 = (0, \dots, 0)^T \end{cases} \quad (1.38)$$

o en la forma $y' = f(t, y)$:

$$\begin{pmatrix} \mathbf{y}'_1(t) \\ \mathbf{y}'_2(t) \end{pmatrix} = \begin{pmatrix} 0 & I \\ -M^{-1}K & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y}_1(t) \\ \mathbf{y}_2(t) \end{pmatrix}, \quad (1.39)$$

siendo: $\begin{cases} \mathbf{y}_1(t) = \mathbf{d}(t), \mathbf{y}_2(t) = \mathbf{d}'(t) \\ \mathbf{y}_1(0) = \mathbf{d}_0, \mathbf{y}_2(0) = \mathbf{d}'_0 \end{cases}$

En este caso la matriz jacobiana viene dada por: $\frac{\partial \mathbf{f}}{\partial \mathbf{d}} = \begin{pmatrix} 0 & I \\ -M^{-1}K & 0 \end{pmatrix}$

Con una discretización de 100 elementos los autovalores máximo y mínimo de la matriz jacobiana son respectivamente $\lambda_{max} = \pm 43,29i$ y $\lambda_{min} = \pm 0,3927i$. Con 1000 elementos resultan $\lambda_{max} = \pm 433,01i$ y $\lambda_{min} = \pm 0,3927i$, observándose cómo la rigidez del sistema aumenta con la discretización. Vamos a estudiar este problema en los tres casos de CI citados anteriormente: senoidal, triangular y pulso.

1. CI Senoidal

Para una discretización de 100 elementos la Figura 1.20, muestra similares soluciones para los cuatro métodos, que coinciden con la solución exacta (1.25). Sin embargo, los tiempos obtenidos son significativamente distintos. Si nos centramos en las dos funciones de Matlab y para 100, 200 y 1000 elementos, la *ode15s* resuelve respectivamente en 32 pasos y 0,022 segundos, 32 pasos y 0,033 segundos, 32 pasos y 0,078 segundos. La *ode45* lo hace respectivamente en 446 pasos y 0,25 segundos, 1136 pasos y 0,80 segundos y en 6167 pasos y 11,57 segundos. La *ode15s* sigue siendo conveniente pero sólo 11 veces más rápida que la *ode45* con 100 elementos (83 veces más rápida en difusión), del orden de 25 veces con 200 elementos y 149 veces más rápida con 1000 elementos.

De nuevo, hay que utilizar muchos pasos para que el método RK4 de paso constante sea estable con 1000 elementos. El intervalo de estabilidad en el eje imaginario es $[0, 2,82i]$ y en este caso el mayor autovalor de la matriz jacobiana es $\lambda = \pm 433,01i$, con lo que el tamaño de paso máximo es $h_{max} = 2,82i/433,01i = 6,5126 \cdot 10^{-3}$. Y en el intervalo temporal $[0, 16]$, el número mínimo de pasos estables será $16/h_{max} = 2457$ pasos.

Los resultados muestran la vibración de la cuerda en un octavo de ciclo (2 segundos), en un cuarto de ciclo (4 segundos), en medio (8 segundos) y un ciclo completo (en 16 segundos, retorno a la posición de equilibrio).

2. CI Triangular

Comparamos la *ode45* con la *ode15s* con 100 elementos de discretización MEF. Los resultados que se muestran en la Figura 1.21 son prácticamente idénticos con ambos métodos y en tiempos de cálculo similares: 999 pasos y 0,60 segundos con la *ode45* y 2414 pasos y 0,87 segundos con la *ode15s*, lo cual contrasta con lo obtenido en difusión. En la misma figura se observa cómo evoluciona la forma de la cuerda en un ciclo de 16 segundos en el que recupera su forma inicial.

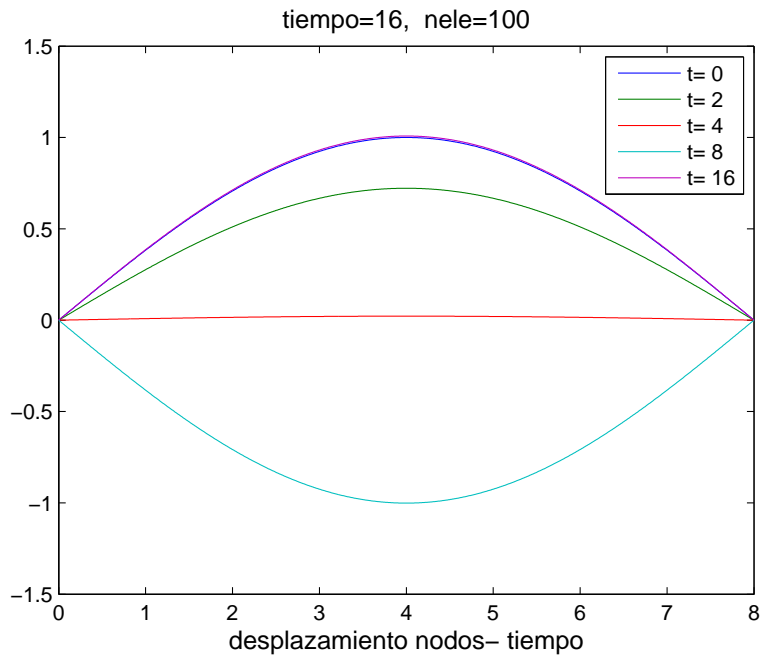


Figura 1.20: Solución de la ecuación de onda CI senoidal (*ode15s*, *ode45*, RK4, trap.).

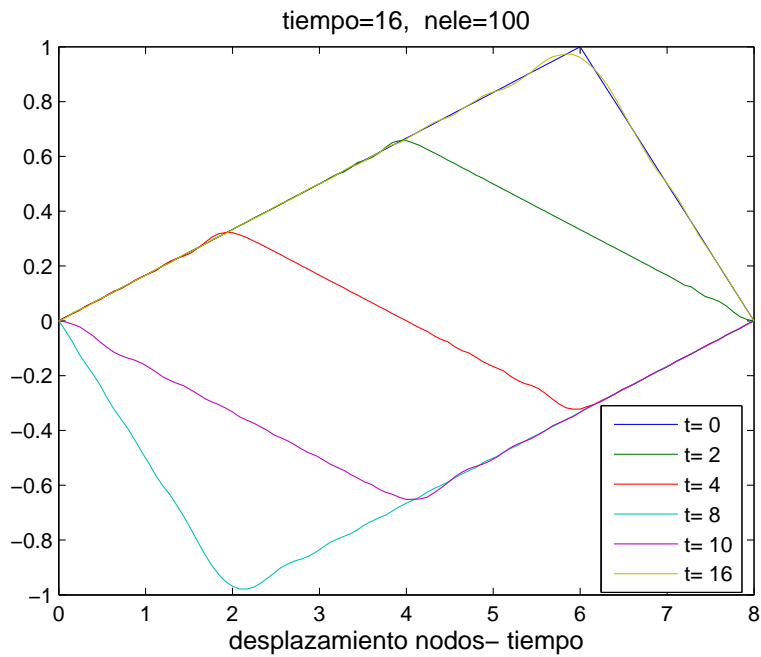


Figura 1.21: Solución de la ecuación de onda CI triangular (métodos *ode15s* y *ode45*).

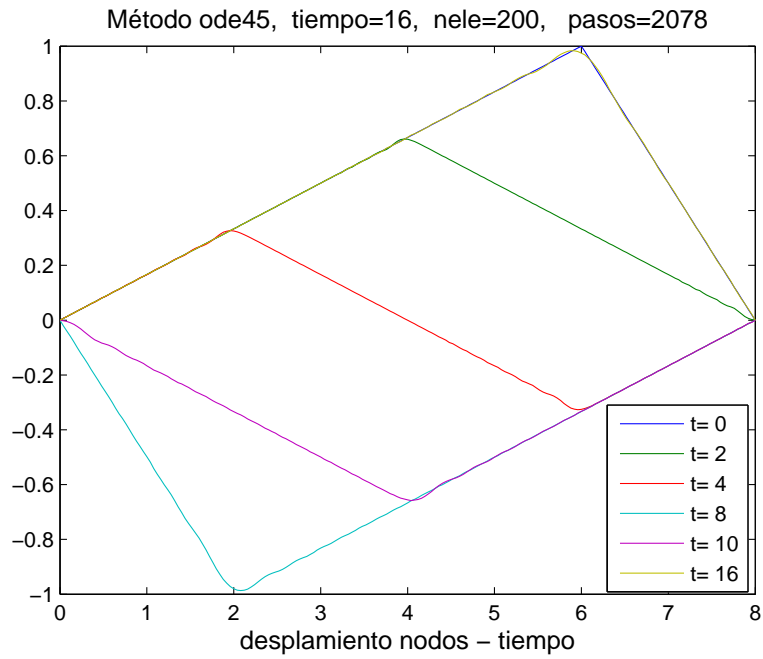


Figura 1.22: Solución de la ecuación de onda CI triangular (método *ode45*).

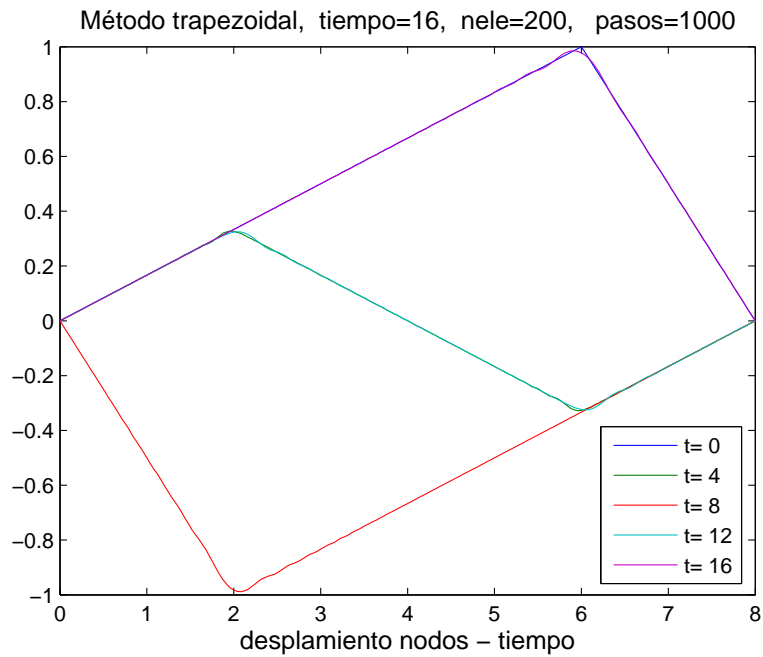


Figura 1.23: Solución de la ec. de onda CI triangular (método trap. 1000 pasos).

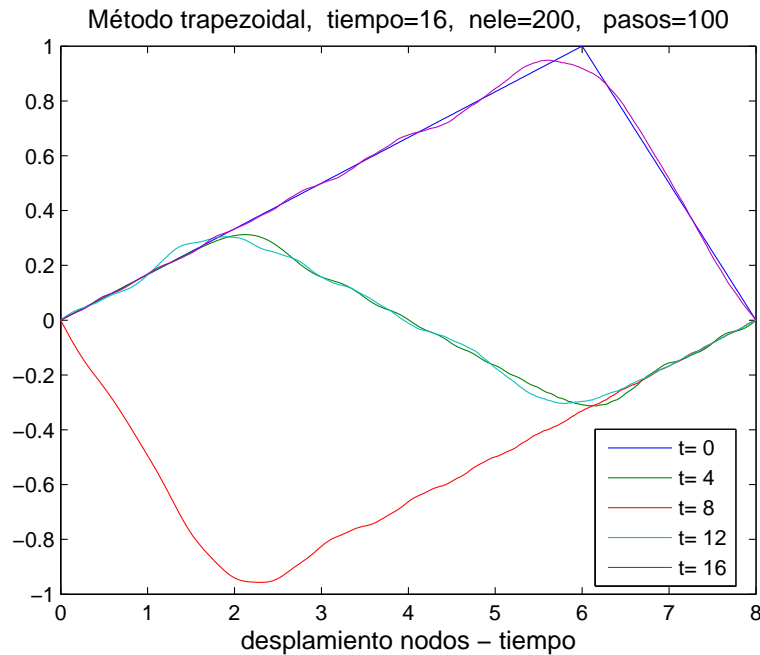


Figura 1.24: Solución de la ec. de onda CI triangular (método trap. 100 pasos).

Con 200 elementos mejora la precisión para la *ode45*, 2078 pasos y 2,69 segundos, como se observa en la Figura 1.22, mientras que la *ode15s* precisa de 4964 pasos y 3,56 segundos, por lo que estamos muy lejos de las diferencias observadas en difusión.

Nos interesamos también por el método trapezoidal, observando que 1000 pasos constantes (0,75 segundos) mantienen la precisión de los resultados como se observa en la Figura 1.23. Sin embargo, 100 pasos constantes de integración (0,041 segundos) no son suficientes (10 lo eran en difusión) ya que la precisión de la solución se deteriora como se observa en la Figura 1.24.

En las dos últimas figuras los tiempos mostrados corresponden a 0, 1/4, 1/2, 3/4 y 1 periodos, por lo que se superponen las formas de ida y vuelta a mitad de recorrido.

3. CI Pulso

También para la CI pulso nos centramos en comparar la *ode45* con la *ode15s* con 100 elementos de discretización MEF, obteniendo resultados que son prácticamente idénticos con ambos métodos Figura 1.25, y en tiempos de cálculo similares. La *ode45* precisa 1101 pasos y 0,62 segundos y la *ode15s* 2857 pasos y 1,38 segundos, lo cual contrasta con lo obtenido en difusión y acentúa lo expuesto en el caso triangular anterior.

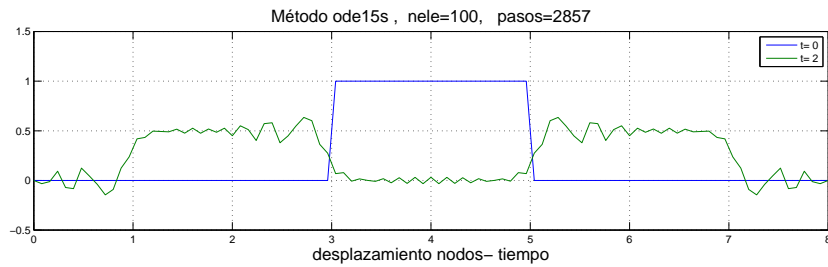


Figura 1.25: Solución de la ec. de onda CI pulso (método *ode15s* 100 elem.).

Los resultados son muy similares con el método trapezoidal, con 100 elementos y dando 1000 pasos constantes (0,092 segundos), como se observa en la Figura 1.26.

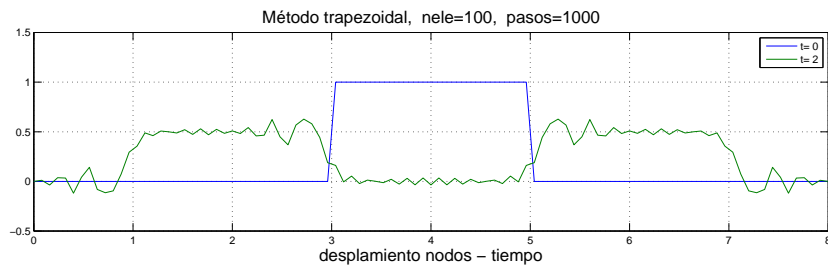


Figura 1.26: Solución de la ec. de onda CI pulso (método trap. 100 elem. 1000 pasos).

Afinando la discretización, y por tanto aumentando la rigidez del sistema, la *ode15s* de paso variable pierde eficiencia frente al método trapezoidal de paso constante. Para 1000 elementos la *ode45* (13135 pasos y 24,28 segundos) proporciona resultados prácticamente idénticos que la *ode15s* (34811 pasos y 162,52 segundos), Figura 1.27. Por lo que, la función especializada no muestra la eficiencia esperada.

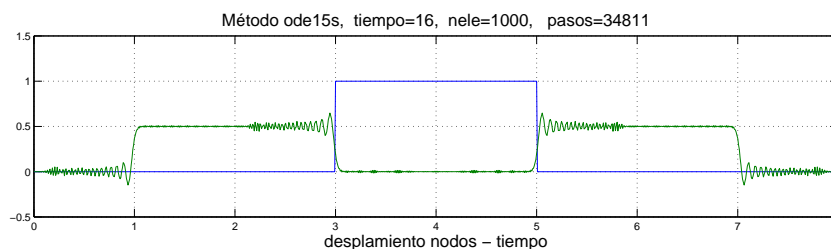


Figura 1.27: Solución de la ec. de onda CI pulso (método *ode15s* 1000 elem.).

El método trapezoidal con 1000 elementos sólo precisa 17,63 segundos, pero se aprecia en los resultados de la Figura 1.28 una diferencia en el rizo, que aparece en distinta posición que el rizo de la *ode15s*. Para aclarar este aspecto compararemos los resultados obtenidos mediante el método trapezoidal y la *ode15s* en una discretización de 400 elementos con los obtenidos por superposición modal sobre la misma discretización Figura 1.11.

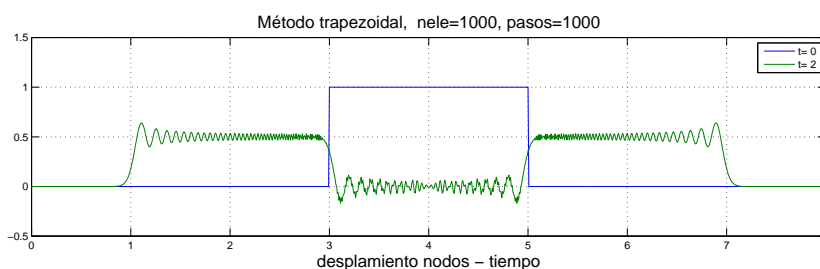


Figura 1.28: Solución de la ec. de onda CI pulso (método trap. 1000 elem. 1000 pasos).

La Figura 1.29 muestra la concordancia de los resultados obtenidos por la *ode15s* con los obtenidos por superposición modal mientras que la Figura 1.30 muestra la discrepancia para el método trapezoidal. Cabe pensar por tanto, que la *ode15s* integra con mayor precisión que el método trapezoidal el sistema de EDOs. Esta hipótesis queda confirmada en la Figura 1.31, donde el método trapezoidal (2000 pasos) ya reproduce los resultados de la superposición modal.

Por último y siguiendo con la misma discretización de 400 elementos, aplicamos el método “alfa” de Hilber-Hughes-Taylor (HHT- α) [43], que proviene del método de Newmark y se comporta de manera similar al trapezoidal pero eliminando las altas frecuencias. Para 1400 pasos en un intervalo de 16 segundos, se observa en la Figura 1.32 la práctica desaparición del rizo conforme a lo esperado.



Figura 1.29: Solución de la ec. de onda CI pulso (método *ode15s* 400 elem.).

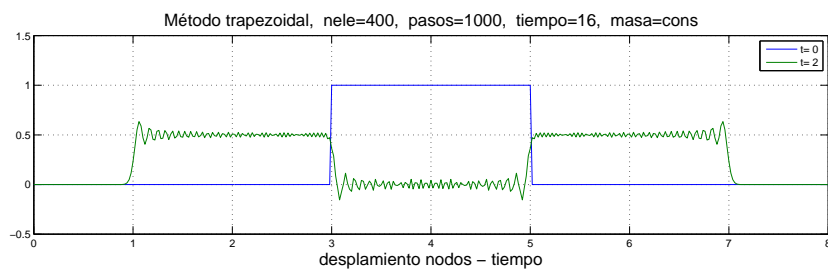


Figura 1.30: Solución de la ec. de onda CI pulso (método trap. 400 elem. 1000 pasos).



Figura 1.31: Solución de la ec. de onda CI pulso (método trap. 400 elem. 2000 pasos).

Existe otra forma de eliminar el rizo en este problema unidimensional, que consiste en compensar los errores provenientes de la semidiscretización y los provenientes del método numérico. Esto ocurre con el método Newmark ($\beta = \frac{1}{6}$, $\gamma = 0,5$) si se sincroniza la longitud del elemento con el tamaño de paso, de forma que en un paso la onda se propaga exactamente la longitud de un elemento [47]. Efectivamente, en la Figura 1.33 se observa la superconvergencia debido a la compensación total de ambos tipos de error.

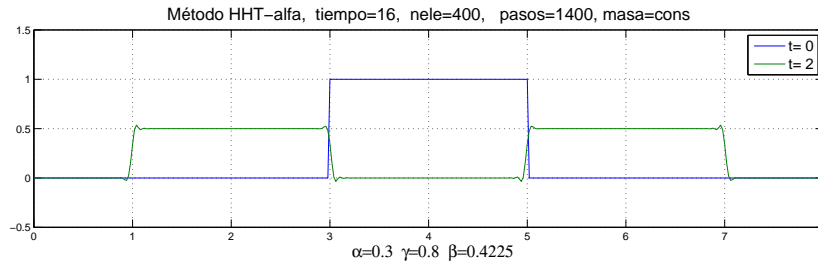


Figura 1.32: Solución de la ec. de onda CI pulso (mét. HHT- α 400 elem. 1400 pasos).

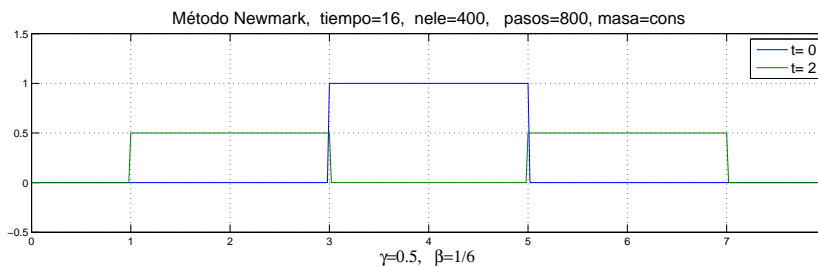


Figura 1.33: Solución ec. onda CI pulso (mét. Newmark $\beta = \frac{1}{6}$, $\gamma = 0,5$ 400 elem.).

4. Conclusiones

En contra de lo esperado, la *ode15s* no es más eficiente que la *ode45* en el problema de la onda. Ambas son precisas (paso adaptativo variable) pero son significativamente más lentas que el método trapezoidal, especialmente programado para sistemas lineales de coeficientes constantes. Sin embargo, la precisión del método trapezoidal depende del tamaño de paso constante especificado para la integración.

En ambas funciones, *ode45* y *ode15s*, se observa la dificultad creciente para resolver el problema al aumentar el número de elementos de la discretización, dificultad que no se aprecia con el método trapezoidal. Ello muestra el carácter rígido del sistema de EDOs proveniente de la semidiscretización MEF, carácter que se acentúa con el número de elementos.

El error de los modos MEF asociados a las frecuencias altas introduce ruido en la solución con independencia del método de integración temporal utilizado y del tamaño de paso especificado.

Las fuentes de error de la solución numérica son la semidiscretización MEF y el truncamiento del algoritmo numérico de integración, pudiendo cancelarse ambas como ocurre en los casos de superconvergencia en 1D.

También es posible eliminar el error asociado a las altas frecuencias si el algoritmo de integración introduce fuerte disipación numérica de las altas frecuencias, como ocurre con el método HHT- α .

1.2.3. Ejemplo 3: Una versión no lineal de la ecuación de onda

El objeto de este ejemplo es similar al anterior, una comparación experimental de distintos algoritmos para la integración temporal de sistemas de EDPs provenientes de la semidiscretización MEF, pero en este caso aplicado a una EDP no lineal, en particular a una versión no lineal de la cuerda de la guitarra que se describe en el Apéndice A y cuya forma es:

$$\rho u_{tt}(x, t) = \left(T + E \cdot S \left(\sqrt{1 + u_x^2(x, t)} - 1 \right) \right) u_{xx}(x, t) \quad (1.40)$$

donde el tiro resultante (fuerza de tensión en la cuerda) es suma del aplicado inicialmente en el montaje (componente lineal T) y el resultante de la deformación elástica de la cuerda que viene dado por la expresión $\bar{T} = E \cdot S \left(\sqrt{1 + u_x^2(x, t)} - 1 \right)$, despreciable sólo bajo la hipótesis de pequeñas deformaciones.

Se utilizan datos reales de una cuerda de guitarra [29]: $L = 0,648$ m, diámetro $diam = 0,41 \cdot 10^{-3}$ m, $S = 0,25 \cdot \pi \cdot diam^2$, frecuencia $freq = 329$, tiro $T = 1,8002 \cdot 10^2$. La tensión de montaje se ajusta para que la primera frecuencia fundamental (octava) sea 329 y se considera el primer modo de vibración natural como condición inicial y un intervalo temporal de 5 ciclos o periodos (0,015198 segundos) correspondiente al comportamiento lineal con pequeñas deformaciones.

Observaremos cuáles son los resultados con 2, 20 y 100 elementos. En primer lugar, comprobaremos la diferencia entre el caso lineal y el no lineal con una discretización muy sencilla de 2 elementos y por tanto, un único grado de libertad activo. Los resultados obtenidos son similares con el método trapezoidal o la *ode15s*. El caso lineal $\bar{T} = 0$ con 2 elementos se muestra en la Figura 1.34, donde se observan los 5 ciclos y algo más, atribuible al error de la discretización. Cuando resolvemos el caso lineal con 100 elementos, ya son cinco ciclos exactos, Figura 1.35.

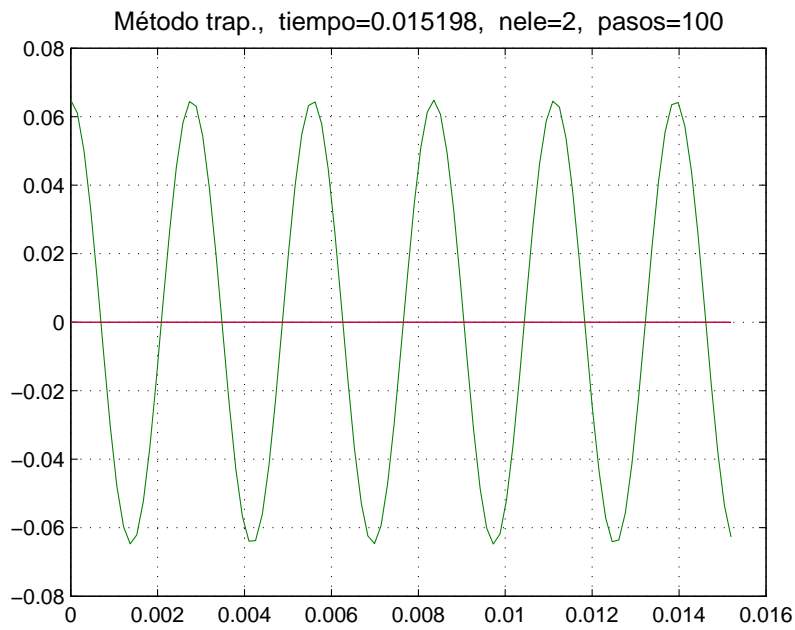


Figura 1.34: Solución ec. onda lineal (método trapezoidal 2 elem.).

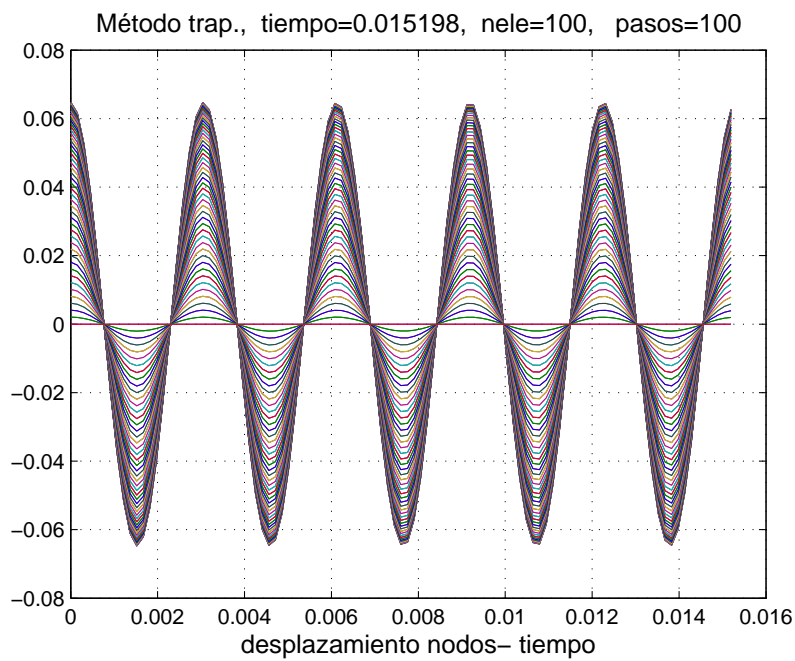


Figura 1.35: Solución ec. onda lineal (método trapezoidal 100 elem.).

En el caso no lineal con 2 elementos, imagen izquierda de la Figura 1.36, aparecen 9 ciclos en vez de 5, como consecuencia de que el tiro ya no permanece constante sino que se incrementa con la aportación no lineal. Cuando resolvemos este caso con la *ode15s*, se obtienen resultados similares, imagen derecha de la Figura 1.36.

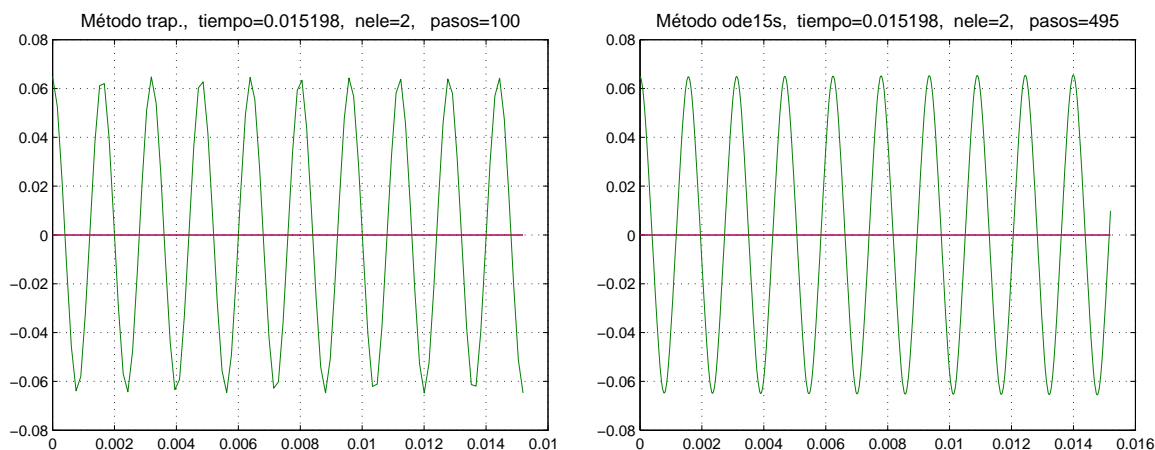


Figura 1.36: Solución ec. onda no lineal (método trapezoidal y *ode15s* 2 elem.).

Si ampliamos el número de elementos a 20 y el intervalo temporal lo reducimos al correspondiente a un periodo del problema lineal (0,0030395), con el método trapezoidal obtenemos los resultados de la imagen izquierda de la Figura 1.37, siendo el tiempo computacional de 19,65 segundos en 200 pasos. Con la *ode15s* obtenemos resultados similares, en 1410 pasos adaptativos pero un tiempo sensiblemente inferior de 12,36 segundos, imagen derecha de la Figura 1.37. Se observa como antes una respuesta de dos ciclos en lugar de uno, debido a que la tensión de la cuerda aumenta con la deformación. También se observa que la tendencia de tiempos (coste computacional) se invierte, a favor de la *ode15s* frente al método trapezoidal. Esto tiene su explicación en que dando 1410 pasos adaptativos recalcula el jacobiano sólo 6 veces. Por contra, el método trapezoidal está programado para que calcule la matriz jacobiana en cada uno de los 200 pasos.

En cuanto a eficiencia, se confirman estos resultados con 100 elementos, donde la *ode15s* requiere 8291 pasos para la resolución y tiempo de computación 351,64 segundos, Figura 1.38. Utilizando el método trapezoidal con 200 pasos se aprecia una degradación de la precisión, imagen izquierda de la Figura 1.39. Con este mismo método pero dando 1000 pasos, los resultados son similares a los obtenidos con la *ode15s* pero el tiempo computacional es mayor, 490,66 segundos, imagen derecha de la Figura 1.39.

Volviendo a la discretización con 20 elementos, y tanto para el método trapezoidal de paso constante, como para la *ode15s* de paso variable, Figura 1.37, se observa la influencia de las altas frecuencias, en forma de vibración rápida y pequeña amplitud (ruido) sobre la vibración principal más lenta y de mayor amplitud. Sin embargo, en la

Figura 1.40 observamos que el método HHT- α de paso constante alisa los resultados de manera apreciable con $\alpha = 0,3$.

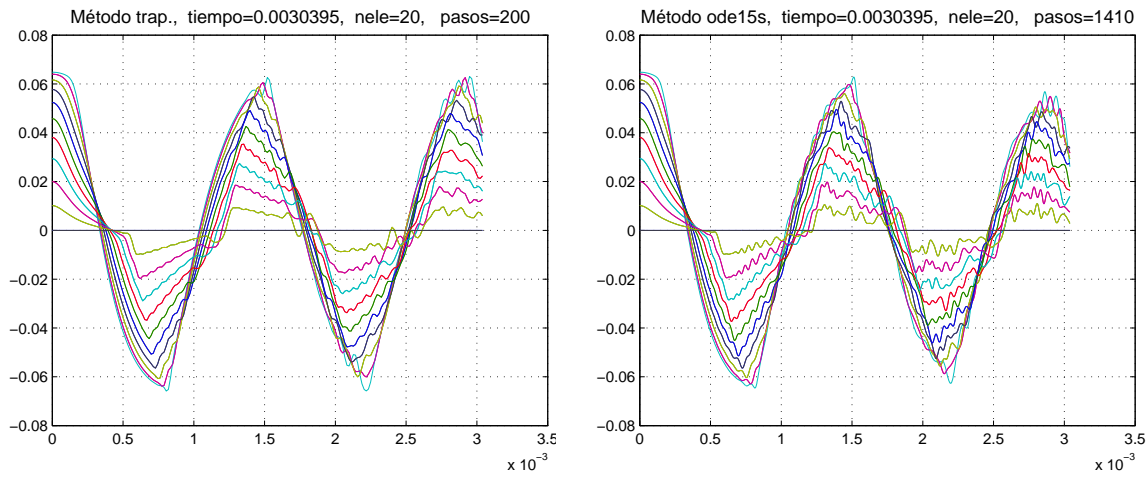


Figura 1.37: Solución ec. onda no lineal (método trapezoidal y *ode15s* 20 elem.).

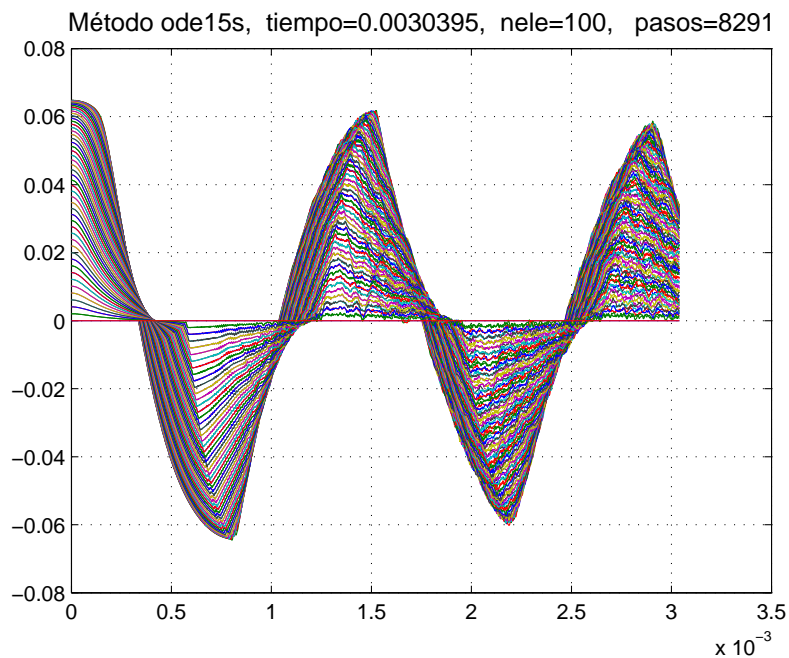


Figura 1.38: Solución ec. onda no lineal (método *ode15s* 100 elem.).

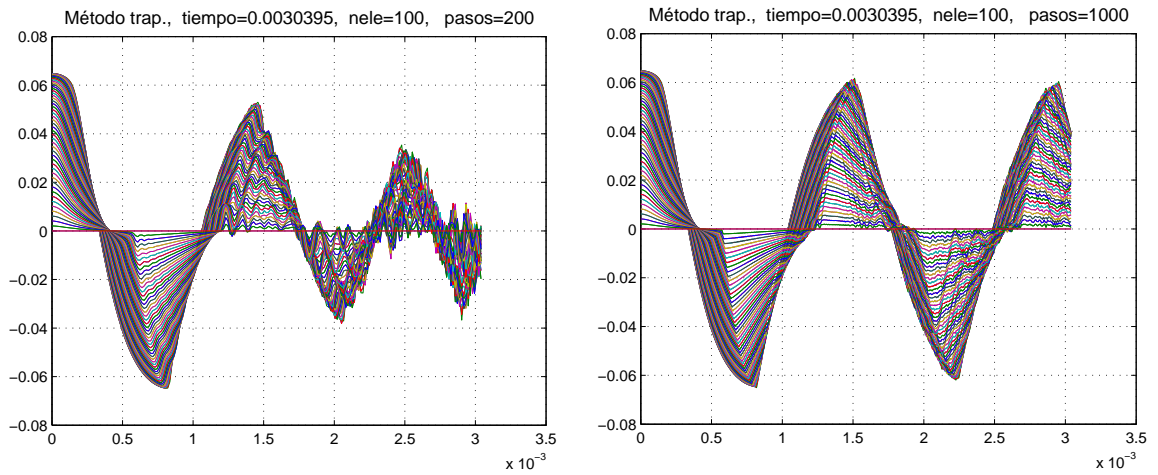


Figura 1.39: Solución ec. onda no lineal (mét. trapezoidal 100 elem. 200 y 1000 pasos).

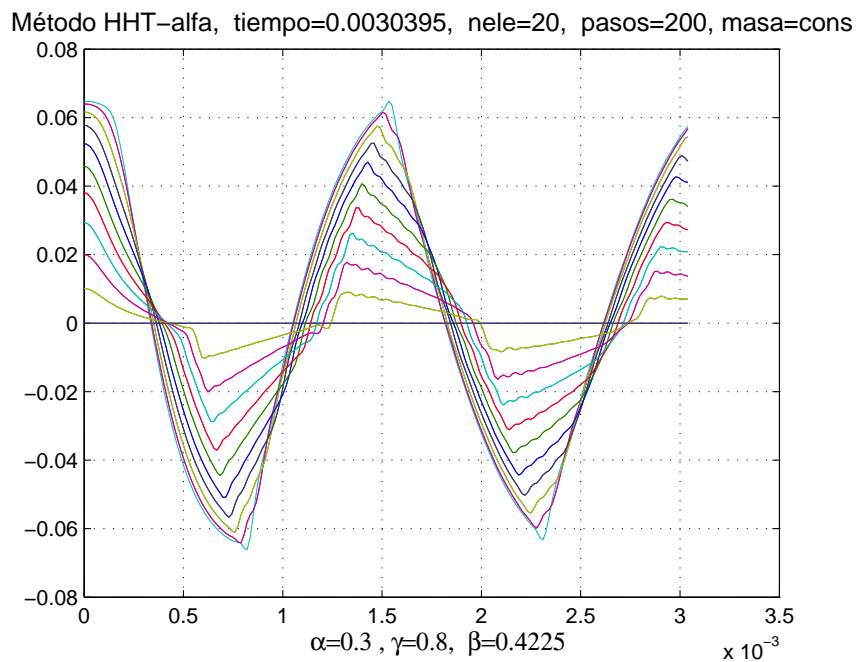


Figura 1.40: Solución ec. onda no lineal (HHT- α , 20 elementos).

También en este sentido, observamos que si alargamos el intervalo temporal (a 5 periodos lineales por ejemplo) los resultados obtenidos con la *ode15s* (tiempo computacional 73,61 segundos) pierden precisión, imagen derecha de la Figura 1.41. Sucede lo mismo cuando utilizamos el método trapezoidal con 1000 pasos (tiempo de computación 103,35 segundos), imagen izquierda de la Figura 1.41.

Incrementando el número de pasos del método trapezoidal hasta 5000, conseguimos que los resultados obtenidos sean similares a los de la *ode15s*, pero el tiempo de computación es sensiblemente mayor. Cabe preguntarse por la influencia de las altas frecuencias en la pérdida de precisión, y efectivamente comprobamos que en este caso el método HHT- α proporciona resultados más aceptables con 1000 pasos constantes, Figura 1.42, que el método trapezoidal (imagen izquierda de la Figura 1.41).

El tiempo que necesita en este caso el método HHT- α es de 86,39 segundos, que es menor que el tiempo requerido por el método trapezoidal (103,35 segundos), lo cual parece indicar que la resolución no lineal de cada paso requiere más iteraciones en presencia de las altas frecuencias, frecuencias que sin embargo tienen poca participación en la solución (y la que tienen es mala, ya que están asociadas a modos artificiales de vibración). Conviene por tanto amortiguar rápidamente (eliminar la aportación a la solución de las altas frecuencias), para mejorar la precisión y también la eficiencia computacional al reducirse el número de iteraciones necesarias para resolver el paso no lineal.

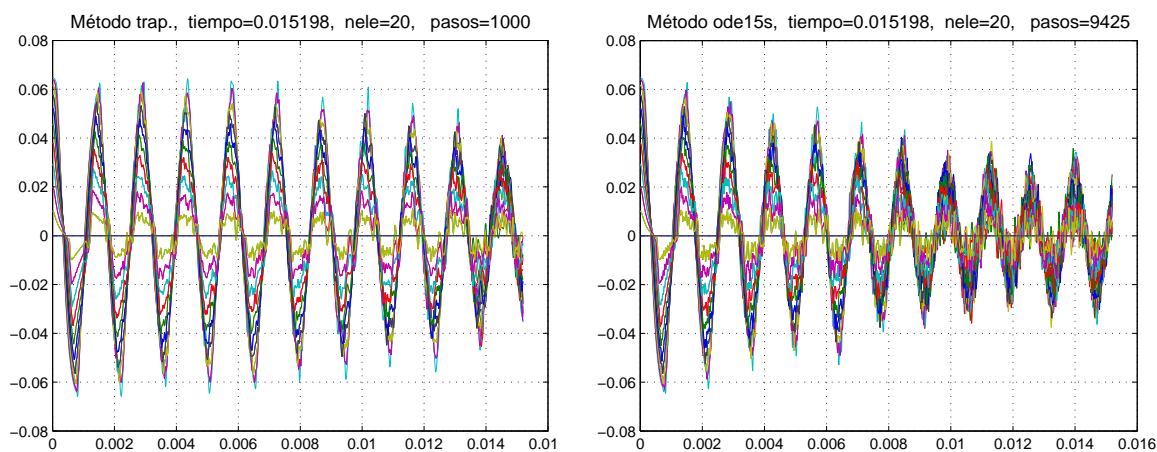


Figura 1.41: Solución ec. onda no lineal (mét. trap. y *ode15s*, 20 elementos).

1.3. Objetivos y organización del trabajo

En las secciones anteriores de introducción y ejemplos hemos constatado:

- El interés en resolver Ecuaciones en Derivadas Parciales (EDPs) con Condiciones Iniciales (CI) y Condiciones de Contorno (CC) y la conveniencia de hacerlo mediante el Método de Elementos Finitos (MEF) en muchas aplicaciones de ingeniería, en particular en dominios de forma cualquiera.

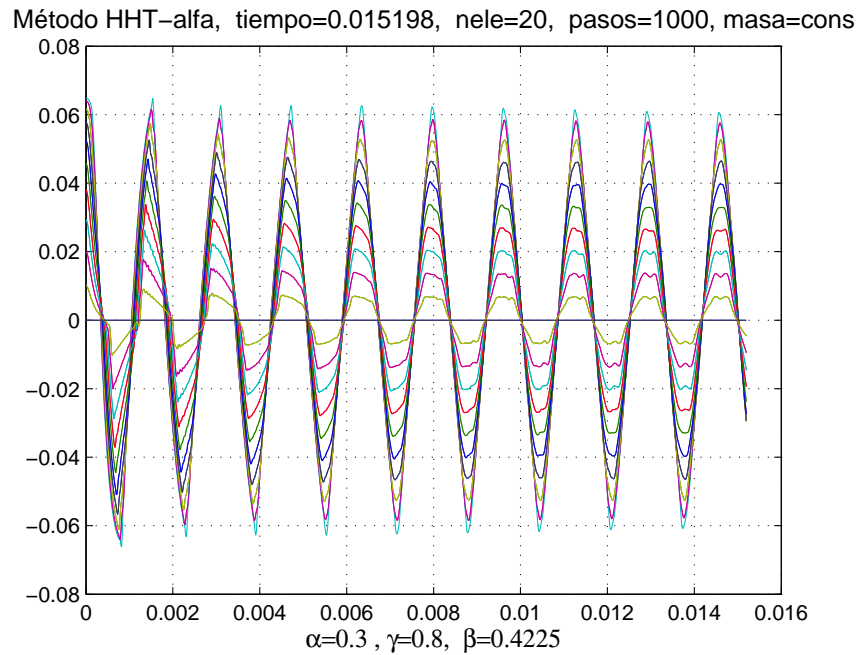


Figura 1.42: Solución ec. onda no lineal (HHT- α , 20 elementos).

- La conveniencia de trabajar con herramientas computacionales avanzadas como la Programación Orientada a Objeto (POO) que faciliten la reutilización, la extensión y el mantenimiento de los códigos que implementan los algoritmos de resolución.
- La opción de hacerlo con Matlab, tanto para la docencia como para el desarrollo de conceptos y prototipos en la investigación. Opción avalada por la creciente utilización de Matlab en el ámbito académico e industrial, por su facilidad de uso, y por la potencia de los recursos computacionales que proporciona, así como por las posibilidades que ofrece para la programación, incluyendo la POO.

También hemos observado que:

- La semidiscretización MEF de las EDPs con CI y CC, conduce a sistemas de Ecuaciones Diferenciales Ordinarias (EDOs), cuyos coeficientes se calculan modularmente, ensamblando integrales numéricas realizadas sobre los elementos de la discretización, cuyos integrandos reflejan las características de las EDPs.
- Afinar la discretización MEF para mejorar la precisión incrementa la rigidez del sistema de EDOs, dificultando su resolución numérica.
- Es posible aprovechar la *odesuite* de Matlab, para resolver numéricamente sistemas de EDOs. En particular, hemos trabajado con la función estándar *ode45* y con la función *ode15s*, que está especialmente adaptada para la resolución de sistemas rígidos, siendo ambas funciones de paso adaptativo.

- Es posible desarrollar con facilidad y en pocas líneas de programación funciones Matlab de paso constante para la integración numérica de sistemas de EDOs. En particular, hemos implementado el método Runge-Kutta de orden 4 (RK4), el método trapezoidal y el método HHT- α .

Finalmente, hemos contrastado experimentalmente los cinco métodos mencionados en las líneas anteriores, con tres casos de CI: senoidal, triangular y pulso, aplicados a la ecuación de difusión y a la ecuación de onda lineal y no lineal, y hemos obtenido las siguientes conclusiones:

- El excelente comportamiento de la *ode15s* frente a la *ode45*, al resolver sistemas lineales rígidos de EDOs de orden 1 procedentes de la semidiscretización MEF del problema de difusión, como cabía esperar.
- La menor eficiencia, en contra de lo esperado, de la *ode15s* frente a la *ode45* en la resolución de los sistemas lineales rígidos de EDOs de orden 1 procedentes de la semidiscretización MEF de problemas de onda y vibraciones.
- La eficiencia del método trapezoidal de paso constante (orden de precisión 2), cuando está especialmente adaptado para la resolución de sistemas lineales de EDOs de coeficientes constantes, frente a la *ode15s* de paso variable, a pesar de que esta última puede dar pasos de orden de precisión hasta 5.
- La similar eficiencia del método HHT- α para la resolución directa de sistemas de EDOs lineales de orden 2 procedentes de la ecuación de onda, frente al método trapezoidal (sistemas equivalentes de orden 1).
- La capacidad del método HHT- α para amortiguar paramétricamente las altas frecuencias de la semidiscretización MEF del problema de onda (vibraciones mecánicas).
- La eficiencia del método HHT- α frente al trapezoidal en la resolución MEF de los problemas dinámicos no lineales, en base a su capacidad para eliminar las altas frecuencias de la vibración.

En este contexto este trabajo pretende:

- Conocer la arquitectura y contrastar la algorítmica de la *ode15s* frente a la *ode45*, identificando los aspectos que caracterizan a la *ode15s* como especialmente adaptada para la resolución de sistemas rígidos.
- Analizar las contradicciones observadas en la aplicación de la *ode15s* a la resolución de sistemas de EDOs rígidos de orden 2.
- Conocer la algorítmica de los métodos clásicos de la mecánica computacional para EDOs de orden 2: método Newmark y método HHT- α , en particular, los aspectos relacionados con la eliminación de las altas frecuencias de la semidiscretización MEF.

- Comparar los métodos BDF de paso variable de la *ode15s*, y los de paso constante con los métodos clásicos de la mecánica computacional: método Newmark y método HHT- α .
- Proponer alternativas BDF para los sistemas de EDOs de orden 1 que ensanchen las regiones de estabilidad, en particular con métodos BDF extendidos.
- Proponer alternativas BDF para los sistemas de EDOs de orden 2 que como el método HHT- α permitan parametrizar la eliminación de altas frecuencias.
- Profundizar en el desarrollo de alternativas de programación, proponiéndose una metodología orientada a objeto que aprovecha la modularidad del MEF para generar un entorno de desarrollo protegido donde sea posible añadir nuevas ideas y desarrollos sin modificar las anteriores.

El trabajo se ha desarrollado con este fin y ha quedado organizado de la siguiente manera:

El Capítulo 2 está dedicado a la revisión y estado del arte de los métodos numéricos para la resolución de EDOs de orden 1, en particular los métodos BDF [30] y los métodos Runge-Kutta, que son la base de las funciones *ode15s* y *ode45* de Matlab respectivamente. También se han revisado las líneas seguidas para construir métodos BDF de orden de precisión mayor que 2 y con buenas características de estabilidad, principalmente los relacionados con la utilización de derivadas de segundo orden y de puntos super-futuros.

El Capítulo 3 está dedicado a la revisión y estado del arte de los métodos numéricos clásicos de la mecánica computacional para la resolución directa de EDOs de orden 2, sin la reducción previa a orden 1. En el marco de los métodos lineales multipaso para sistemas de EDOs de orden 2, y en paralelo con los métodos lineales para sistemas de orden 1, se deducen las características de precisión y de estabilidad de los métodos HHT- α y Newmark. Particularmente se presta atención al cálculo de los radios espectrales, amortiguamiento algorítmico y error relativo en el periodo, implementando los procedimientos computacionales que permiten representar estos parámetros en función de la frecuencia.

En el Capítulo 4 hemos estudiado la algorítmica de las funciones *ode45* y *ode15s* de Matlab. Se han analizado las medidas de error que utilizan ambas, es decir, la estimación del error que utilizan, de qué depende el tamaño de paso que utilizan y cuándo y cómo se produce el cambio de tamaño de paso. También hemos aplicado estas dos funciones a EDOs sencillas con el objeto de apreciar su eficiencia.

En el Capítulo 5 se exploran posibles mejoras de los métodos BDF encaminadas a ampliar las regiones de estabilidad con el fin de subsanar la ineficiencia computacional detectada en la aplicación de la *ode15s* en los problemas vibratorios. En particular, hemos desarrollado dos métodos numéricos para EDOs de orden 1 tomando como base dos métodos que utilizan puntos super-futuros: el método Extended Backward Differentiation Formulae (EBDF) [16] y el Modified Extended Backward Differentiation Formulae

(MEBDF) [18]. En ellos, cambiando el predictor BDF por NDF (Numerical Differentiation Formulae) [62] se ha conseguido mejorar la estabilidad de los métodos de origen, verificándose su buen comportamiento en varios problemas *de prueba*.

En el Capítulo 6 se exploran posibles mejoras de los métodos BDF orientadas a controlar el amortiguamiento algorítmico, como lo hacen los métodos HHT- α . El capítulo comienza con el estudio del radio espectral de algunos método multipaso para resolver EDOs de orden 1. Nos hemos percatado de la capacidad de las BDF para aniquilar las frecuencias altas y hemos propuesto un método nuevo al que hemos llamado BDF- α , que además de ser incondicionalmente estable y de orden de precisión 2 para ciertos valores del parámetro α , proporciona una gran variedad de opciones para anular o disminuir las frecuencias altas. Es decir, proporciona control paramétrico del amortiguamiento algorítmico.

El Capítulo 7 describe la implementación de un entorno para la resolución de EDOs utilizando las metodologías de la POO, extendiendo un trabajo previo, desde el MEF cuasi-estático al dinámico. Además, hemos aplicado los nuevos métodos que se han desarrollado durante este trabajo a EDPs con CI y CC.

Finalmente, en el Capítulo 8 se han recogido las conclusiones y las líneas futuras de investigación de este trabajo.

Capítulo 2

Métodos para EDOs de orden 1

Los descubrimientos matemáticos, pequeños o grandes [...], nunca se generan espontáneamente. Presuponen siempre una sólida base de conocimientos preliminares bien madurados por el trabajo consciente y subconsciente.

HENRI POINCARÉ

Dado un sistema de EDOs de orden 1 y de condición inicial:

$$y'(t) = f(t, y(t)), \quad y(a) = y_0 \quad (2.1)$$

siendo $f : [a, b] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ continua en su primera variable y satisfaciendo la condición de Lipschitz en la segunda variable, entonces, existe una única solución $y(t)$ para el problema (2.1) [14]. Buscamos una solución de (2.1) en el intervalo $a \leq t \leq b$, siendo a, b finitos. Los dos métodos numéricos más sencillos que permiten aproximar una solución de la misma son:

- El método explícito de Euler que viene dado por: $y_{n+1} = y_n + hf_n$.
- El método implícito o regresivo de Euler, cuya expresión es: $y_{n+1} = y_n + hf_{n+1}$

donde el parámetro h es el tamaño de paso. Ambos métodos son unipaso porque la solución numérica de la EDO se calcula utilizando el valor obtenido en el paso anterior.

Una variante de los métodos unipaso son los métodos unipaso multietapa, destacando entre ellos los métodos Runge-Kutta. Estos métodos utilizan la evaluación de la derivada en puntos intermedios del paso.

Otra variante de los métodos unipaso la constituyen los métodos multipaso que utilizan k aproximaciones anteriores $y_n, y_{n+1}, \dots, y_{n+k-1}$ de la solución exacta en los puntos $t_n, t_n + h, \dots, t_n + (k-1)h$ para calcular el valor y_{n+k} en el punto $t_{n+k} = t_n + kh$. Un método multipaso se puede escribir de esta manera:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\phi(f_{n+k}, f_{n+k-1}, \dots, f_n, t_n; h) \quad (2.2)$$

siendo $f_{n+j} = f(t_{n+j}, y_{n+j})$.

Si la relación entre los valores y_{n+j}, f_{n+j} para $j = 0, 1, \dots, k$ es lineal, el método multipaso es lineal y la expresión (2.2) se puede escribir como :

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j} \quad (2.3)$$

siendo α_j, β_j constantes, $\alpha_k \neq 0$ y α_0 y β_0 no nulas a la vez. El método (2.3) es explícito si $\beta_k = 0$ e implícito si $\beta_k \neq 0$. El método es unipaso en el caso en que $k = 1$.

Tres métodos multipaso lineales clásicos son: los métodos de Adams Bashforth, los métodos de Adams Moulton y los métodos BDF (Backward Differentiation Formulae).

Los métodos BDF fueron propuestos para ecuaciones diferenciales rígidas (stiff) y desde el libro de Gear (1971) [30], los códigos de ordenador basados en éstos han sido los más destacados y los más usados para todas las computaciones rígidas. No obstante, sus características de estabilidad empeoran al aumentar el orden. Así, en los últimos años se ha trabajado en la búsqueda de métodos multipaso con mejores condiciones de estabilidad, siguiendo sobre todo tres direcciones [37] que se explicarán más adelante:

- utilizando derivadas de orden superior (Sección 2.2.6.1),
- utilizando puntos que están fuera del intervalo de integración: puntos super-futuros (Sección 2.2.6.2),
- combinando dos métodos multipaso o combinando distintas técnicas para crear métodos multipaso (Sección 2.2.6.3).

Empezaremos este capítulo explicando algunos conceptos generales sobre los métodos numéricos para resolver EDOs de orden 1, tales como, el error, la estabilidad y el concepto de rigidez asociado. Después, nos centraremos en dos familias de métodos numéricos: los métodos multipaso lineales y los métodos unipaso multietapa. En particular haremos hincapié en los métodos BDF, haciendo un estado del arte que muestra métodos que servirán de base en desarrollos posteriores de esta tesis.

2.1. Error, estabilidad y rigidez

2.1.1. Error y orden de precisión

El error global de truncamiento de un método es la diferencia entre el valor exacto y el calculado mediante el método:

$$GTE_{n+k} = y(t_{n+k}) - y_{n+k} \quad (2.4)$$

Se conoce como asunción de localización [52] el hecho de considerar los valores calculados mediante el método iguales a los valores exactos en los pasos previos al actual paso en t_{n+k} :

$$y_{n+j} = y(t_{n+j}), \quad \text{para } j = 0, 1, \dots, k-1 \quad (2.5)$$

El valor que se calcula haciendo la asunción de localización lo denotaremos por y_{n+k}^* .

El error local de truncamiento de un método es la diferencia entre el valor exacto y el valor y_{n+k}^* :

$$LTE_{n+k} = y(t_{n+k}) - y_{n+k}^* \quad (2.6)$$

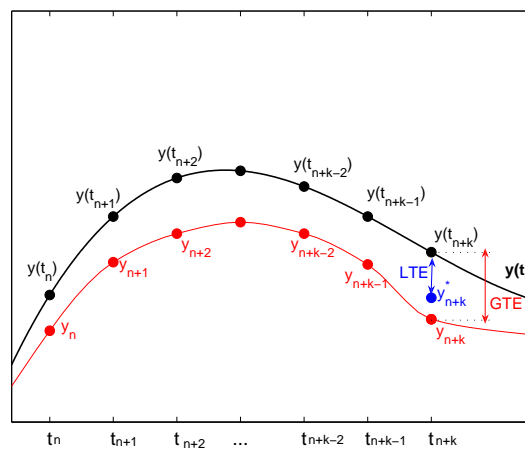


Figura 2.1: Ilustración del error local y global de truncamiento.

Las definiciones de error global de truncamiento y error local de truncamiento que hemos dado son válidas tanto para métodos unipaso (entre ellos para métodos multietapa) como para métodos multipaso.

Debido a la acumulación de error, el error global tiene una potencia menos en h que la que tiene el error local de truncamiento. Es decir: $LTE = O(h^{p+1}) \Rightarrow GTE = O(h^p)$ [52]. Un método numérico es de orden p si el error local de truncamiento es $O(h^{p+1})$.

2.1.2. Estabilidad

La estabilidad de una EDO (o sistema de EDOs) de orden 1 muestra la sensibilidad de la solución exacta ante perturbaciones. Igualmente, un método numérico es estable si no se amplifican las perturbaciones.

Para hacer el estudio de estabilidad lineal de los métodos numéricos se procede de la siguiente manera:

- Por simplicidad se asume que el Jacobiano $\partial f/\partial y$ del sistema de EDOs (2.1) es diagonalizable. Esto significa que existe una matriz T cuyas columnas son los autovectores del jacobiano, tal que:

$$T^{-1} \frac{\partial f}{\partial y} T = \text{diag} \{ \lambda_1, \lambda_2, \dots, \lambda_d \} \quad (2.7)$$

siendo λ_i los autovalores de $\partial f/\partial y$.

Haciendo un cambio de variable $y(t) = Tw(t)$, es posible desacoplar las ecuaciones de tal forma que la componente j -ésima satisfaga $w'_j = \lambda_j w_j$. Cada una de estas ecuaciones es un ejemplo de lo que se conoce como ecuación de test: una EDO escalar lineal homogénea y de coeficientes constantes $y' = \lambda y$.

- Tras aplicar el método numérico a la ecuación de test, se llega a una ecuación en diferencias, y por ejemplo, para un método multipaso (de k pasos) lineal (2.3) se obtiene la ecuación en diferencias de orden k con la forma:

$$\sum_{j=0}^k \alpha_j y_{n+j} = \hat{h} \sum_{j=0}^k \beta_j y_{n+j}, \quad \text{donde} \quad \hat{h} = \lambda h \quad (2.8)$$

La ecuación escalar de orden k (2.8) se puede escribir como un sistema de k ecuaciones en diferencias de orden 1:

$$\begin{pmatrix} y_{n+1} \\ y_{n+2} \\ \dots \\ y_{n+k} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & \dots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix} \cdot \begin{pmatrix} y_n \\ y_{n+1} \\ \dots \\ y_{n+k-1} \end{pmatrix} \quad (2.9)$$

siendo: $a_{ij} = a_{ij}(\hat{h})$. Escribiremos la expresión (2.9) como:

$$Y_{n+k} = A(\hat{h}) Y_{n+k-1} \quad (2.10)$$

donde $Y_{n+k} = (y_{n+1}, y_{n+2}, \dots, y_{n+k})^T$, $Y_{n+k-1} = (y_n, y_{n+1}, \dots, y_{n+k-1})^T$ y A una matriz de dimensión $k \times k$ a la que se le llama factor de amplificación.

En el caso de métodos unipaso ($k = 1$), A es una función escalar de \hat{h} , llamada función de estabilidad y la ecuación (2.10) se reduce a:

$$y_{n+1} = R(\hat{h}) y_n \quad (2.11)$$

- La estabilidad numérica requiere que el módulo de los autovalores de A sea igual o inferior a la unidad.

- Los autovalores de A son las raíces del polinomio característico o polinomio de estabilidad que se define como:

$$p(r) = \det(A - rI) \quad (2.12)$$

- El autovalor con mayor módulo de A recibe el nombre de radio espectral y se denota por ρ :

$$\rho = \max \{ |\rho_i| : \rho_i \text{ autovalor de } A \} \quad (2.13)$$

Definición. Un método es estable si el polinomio dado por (2.12) satisface la condición de raíz, es decir: las raíces de (2.12) caen dentro o en la frontera del círculo unidad del plano complejo.

Definición (Región de estabilidad). La región de estabilidad o región de estabilidad absoluta del método, es el conjunto S que verifica:

$$S = \left\{ \hat{h} \in \mathbb{C} : \left\{ \begin{array}{l} |r_j(\hat{h})| \leq 1 \quad \forall \hat{h}, r_j \text{ raíz simple de (2.12)} \\ |r_j(\hat{h})| < 1 \quad \forall \hat{h}, r_j \text{ raíz múltiple de (2.12)} \end{array} \right. \right\} \quad (2.14)$$

donde: $\hat{h} = h\lambda$.

La frontera de la región de estabilidad de un método numérico está formada por los valores \hat{h} para los cuales $r(\hat{h}) = 1$. Esta frontera se dibuja haciendo $r(\hat{h}) = e^{i\theta}$ y dando valores a $\theta \in [0, 2\pi)$. Para saber si la región de estabilidad es la parte interior o exterior de esa frontera debe comprobarse si se cumple $|r(\hat{h})| < 1$. Y éstos son los puntos que forman la región de estabilidad del método.

Definición (A-estabilidad o estabilidad absoluta). (Dahlquist 1963) Un método numérico se dice que es A-estable si $\mathbb{C}^- \in S$, siendo S la región de estabilidad del método. Ver Figura 2.2.

Teorema. (Dahlquist 1963) Un método multipaso A-estable debe ser de orden $p \leq 2$. El método trapezoidal es el método A-estable de orden 2 con menor constante de error siendo éste: $C = -\frac{1}{12}$. La demostración se puede encontrar en la referencia [37].□

Definición ($A(\alpha)$ -estabilidad). Sea S la región de estabilidad de un método. Éste es $A(\alpha)$ -estable con $0 < \alpha < \frac{\pi}{2}$ si:

$$C_\alpha = \left\{ \hat{h} : -\alpha < \pi - \arg(\hat{h}) < \alpha, \hat{h} \neq 0 \right\} \subseteq S \quad (2.15)$$

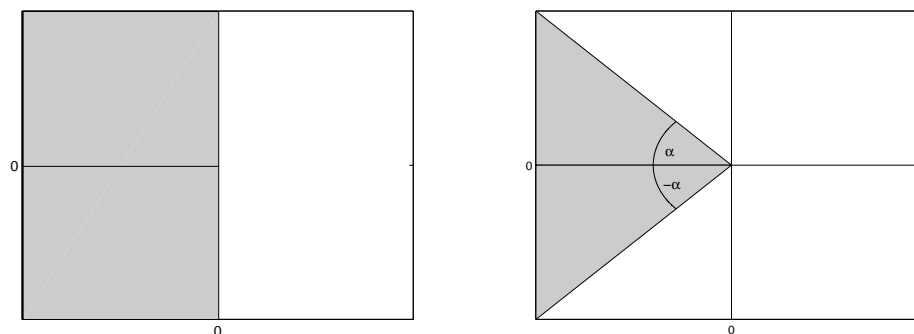


Figura 2.2: A-estabilidad y $A(\alpha)$ -estabilidad.

2.1.3. Rigidez

La rigidez es un concepto delicado, difícil e importante en la solución numérica de ecuaciones diferenciales ordinarias. Varios autores [50, 52, 67] coinciden en el hecho de que no existe una definición rigurosa de la rigidez. Depende de la ecuación diferencial, de las condiciones iniciales, del método numérico que se está utilizando para la resolución de la ecuación diferencial y del intervalo temporal en el que estamos hallando la solución. En este trabajo utilizaremos el concepto de rigidez definido en [40]: la rigidez surge cuando en la solución existen autovalores de distintas magnitudes. Esta diferencia de magnitud puede existir tanto en la parte real como en la imaginaria de los mismos.

La rigidez surge cuando algunas componentes de la solución decaen más rápidamente que otras [50], y precisan por razones de estabilidad un tamaño de paso h muy pequeño que afecta a la integración de todo el sistema.

En la referencia [51], también se define la rigidez de un problema no lineal. Así, se dice que el problema $y' = f(t, y)$ es rígido si los autovalores del Jacobiano $J = \partial f / \partial y$ se comportan de la misma manera que los autovalores de un sistema lineal. En el caso del sistema no lineal, los autovalores dependerán de la variable t .

La definición dada implica que el problema tiene varios autovalores, por tanto hay varias EDOs. Sin embargo, también una única EDO puede ser rígida. Por ejemplo, resulta rígida la EDO lineal no homogénea que se propone en [15]:

$$y' = -40y + 40t + 1, \quad y(0) = 4 \quad (2.16)$$

La solución exacta de (2.16) es $y = t + 4e^{-40t}$, que consta de dos partes: la parte de t que varía lentamente y la parte de e^{-40t} que varía más rápidamente. En este problema estamos obligados a utilizar tamaños de paso pequeños cuando estamos intentando aproximar una función que rápidamente se convierte en $y \approx t$. La rigidez de este problema es consecuencia de la diferencia de magnitud entre la solución homogénea y la solución particular.

En este trabajo tratamos con el sistema de EDOs fuertemente rígido obtenido tras la semidiscretización MEF. Por ejemplo, la rigidez de la EDP de difusión en un problema 1D crece al aumentar el número de elementos de la semidiscretización, al disminuir la longitud de la varilla o al aumentar la difusividad térmica. En el caso de la rigidez de la EDP onda 1D, también ésta crece al aumentar el número de elementos de la semidiscretización, al disminuir la longitud de la cuerda o al aumentar la velocidad de propagación de la onda, ver Apéndice B. En este Apéndice también mostramos la relación que existe entre los autovalores de la EDP de difusión y onda: los autovalores de la EDP onda son la raíz cuadrada de los autovalores de la EDP de difusión.

2.2. Métodos multipaso lineales

Recordamos en esta sección las características de precisión y estabilidad de los métodos Adams y BDF (Backward Differentiation Formulae), que se integran en la familia más amplia de los métodos multipaso lineales, cuya expresión tiene la forma (2.3):

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j}$$

- Para estudiar el orden de precisión de los métodos multipaso lineales, se define el siguiente operador lineal diferencial:

$$\mathcal{L}[y(t); h] = \sum_{j=0}^k [\alpha_j y(t + jh) - h \beta_j y'(t + jh)] \quad (2.17)$$

siendo $y(t)$ una función arbitraria y diferenciable cuantas veces queramos en el intervalo $[a, b]$. Pero si $y(t)$ es una solución de la EDO (2.1), el operador define el error local de truncamiento del método. Entonces, utilizando desarrollos de Taylor de las funciones $y(t)$ e $y'(t)$ alrededor del valor t_n se tiene [6, 51]:

$$\mathcal{L}[y(t); h] = C_0 y(t_n) + C_1 h y'(t_n) + C_2 h^2 y''(t_n) + \dots + C_q h^q y^{(q)}(t_n) + \dots \quad (2.18)$$

siendo las constantes C_i de la expresión (2.18):

$$\begin{cases} C_0 = \sum_{i=0}^k \alpha_i \\ C_1 = \sum_{i=0}^k i \alpha_i - \sum_{i=0}^k \beta_i \\ C_q = \frac{1}{q!} \left(\sum_{i=0}^k i^q \alpha_i \right) - \frac{1}{(q-1)!} \left(\sum_{i=0}^k i^{q-1} \beta_i \right), \quad q \geq 2 \end{cases} \quad (2.19)$$

La expresión (2.18) corresponde al error local de truncamiento del método numérico (2.3). Este método es de orden p si el error local de truncamiento es $O(h^{p+1})$, es

decir, si $C_0 = C_1 = \dots = C_p = 0$ y $C_{p+1} \neq 0$. Por lo que el error local de truncamiento queda:

$$LTE_{n+k} = C_{p+1} h^{p+1} y^{(p+1)}(t_n) + O(h^{p+2}) \quad (2.20)$$

Si se cumple $C_0 = C_1 = 0$, el método numérico mínimamente es de orden $p = 1$.

- Para el estudio de la estabilidad de los métodos multipaso lineales, aplicamos el método a la ecuación de test $y' = \lambda y$, y llegamos a la siguiente ecuación en diferencias de orden k :

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\lambda \sum_{j=0}^k \beta_j y_{n+j} \quad (2.21)$$

La ecuación (2.21) se puede resolver utilizando directamente el método de Lagrange [38, 56], sin pasar previamente a un sistema de k ecuaciones en diferencias de orden 1. Para ello es suficiente ensayar sucesiones del tipo $y_m = r^m$ para obtener el polinomio característico (2.12) en la forma:

$$p(r) = \sum_{j=0}^k \alpha_j r^j - \hat{h} \sum_{j=0}^k \beta_j r^j = \rho(r) - \hat{h}\sigma(r) \quad (2.22)$$

siendo: $\hat{h} = \lambda h$, y $\rho(r), \sigma(r)$ el primer y segundo polinomio característicos asociados al método multipaso lineal:

$$\begin{cases} \rho(r) = \sum_{j=0}^k \alpha_j r^j \\ \sigma(r) = \sum_{j=0}^k \beta_j r^j \end{cases} \quad (2.23)$$

Para dibujar la frontera de la región de estabilidad hay que hallar las raíces de (2.22) e imponer la condición de que su módulo sea la unidad, es decir $r = e^{i\theta}$, de donde resulta:

$$\hat{h} = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} \quad (2.24)$$

- Para obtener de la expresión (2.3) los métodos explícitos de Adams Bashforth hacemos $\alpha_k = 1, \alpha_{k-1} = -1, \alpha_j = 0$ para $j = 0, 1, \dots, k-2$, $\beta_k = 0, \beta_j = \tilde{\beta}_{k-j}$ para $j = 0, 1, \dots, k-1$, y entonces:

$$y_{n+k} = y_{n+k-1} + h \sum_{j=1}^k \tilde{\beta}_j f_{n+k-j}$$

Para obtener de la expresión (2.3) los métodos implícitos de Adams Moulton hacemos $\alpha_k = 1, \alpha_{k-1} = -1, \alpha_j = 0$ para $j = 0, 1, \dots, k-2$, $\beta_0 = 0, \beta_j = \beta_{k-j}^*$ para

$j = 1, 2, \dots, k$, y entonces:

$$y_{n+k} = y_{n+k-1} + h \sum_{j=1}^k \beta_j^* f_{n+k+1-j}$$

siendo $\tilde{\beta}_j$ y β_j^* constantes de los métodos de Adams Bashforth y Moulton respectivamente.

Los métodos de Adams constituyeron la primera generalización al método de Euler, y fueron ideados por John Couch Adams para solucionar un problema de F. Bashforth que ocurría en un problema de acción capilar [8]. Estos métodos utilizan más información que el método de Euler para poder avanzar un paso. En el libro de balística [54], se utilizan estos métodos numéricos para calcular la trayectoria de un proyectil.

- Para obtener de la expresión (2.3) los métodos BDF hacemos $\alpha_j = \hat{\alpha}_j$ para $j = 0, 1, \dots, k$, $\beta_k = 1$, $\beta_j = 0$ para $j = 0, 1, \dots, k-1$, entonces:

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h f_{n+k}$$

siendo $\hat{\alpha}_j$ las constantes de los métodos BDF. Las BDF fueron introducidas por Gear [30] y se han utilizado mucho para resolver problemas rígidos debido a las buenas características de estabilidad que presentan.

2.2.1. Interpolación polinómica. Diferencias regresivas

En las líneas anteriores hemos establecido las formas específicas de los métodos multipaso de Adams y BDF a falta de concretar las expresiones de los coeficientes. Para ello, la herramienta es la interpolación polinómica que exponemos a continuación.

La función $P(t)$ que interpola un conjunto de datos $\{(t_i, g_i) : i = 0, 1, 2, \dots, p-1\}$ se puede escribir como combinación lineal de una serie de funciones de base dadas por $\{\phi_j(t) : j = 0, 1, 2, \dots, p-1\}$ [12]:

$$P(t) = \sum_{j=0}^{p-1} x_j \phi_j(t) \quad (2.25)$$

siendo x_j constantes que hay que determinar.

Cuando los p valores $\{(t_i, g_i) : i = 0, 1, 2, \dots, p-1\}$ están separados una distancia igual h , es decir, $h = t_{i+1} - t_i$ para $i = 0, 1, 2, \dots, p-2$, las funciones $\phi_j(t)$ y las constantes x_j de (2.25) vienen dadas por:

$$\begin{cases} \phi_j(t) = \prod_{m=0}^{j-1} (t - t_{(p-1)-m}) \\ x_j = \nabla^j g_{p-1} \frac{1}{j! h^j} \end{cases} \quad (2.26)$$

Los términos $\nabla^j g_{p-1}$ son diferencias regresivas y se calculan teniendo en cuenta: $\nabla^j g_{p-1} = \nabla (\nabla^{j-1} g_{p-1})$ y $\nabla g_{p-1} = g_{p-1} - g_{p-2}$. También se cumple la siguiente igualdad:

$$\nabla^{j+1} g_i = \nabla^j g_i - \nabla^j g_{i-1}, \quad \forall i, j \quad (2.27)$$

El polinomio interpolador que se tiene cuando $\phi_j(t)$ y x_j vienen dadas por (2.26), es el polinomio interpolador regresivo de Newton:

$$P(t) = \sum_{j=0}^{p-1} x_j \phi_j(t) = \sum_{j=0}^{p-1} \nabla^j g_{p-1} \frac{1}{j! h^j} \prod_{m=0}^{j-1} (t - t_{p-1-m}) \quad (2.28)$$

El polinomio (2.28) puede adquirir distintas formas dependiendo de los puntos por los que pasa. Algunas de estas formas son:

- Cuando el polinomio $P(t)$ pasa por los $(k+1)$ puntos $\{(t_{n+i}, y_{n+i}) : i = 0, 1, 2, \dots, k\}$ tiene la siguiente expresión:

$$P(t) = y_{n+k} + \sum_{j=1}^k \nabla^j y_{n+k} \frac{1}{j! h^j} \prod_{m=0}^{j-1} (t - t_{n+k-m}) \quad (2.29)$$

- Cuando el polinomio $P(t)$ pasa por los k puntos $\{(t_{n+i}, f_{n+i}) : i = 1, 2, \dots, k\}$, siendo $f_{n+i} = f(t_{n+i}, y_{n+i})$, tiene la siguiente expresión:

$$P(t) = f_{n+k} + \sum_{j=1}^{k-1} \nabla^j f_{n+k} \frac{1}{j! h^j} \prod_{m=0}^{j-1} (t - t_{n+k-m}) \quad (2.30)$$

- Cuando el polinomio $P(t)$ pasa por los k puntos $\{(t_{n+i}, f_{n+i}) : i = 0, 1, 2, \dots, k-1\}$, siendo $f_{n+i} = f(t_{n+i}, y_{n+i})$, tiene la siguiente expresión:

$$P(t) = f_{n+k-1} + \sum_{j=1}^{k-1} \nabla^j f_{n+k-1} \frac{1}{j! h^j} \prod_{m=0}^{j-1} (t - t_{n+k-1-m}) \quad (2.31)$$

2.2.2. Métodos de Adams Bashforth

Suponiendo que se conocen las aproximaciones numéricas $y_n, y_{n+1}, \dots, y_{n+k-1}$ a las soluciones exactas $y(t_n), y(t_{n+1}), \dots, y(t_{n+k-1})$ de la ecuación diferencial dada por:

$$y' = f(t, y), \quad y(t_0) = y_0 \quad (2.32)$$

Adams considera la expresión (2.32) de la siguiente forma [38]:

$$y(t_{n+k}) = y(t_{n+k-1}) + \int_{t_{n+k-1}}^{t_{n+k}} f(t, y(t)) dt \quad (2.33)$$

Como las aproximaciones numéricas $y_n, y_{n+1}, \dots, y_{n+k-1}$ son conocidas, se pueden calcular los valores $f_{n+j} = f(t_{n+j}, y_{n+j})$ para $j = 0, 1, \dots, k-1$. Sustituimos la función $f(t, y(t))$ de (2.33) por un polinomio $P(t)$ que pasa por los puntos $\{(t_{n+j}, f_{n+j}) : j = 0, 1, \dots, k-1\}$ y que viene dado por (2.31):

$$y_{n+k} = y_{n+k-1} + \int_{t_{n+k-1}}^{t_{n+k}} P(t) dt \quad (2.34)$$

Tras integrar (2.34) se consigue el método de Adams Bashforth:

$$y_{n+k} = y_{n+k-1} + h \sum_{j=0}^{k-1} \gamma_j \nabla^j f_{n+k-1} \quad (2.35)$$

siendo los valores de γ_j los recogidos en la Tabla 2.1.

j	0	1	2	3	4	5	6	7	8
γ_j	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$	$\frac{5257}{17280}$	$\frac{1070017}{3628800}$

Tabla 2.1: Coeficientes del método de Adams Basforth [38].

Desarrollando las diferencias hacia atrás de (2.35) llegamos a esta otra forma equivalente de los métodos de Adams Bashforth:

$$y_{n+k} = y_{n+k-1} + h \sum_{j=1}^k \tilde{\beta}_j f_{n+k-j} \quad (2.36)$$

donde: $\tilde{\beta}_j = (-1)^{j-1} \sum_{i=j-1}^{k-1} \gamma_i \binom{i}{j-1}$.

Aplicando (2.24) obtenemos las fronteras de las regiones de estabilidad de los métodos de Adams Bashforth, ver Figura 2.3. Para $k = 1$ obtenemos el círculo del método de Euler centrado en el punto $(-1, 0)$ y de radio 1. Se puede observar que la región de estabilidad decrece cuando crece el orden. Estos métodos no son adecuados para los problemas rígidos porque su región de estabilidad es *pequeña*.

2.2.3. Métodos de Adams Moulton

En los métodos de Adams Moulton, el polinomio del método de Adams Bashforth se reemplaza por otro que pasa por los puntos $\{(t_{n+j}, f_{n+j}) : j = 1, 2, \dots, k\}$ y que viene dado por (2.30). Sustituyendo este polinomio en la expresión (2.33) e integrando se llega al siguiente método implícito:

$$y_{n+k} = y_{n+k-1} + h \sum_{j=0}^{k-1} \gamma_j^* \nabla^j f_{n+k} \quad (2.37)$$

siendo los valores de γ_j^* los recogidos en la Tabla 2.2. Desarrollando las diferencias hacia atrás de (2.37), se llega a esta forma de los métodos de Adams Moulton:

$$y_{n+k} = y_{n+k-1} + h \sum_{j=1}^k \beta_j^* f_{n+k+1-j} \quad (2.38)$$

siendo: $\beta_j^* = (-1)^j \sum_{i=j}^{k-1} \gamma_i^*(j)$

j	0	1	2	3	4	5	6	7	8
γ_j^*	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$	$-\frac{275}{24192}$	$-\frac{33953}{3628800}$

Tabla 2.2: Coeficientes del método de Adams Moulton [38].

El método de Adams Moulton para $k = 1$ es el método de Euler regresivo de orden 1 y para $k = 2$ se obtiene el método trapezoidal (de orden 2).

De nuevo aplicando (2.24), se consiguen las fronteras de las regiones de estabilidad de Adams Moulton, Figura 2.3. Las regiones de estabilidad del método Adams Moulton son más grandes que las regiones de estabilidad del método explícito Adams Bashforth, pero no cubren toda la parte negativa del plano complejo. Al tratarse de métodos que no son A-estables, no son los más adecuados para resolver EDOs rígidas.

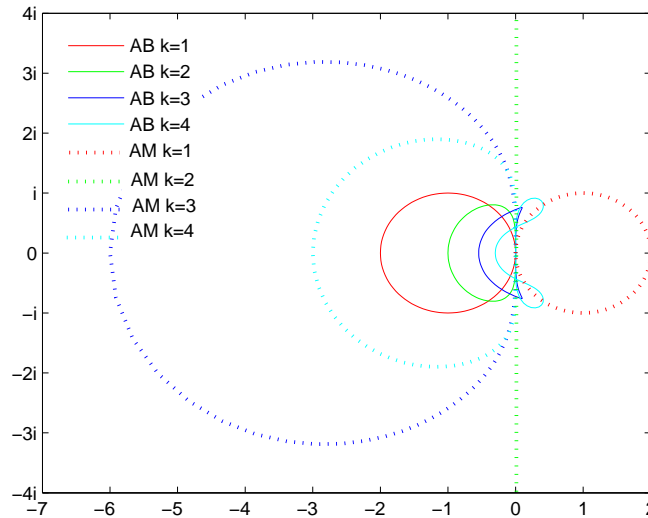


Figura 2.3: Regiones de estabilidad de Adams Bashforth y Moulton (interior a curva).

2.2.4. Métodos BDF

Las BDF son métodos multipaso lineales que utilizan el valor de la derivada en un único punto, el resto son valores de y . Para llegar a la expresión del método BDF, se tienen las aproximaciones numéricas $y_n, y_{n+1}, \dots, y_{n+k-1}$ a la solución exacta del problema $y' = f(t, y(t))$, $y(t_0) = y_0$. Se considera el polinomio $P(t)$ que interpola los valores $\{(t_{n+j}, y_{n+j}) : j = 0, \dots, k\}$ y que viene dado por (2.29). La incógnita y_{n+k} se calcula de forma que el polinomio $P(t)$ satisfaga la ecuación diferencial en el último punto de la malla, es decir:

$$P'(t_{n+k}) = f(t_{n+k}, y_{n+k}) \quad (2.39)$$

De esta manera llegamos a la expresión de los métodos BDF:

$$\sum_{j=1}^k \frac{1}{j} \nabla^j y_{n+k} = h f_{n+k} \quad (2.40)$$

Desarrollando las diferencias hacia atrás de (2.40), se obtiene otra expresión equivalente de los métodos BDF:

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h f_{n+k} \quad (2.41)$$

k	$\hat{\alpha}_0$	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$	$\hat{\alpha}_6$	$\hat{\alpha}_7$	C	C_{k+1}
1	-1	1							$-\frac{1}{2}$	$-\frac{1}{2}$
2	$\frac{1}{2}$	-2	$\frac{3}{2}$						$-\frac{1}{3}$	$-\frac{2}{9}$
3	$-\frac{1}{3}$	$\frac{3}{2}$	-3	$\frac{11}{6}$					$-\frac{1}{4}$	$-\frac{3}{22}$
4	$\frac{1}{4}$	$-\frac{4}{3}$	3	-4	$\frac{25}{12}$				$-\frac{1}{5}$	$-\frac{12}{125}$
5	$-\frac{1}{5}$	$\frac{5}{4}$	$-\frac{10}{3}$	5	-5	$\frac{137}{60}$			$-\frac{1}{6}$	$-\frac{10}{137}$
6	$\frac{1}{6}$	$-\frac{6}{5}$	$\frac{15}{4}$	$-\frac{20}{3}$	$\frac{15}{2}$	-6	$\frac{147}{60}$		$-\frac{1}{7}$	$-\frac{20}{343}$

Tabla 2.3: Coeficientes y constantes de error de los métodos BDF.

Y siguiendo el procedimiento descrito en (2.24), dibujamos las fronteras de la regiones de estabilidad, Figura 2.4. En la Tabla 2.4 se recogen los ángulos de $A(\alpha)$ -estabilidad. Sólo los métodos BDF de órdenes 1 y 2 son A -estables, pero las regiones de estabilidad de estos métodos son más amplias que las de los métodos de Adams [7].

El error de truncamiento del método BDF de orden k viene dado por [52]:

$$LTE_{n+k} = C_{k+1} h^{k+1} y^{(k+1)}(t_n) + O(h^{k+2}) \quad (2.42)$$

k	1	2	3	4	5	6
$A(\alpha)$	90°	90°	86.03°	73.35°	51.84°	17.84°

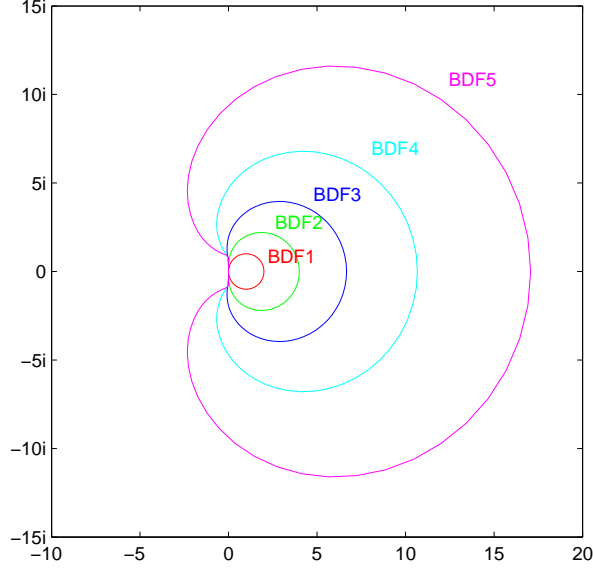
Tabla 2.4: $A(\alpha)$ -estabilidad de los métodos BDF.

Figura 2.4: Regiones de estabilidad de los métodos BDF (exterior a curva).

donde:

$$C_{k+1} = \frac{-1}{\gamma_k(k+1)}, \quad \gamma_k = \hat{\alpha}_k = \sum_{j=1}^k \frac{1}{j} \quad (2.43)$$

En referencias como [13, 38] se utiliza la constante de error C_{k+1} de la siguiente manera:

$$C_{k+1}^* = \frac{C_{k+1}}{\rho'(1)} = \frac{C_{k+1}}{\sigma(1)} \quad (2.44)$$

siendo $\rho(r)$ y $\sigma(r)$ los polinomios característicos asociados al método dados por (2.23). Es inmediato comprobar que $\rho'(1) = \sigma(1)$. Por una parte tenemos $\rho'(1) = \sum_{j=0}^k j\alpha_j$ y por otra, $\sigma(1) = \sum_{j=0}^k \beta_j$. Y utilizando (2.19) la diferencia de ambas resulta:

$$\rho'(1) - \sigma(1) = \sum_{j=0}^k j\alpha_j - \sum_{j=0}^k \beta_j = C_1 \quad (2.45)$$

que si el orden del método es $p \geq 1$, se cumple $C_1 = 0$. Por lo que se verifica la igualdad:

$$\rho'(1) = \sigma(1) \quad (2.46)$$

Sustituyendo (2.43) en (2.44), obtenemos las constantes de error para las BDF:

$$C_{k+1}^* = \frac{-1}{k+1} \quad (2.47)$$

Se explica en el siguiente ejemplo el motivo por el que se utilizan dos constantes distintas. Por ejemplo, la *odesuite* de Matlab utiliza la constante de error (2.47) y es ésta la que también utilizaremos en este trabajo, denotando por C a C_{k+1}^* .

Ejemplo del cálculo del error local de truncamiento: Hemos calculado el error local de truncamiento del método BDF de orden 2 que viene dado por:

$$y_{n+2} = \frac{4}{3}y_{n+1} - \frac{1}{3}y_n + \frac{2}{3}hy'_{n+2} \quad (2.48)$$

Se trata de un método multipaso que tiene la forma (2.3) siendo: $\alpha_0 = \frac{1}{3}$, $\alpha_1 = -\frac{4}{3}$, $\alpha_2 = 1$, $\beta_2 = \frac{2}{3}$. Los polinomios característicos del método BDF2 son:

$$\begin{cases} \rho(r) = \sum_{j=0}^k \alpha_j r^j = \frac{1}{3} - \frac{4}{3}r + r^2 \\ \sigma(r) = \sum_{j=0}^k \beta_j r^j = \frac{2}{3}r^2 \end{cases} \quad (2.49)$$

Por la definición (2.6) del error local de truncamiento tenemos:

$$LTE = y(t_{n+2}) - y_{n+2}^* \quad (2.50)$$

donde y_{n+2}^* se obtiene haciendo la asunción de localización (2.5). El problema de los métodos implícitos es establecer el valor de la derivada y'_{n+k} (y'_{n+2} de este ejemplo). Una opción posible es asignar a y'_{n+2} el valor exacto $y'(t_{n+2})$ y desarrollarlo en serie de Taylor junto a los demás términos en torno a t_n , como se indica a continuación:

$$y_{n+2}^* = \frac{4}{3}y(t_{n+1}) - \frac{1}{3}y(t_n) + \frac{2}{3}hy'(t_{n+2}) \quad (2.51)$$

Utilizaremos los siguientes desarrollos de Taylor centrados en el punto t_n :

$$\begin{cases} y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{1}{2!}h^2y''(t_n) + \frac{1}{3!}h^3y'''(t_n) + \dots \\ y(t_{n+2}) = y(t_n) + 2hy'(t_n) + 2h^2y''(t_n) + \frac{4}{3}h^3y'''(t_n) + \dots \\ y'(t_{n+2}) = y'(t_n) + 2hy''(t_n) + 2h^2y'''(t_n) + \dots \end{cases} \quad (2.52)$$

Sustituyendo los desarrollos (2.52) en (2.50) y reordenando términos se tiene:

$$\begin{aligned} LTE = y(t_n) & \left(1 - \frac{4}{3} + \frac{1}{3}\right) + hy'(t_n) \left(2 - \frac{4}{3} - \frac{2}{3}\right) \\ & + h^2y''(t_n) \left(2 - \frac{2}{3} - \frac{4}{3}\right) + h^3y'''(t_n) \left(\frac{4}{3} - \frac{4}{18} - \frac{4}{3}\right) + O(h^4) \end{aligned} \quad (2.53)$$

Por lo que el error local de truncamiento del método BDF2 es:

$$LTE = C_3 h^3 y'''(t_n) + O(h^4) \quad (2.54)$$

siendo $C_3 = -\frac{2}{9}$

La segunda opción, probablemente más coherente, consiste en asignar a la derivada y'_{n+2} el valor $(y_{n+2}^*)'$ y resolver la ecuación no lineal:

$$y_{n+2}^* = \frac{4}{3}y(t_{n+1}) - \frac{1}{3}y(t_n) + \frac{2}{3}h(y_{n+2}^*)' \quad (2.55)$$

Entonces, la constante de error toma la forma:

$$C_3^* = \frac{C_3}{\rho'(1)} = \frac{C_3}{\sigma(1)} \quad (2.56)$$

donde C_3 viene dada por (2.43). En la práctica esta segunda opción (2.55) es más difícil de desarrollar, por lo que se recurre a la primera (asignando $y'(t_{n+2})$ a y'_{n+2}) para determinar C_3 y calcular posteriormente C_3^* utilizando (2.56).

También se llega al valor de C_3 sustituyendo los valores α_i, β_i correspondientes al método BDF2 en las expresiones dadas por (2.19), obteniéndose $C_0 = C_1 = C_2 = 0$, $C_3 = -2/9$. Queda así demostrado que el método es de orden 2. Para calcular la constante de error C_3^* necesitamos calcular $\rho'(1)$ o $\sigma(1)$ (ambos son iguales (2.46)):

$$\begin{cases} \rho'(r) = -\frac{4}{3} + 2r \Rightarrow \rho'(1) = \frac{2}{3} \\ \sigma(1) = \frac{2}{3} \end{cases} \quad (2.57)$$

Y finalmente llegamos a la constante de error C_3^* del método BDF2:

$$C_3^* = \frac{C_3}{\rho'(1)} = \frac{C_3}{\sigma(1)} = \frac{-2/9}{2/3} = \frac{-1}{3} \quad (2.58)$$

Cabe decir que en los métodos de Adams Moulton no se suscita esta problemática, ya que $\rho'(1) = \sigma(1) = 1$, y ambas versiones de la constante de error coinciden.

2.2.5. Métodos NDF

Una de las modificaciones hechas a los métodos BDF son los métodos NDF (Numerical Differentiation Formulae) [62]. Es una modificación barata computacionalmente, que consiste en anticipar una diferencia de orden $(k+1)$ cuando estamos trabajando en orden k . La NDF de orden k viene dada por la siguiente expresión:

$$\sum_{j=1}^k \frac{1}{j} \nabla^j y_{n+k} = hf_{n+k} + \kappa \gamma_k \nabla^{k+1} y_{n+k} \quad (2.59)$$

donde la constante γ_k está dada por (2.43). El coeficiente κ fue introducido por Klopfenstein y Shampine para maximizar el ángulo de $A(\alpha)$ -estabilidad al mismo tiempo que se reducía el error local de truncamiento. Las NDF son menos estables que las BDF pero más precisas, ya que su constante de error es menor que el de las BDF:

$$C = \left(-\frac{1}{k+1} - \kappa\gamma_k \right) \quad (2.60)$$

Shampine sólo propone añadir este sumando en órdenes $k = 1, 2, 3, 4$ ya que a partir de orden 5 no se gana demasiado en eficacia. Las características de estos métodos se recogen en la Tabla 2.5.

k	κ de las NDF	%tamaño de paso	$A(\alpha)$ de las BDF	$A(\alpha)$ de las NDF
1	-0.1850	26 %	90°	90°
2	-1/9	26 %	90°	90°
3	-0.0823	26 %	86°	80°
4	-0.0415	12 %	73°	66°

Tabla 2.5: NDFs de Klopfenstein y Shampine: eficiencia y estabilidad respecto BDF.

2.2.6. Métodos multipaso lineales mejorados

Como ya hemos adelantado en la introducción de este capítulo, las tres líneas generales que se han seguido en los últimos años para construir métodos multipaso lineales con buenas características de estabilidad han consistido en utilizar derivadas de orden superior, puntos super-futuros y en combinar dos métodos multipaso [37].

2.2.6.1. Métodos multipaso lineales que utilizan la 2ª derivada

Los métodos de Enright [24] y los métodos SDBDF (Second derivative BDF method) [37] son métodos que utilizan la segunda derivada. Ambos métodos son A-estables hasta orden $p = 4$. Las características de estabilidad de los métodos SDBDF son mejores que las de los métodos de Enright, pero las constantes de error C de estas últimas son menores, ver Tablas 2.6 y 2.7. Se pueden ver sus regiones de estabilidad en las Figuras 2.5 y 2.6.

En [49] se presenta el método *New Efficient Second Derivative Multistep Methods* que utiliza la segunda derivada y que depende de dos parámetros β^* y γ^* . Es un método que presenta mejores características de estabilidad que los métodos SDBDF para ciertos valores de los parámetros, ver Tablas 2.8 y 2.9.

k	1	2	3	4	5	6	7
p	3	4	5	6	7	8	9
$A(\alpha)$	90°	90°	87.88°	82.03°	73.10°	59.95°	37.61°
C	0.01389	0.00486	0.00236	0.00136	0.00086	0.00059	0.00042

Tabla 2.6: Características de los métodos de Enright.

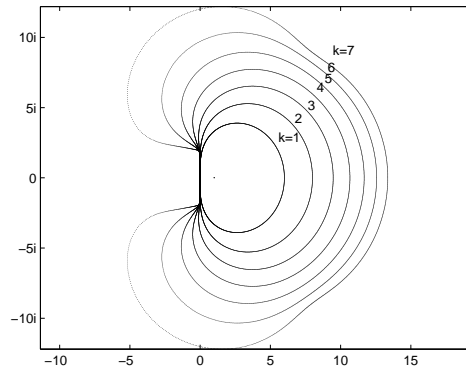


Figura 2.5: Regiones de estabilidad de los métodos de Enright (exterior a curva).

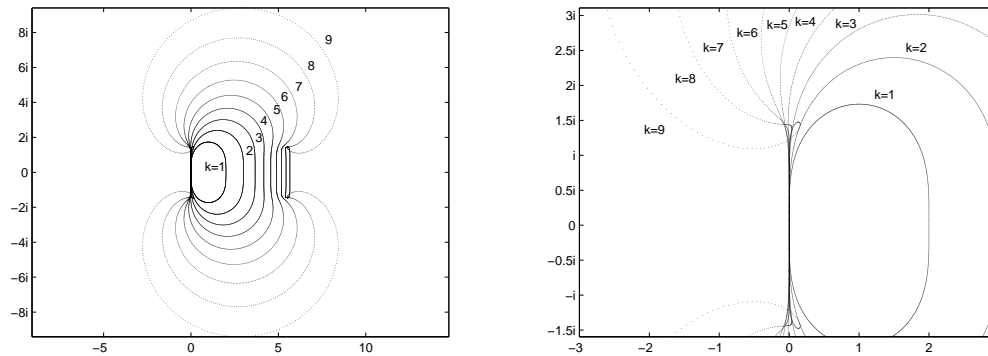


Figura 2.6: Regiones de estabilidad de los métodos SDBDF (exterior a curva).

k	1	2	3	4	5	6	7	8	9	10
p	2	3	4	5	6	7	8	9	10	11
$A(\alpha)$	90°	90°	90°	89.36°	86.35°	80.82°	72.53°	60.71°	43.39°	12.34°
C	0.1667	0.0556	0.0273	0.0160	0.0104	0.0073	0.0054	0.0041	0.0032	0.0026

Tabla 2.7: Características de los métodos SDBDF.

k	1	2	3	4	5	6	7	8	9
p	2	3	4	5	6	7	8	9	10
$A(\alpha)$	90°	90°	90°	89°	86°	81.7°	75°	63.5°	47.6°

Tabla 2.8: Características del método New Efficient SDMM con $\beta^* = -0,2, \gamma^* = 0,2$.

k	1	2	3	4	5	6	7	8	9
p	2	3	4	5	6	7	8	9	10
$A(\alpha)$	90°	90°	90°	89.9°	87.3°	84.2°	80°	71°	57.8°

Tabla 2.9: Características del método New Efficient SDMM con $\beta^* = -0,05, \gamma^* = 0,9$.

2.2.6.2. Métodos multipaso extendidos

Con el propósito de aumentar la estabilidad de los métodos BDF, Cash extendió estos métodos añadiendo puntos super-futuros [16]. Al método se le llamó EBDF (Extended Backward Differentiation Formula) y consiste en la aplicación sucesiva de dos predictores BDF en instantes t_{n+k} y t_{n+k+1} y un corrector que viene dado por:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\beta_k f_{n+k} + h\beta_{k+1} \bar{f}_{n+k+1} \quad (2.61)$$

\bar{f}_{n+k+1} es un nuevo término que se calcula en el instante super-futuro t_{n+k+1} aplicando sucesivamente la BDF estándar. Los métodos EBDF son A-estables hasta orden $p = 4$, ver Tabla 2.10.

k	1	2	3	4	5	6	7	8
p	2	3	4	5	6	7	8	9
$A(\alpha)$	90°	90°	90°	87.61°	80.21°	67.73°	48.82°	19.98°

Tabla 2.10: Características de los métodos EBDF.

En [18] y [19] se introduce el método MEBDF (Modified Extended Backward Differentiation Formulae) que tiene mejores características de estabilidad que el método EBDF (ver Tabla 2.11). Este nuevo método utiliza los mismos predictores que el EBDF y un nuevo corrector:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\hat{\beta}_k f_{n+k} + h\beta_{k+1} \bar{f}_{n+k+1} + h(\beta_k - \hat{\beta}_k) \bar{f}_{n+k} \quad (2.62)$$

En [20] se propone un código para las MEBDF y en [46] se presentan los métodos llamados MF-MEBDF que basados en [11] consiguen optimizar la computación de los

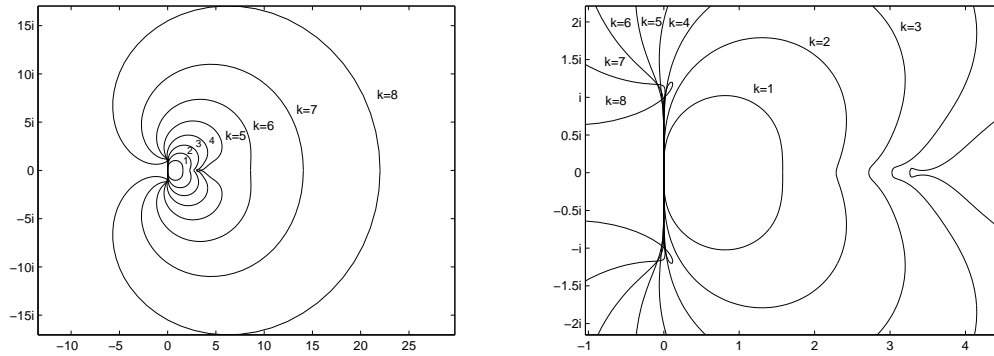


Figura 2.7: Regiones de estabilidad de los métodos EBDF (exterior a curva).

k	1	2	3	4	5	6	7	8
p	2	3	4	5	6	7	8	9
$A(\alpha)$	90°	90°	90°	88.36°	83.14°	74.50°	62°	42.88°

Tabla 2.11: Características de los métodos MEBDF.

métodos MEBDF. En [57] se proponen dos nuevos métodos multipaso extendidos para $k = 1, 2, 3, 4$, llamados bloque avanzado implícito. Ambos métodos siguen el esquema de los métodos EBDF y MEBDF: predicen dos veces utilizando BDFs de orden k y corrigen con unos nuevos correctores. Uno de estos correctores (el primero) mejora las características de estabilidad de los métodos EBDF y MEBDF en $k = 4$ (ver Tabla 2.12).

k	1	2	3	4
p	2	3	4	5
$A(\alpha)$ Corrector 1	90°	90°	90°	90°
$A(\alpha)$ Corrector 2	90°	89.5°	72°	39°

Tabla 2.12: Características del bloque avanzado implícito.

2.2.6.3. Combinación de métodos multipaso

También se puede crear un método realizando combinaciones lineales de métodos multipaso existentes. Los métodos combinados (*blended*) [66] y los métodos A-BDF [28] son métodos de este tipo.

Los métodos combinados [66] son una combinación lineal del método de Adams Moulton de orden $(k + 1)$ y del método BDF de orden k :

$$\left\{ AMF^{(k+1)} - \gamma^k \cdot h \cdot J \cdot BDF^{(k)} \right\} = 0 \quad (2.63)$$

siendo γ^k un parámetro libre y $J = \partial f / \partial y$. La fórmula (2.63) es de orden $p = (k + 1)$ para $\forall \gamma^k$. Eligiendo $\gamma^{(k)} = -k\gamma_k^*$, siendo γ_k^* los valores de la Tabla 2.2 correspondientes a Adams Moulton, es posible conseguir A-estabilidad hasta orden $p = 4$. Si elegimos $\gamma^{(k)}$ de tal forma que maximice el ángulo de A-estabilidad se consigue A-estabilidad hasta orden $p = 4$ y $A(28,68^\circ)$ -estabilidad en orden $p = 12$ [37, 66]. En la Figura 2.8 se muestran las regiones de estabilidad de los métodos combinados para $\gamma^{(k)} = -k\gamma_k^*$, siendo γ_k^* los valores correspondientes al método de Adams Moulton.

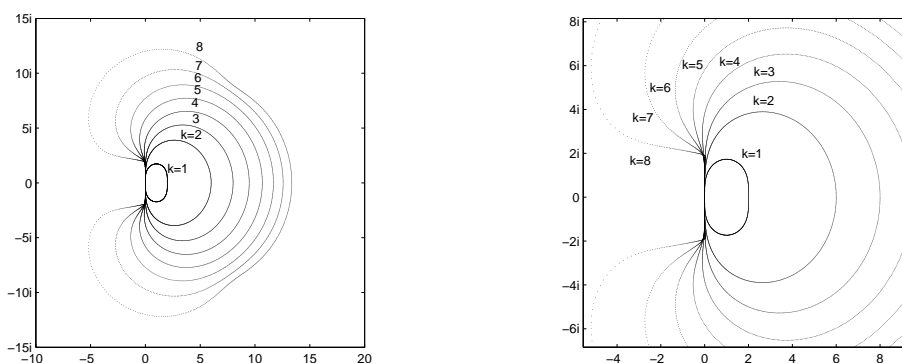


Figura 2.8: Regiones de estabilidad de los métodos combinados (exterior a curva).

También los métodos A-BDF [28] son combinación lineal de dos métodos multipaso existentes: del método BDF implícito (BDF de Gear) y del BDF explícito:

$$BDF^{(i)} - a \cdot BDF^{(e)} = 0 \quad (2.64)$$

siendo a un parámetro. Estos métodos son A-estables para $k = 1, 2$ sólo si $a \in [-1, 1)$, y $A(\alpha)$ -estables hasta orden 7.

Otra opción es combinar las técnicas de dos métodos existentes. En [44] se utiliza el método A-BDF como predictor dentro del esquema EBDF, creando el método A-EBDF. Ajustando el parámetro a del método A-BDF se consigue que los métodos A-EBDF sean A-estables para $p \leq 4$ y $A(\alpha)$ -estables hasta orden 9 siendo el ángulo de estabilidad mayor que el de los métodos BDF, A-BDF y EBDF, ver Tabla 2.13.

Cuando se utilizan segundas derivadas dentro del esquema de los métodos extendidos, es posible crear métodos A-estables hasta orden $p = 6$, o incluso hasta orden $p = 8$. Es el caso de los métodos E2BD [17] en los que se presentan dos clases de métodos y los métodos New SDMM [45], ver Tablas 2.14 y 2.15.

k	1	2	3	4	5	6	7	8
p	2	3	4	5	6	7	8	9
$A(\alpha)$	90°	90°	90°	88.85°	84.2°	75°	61°	30.50°

Tabla 2.13: Características de los métodos A-EBDF.

k	1	2	3	4	5	6
p	4	5	6	7	8	9
$A(\alpha)$ clase 2	90°	90°	90°	89°	87°	83°
$A(\alpha)$ clase 1	90°	90°	90°	90°	90°	89°

Tabla 2.14: Características de los métodos E2BD.

k	1	2	3	4	5	6	7	8	9	10	11	12
p	3	4	5	6	7	8	9	10	11	12	13	14
$A(\alpha)$	90°	90°	90°	90°	89.8°	88.3°	85.3°	80.5°	73.5°	61.9°	50.3°	29.9°

Tabla 2.15: Características de los métodos New SDMM.

2.3. Métodos Runge-Kutta

Los métodos Runge-Kutta son métodos que en cada paso utilizan la evaluación de la derivada en puntos intermedios del paso. Se les llama métodos multietapa o de s-etapas. Esta idea generalmente se le atribuye a Runge (1895), pero también Heun (1900) y Kutta (1901) hicieron sus contribuciones. En concreto, fue Kutta quien caracterizó completamente el conjunto de métodos Runge-Kutta de orden 4 y propuso los primeros métodos de orden 5. No fue hasta el trabajo de Huta (1956) cuando se introdujeron métodos de orden 6. Desde que se introdujeron los ordenadores digitales, existe más interés en los métodos Runge-Kutta y muchos investigadores han contribuido en extensiones de la teoría y en el desarrollo de métodos particulares. Este es el caso de [58], en el que se construye un método Runge-Kutta de orden 5. Aunque al principio todos los estudios se centrasen en los métodos Runge-Kutta explícitos, hoy en día también interesa el estudio de los métodos Runge-Kutta implícitos, ya que éstos son apropiados para la resolución de ecuaciones diferenciales rígidas.

2.3.1. Introducción al método

Dada una EDO de valor inicial, el método general de s -etapas de Runge-Kutta viene dado por:

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i \quad (2.65)$$

$$\text{siendo: } k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^s a_{ij} k_j), \quad i = 1, 2, \dots, s \quad (2.66)$$

Las constantes a_{ij}, b_i, c_i de las fórmulas de avance (2.65) y (2.66) de un método Runge-Kutta se recogen en una tabla de coeficientes que se conoce como Tabla Butcher.

c_1	a_{11}	a_{12}	\cdots	a_{1s}
c_2	a_{21}	a_{22}	\cdots	a_{2s}
\cdots	\cdots	\cdots	\cdots	\cdots
c_s	a_{s1}	a_{s2}	\cdots	a_{ss}
	b_1	b_2	\cdots	b_s

Tabla 2.16: Tabla Butcher de un método Runge-Kutta.

La Tabla Butcher 2.16 también se puede expresar matricialmente, Tabla 2.17, definiendo los vectores \mathbf{b} y \mathbf{c} de dimensión s y la matriz A de dimensión $s \times s$ como:

$$\mathbf{b} = [b_1, b_2, \dots, b_s]^T, \quad \mathbf{c} = [c_1, c_2, \dots, c_s]^T, \quad A = [a_{ij}]$$

$$\frac{\mathbf{c}}{\mathbf{b}^T} \left| \begin{array}{c} A \end{array} \right.$$

Tabla 2.17: Representación matricial de la Tabla Butcher.

Cada componente del vector \mathbf{c} indica la etapa dentro del paso y el vector \mathbf{b}^T es un vector de pesos. La matriz A indica la dependencia de las etapas respecto a las derivadas halladas en etapas anteriores (método explícito) o en todas las etapas (método implícito). Un método Runge-Kutta puede ser explícito (A estrictamente triangular inferior), implícito (A no es triangular inferior) o semi-implícito (A triangular inferior). En el caso explícito la fórmula (2.66) toma la siguiente forma:

$$k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j), \quad i = 1, 2, \dots, s \quad (2.67)$$

Aparte de las condiciones de orden que veremos, se establece en general la siguiente relación entre las constantes a_{ij}, c_i [13]:

$$c_i = \sum_{j=1}^s a_{ij}, \quad i = 1, 2, \dots, s \quad (2.68)$$

2.3.2. Condiciones de orden

La expresión del error local de truncamiento de los métodos Runge-Kutta es un poco compleja. Empezaremos viendo cuáles son las condiciones de orden de un método Runge-Kutta explícito de 3-etapas y de orden 3. En el caso de los métodos unipaso, la expresión del error local de truncamiento (2.6) queda:

$$LTE_{n+1} = y(t_{n+1}) - y_{n+1}^*$$

siendo y_{n+1}^* calculado mediante la asunción de localización (2.5). Consideramos el desarrollo en serie de Taylor de $y(t_{n+1})$:

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \frac{h^3}{6}y'''(t_n) + O(h^4) \quad (2.69)$$

Supondremos que la función $f(t, y)$ es lo suficientemente derivable como para que existan y sean continuas sus derivadas parciales. Utilizamos las notaciones abreviadas para la función $f(t, y)$ y para sus derivadas parciales:

$$f = f(t, y), \quad f_t = \frac{\partial f}{\partial t}, \quad f_y = \frac{\partial f}{\partial y}, \quad f_{tt} = \frac{\partial^2 f}{\partial t^2}, \quad f_{yy} = \frac{\partial^2 f}{\partial y^2}, \quad f_{ty} = f_{yt} = \frac{\partial^2 f}{\partial y \partial t}, \quad \text{etc.} \quad (2.70)$$

Escribimos las derivadas de (2.69) en función de las notaciones abreviadas (2.70):

$$\begin{cases} y'(t_n) = f \\ y''(t_n) = f_t + f_y y' = f_t + f f_y \\ y'''(t_n) = f_{tt} + f_{ty} f + f(f_{yt} + f f_{yy}) + f_y(f_t + f f_y) = \\ \quad = f_{tt} + 2f_{ty} f + f^2 f_{yy} + f_y(f_t + f f_y) \end{cases} \quad (2.71)$$

Introducimos las funciones $F = f_t + f f_y$ y $G = f_{tt} + 2f_{ty} f + f^2 f_{yy}$ para abreviar aún más la notación utilizada en (2.71). Así, el desarrollo de Taylor (2.69) queda:

$$y(t_{n+1}) = y(t_n) + hf + \frac{1}{2}h^2 F + \frac{1}{6}h^3 (F f_y + G) + O(h^4) \quad (2.72)$$

Para el caso de un método Runge-Kutta explícito de 3-etapas, el valor y_{n+1}^* viene dado por:

$$y_{n+1}^* = y(t_n) + h \sum_{i=1}^3 b_i k_i \quad (2.73)$$

siendo: $k_i = f(t_n + c_i h, y(t_n) + h \sum_{j=1}^{i-1} a_{ij} k_j)$ para $i = 1, 2, 3$. Desarrollando los términos k_i y utilizando las expresiones de F y G se llega a:

$$\begin{cases} k_1 = f \\ k_2 = f + hc_2 F + \frac{1}{2} h^2 c_2^2 G + O(h^3) \\ k_3 = f + hc_3 F + h^2 \left(c_2 a_{32} F f_y + \frac{1}{2} c_3^2 G \right) + O(h^3) \end{cases} \quad (2.74)$$

Y sustituyendo (2.74) en (2.73) se tiene:

$$\begin{aligned} y_{n+1}^* = y(t_n) &+ hf(b_1 + b_2 + b_3) + h^2 F(b_2 c_2 + b_3 c_3) \\ &+ \frac{1}{2} h^3 [2b_3 c_2 a_{32} F f_y + (b_2 c_2^2 + b_3 c_3^2) G] + O(h^4) \end{aligned} \quad (2.75)$$

Finalmente, sustituimos las expresiones (2.72) y (2.75) en la expresión del error local de truncamiento y éste queda:

$$\begin{aligned} LTE_{n+1} &= hf(1 - (b_1 + b_2 + b_3)) + h^2 F \left(\frac{1}{2} - (b_2 c_2 + b_3 c_3) \right) \\ &+ h^3 \left(F f_y \left(\frac{1}{6} - b_3 c_2 a_{32} \right) + G \left(\frac{1}{6} - \frac{1}{2} (b_2 c_2^2 + b_3 c_3^2) \right) \right) + O(h^4) \end{aligned} \quad (2.76)$$

De donde se llega a las condiciones de orden de un método Runge-Kutta explícito y de 3-etapas:

- El método será de orden 1 si se cumple: $1 - (b_1 + b_2 + b_3) = 0 \Rightarrow \sum_{i=1}^3 b_i = 1$.
- El método será de orden 2 si también se cumple la siguiente condición:

$$\frac{1}{2} - (b_2 c_2 + b_3 c_3) = 0 \Rightarrow \sum_{i=1}^3 b_i c_i = \frac{1}{2} \quad (2.77)$$

- El método será de orden 3 si también se cumplen las siguientes condiciones:

$$\begin{cases} \frac{1}{6} - \frac{1}{2} (b_2 c_2^2 + b_3 c_3^2) = 0 \Rightarrow \sum_{i=1}^3 b_i c_i^2 = \frac{1}{3} \\ \frac{1}{6} - b_3 c_2 a_{32} = 0 \Rightarrow \sum_{i=1}^3 b_i \left(\sum_{j < i} a_{ij} c_j \right) = \frac{1}{6} \end{cases} \quad (2.78)$$

En general, el error local de truncamiento de un método Runge-Kutta de s-etapas viene dado por la siguiente expresión [14]:

$$LTE = \sum_{i=1}^s h^{i+1} \left(\sum_{r=i+1}^s \frac{1}{\sigma_{r,q}} \left(\frac{1}{\gamma_{r,q}} - \phi_{r,q} \right) D_{r,q}(y(t_n)) \right) \quad (2.79)$$

r	q	$\sigma_{r,q}$	$\gamma_{r,q}$	$\phi_{r,q}$	$D_{r,q}$
1	1	1	1	$\sum_{i=1}^s b_i$	f
2	1	1	2	$\sum_{i=1}^s b_i c_i$	$f' f$
3	1	2	3	$\sum_{i=1}^s b_i c_i^2$	$f''(f, f)$
3	2	1	6	$\sum_{i,j=1}^s b_i \left(\sum_{j<i} a_{ij} c_j \right)$	$f' f' f$
4	1	6	4	$\sum_{i=1}^s b_i c_i^3$	$f'''(f, f, f)$
4	2	1	8	$\sum_{i,j=1}^s b_i c_i \left(\sum_{j<i} a_{ij} c_j \right)$	$f''(f, f' f)$
4	3	2	12	$\sum_{i,j=1}^s b_i \left(\sum_{j<i} a_{ij} c_j^2 \right)$	$f' f''(f, f)$
4	4	1	24	$\sum_{i,j,k=1}^s b_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} c_k \right) \right)$	$f' f' f' f$
5	1	24	5	$\sum_{i=1}^s b_i c_i^4$	$f^{(4)}(f, f, f, f)$
5	2	2	10	$\sum_{i,j=1}^s b_i c_i^2 \left(\sum_{j<i} a_{ij} c_j \right)$	$f^{(3)}(f, f, f' f)$
5	3	2	15	$\sum_{i,j=1}^s b_i c_i \left(\sum_{j<i} a_{ij} c_j^2 \right)$	$f''(f, f''(f, f))$
5	4	1	30	$\sum_{i,j,k=1}^s b_i c_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} c_k \right) \right)$	$f''(f, f' f' f)$
5	5	2	20	$\sum_{i=1}^s b_i \left(\sum_{j=1}^s a_{ij} c_j \right)^2$	$f''(f' f, f' f)$
5	6	6	20	$\sum_{i,j=1}^s b_i \left(\sum_{j<i} a_{ij} c_j^3 \right)$	$f' f'''(f, f, f)$
5	7	1	40	$\sum_{i,j,k=1}^s b_i \left(\sum_{j<i} a_{ij} c_j \left(\sum_{k<j} a_{jk} c_k \right) \right)$	$f' f''(f, f' f)$
5	8	2	60	$\sum_{i,j,k=1}^s b_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} c_k^2 \right) \right)$	$f' f' f''(f, f)$
5	9	1	120	$\sum_{i,j,k,m=1}^s b_i \left(\sum_{j<i} a_{ij} \left(\sum_{k<j} a_{jk} \left(\sum_{m<k} a_{km} c_m \right) \right) \right)$	$f' f' f' f' f$

Tabla 2.18: Valores de $\sigma_{r,q}$, $\gamma_{r,q}$, $\phi_{r,q}$ y $D_{r,q}$ hasta orden 5.

siendo $\sigma_{r,q}$ y $\gamma_{r,q}$ constantes, $\phi_{r,q}$ una constante que depende de los coeficientes a_{ij} , b_i , c_i del método y $D_{r,q}$ distintos sumandos de la derivada de orden r de $y(t)$. Se han recogido en la Tabla 2.18 los valores de $\sigma_{r,q}$, $\gamma_{r,q}$, $\phi_{r,q}$ y $D_{r,q}$ correspondientes a órdenes 1 – 5.

Un método Runge-Kutta es de orden p si se cumple:

$$\frac{1}{\sigma_{r,q}} \left(\frac{1}{\gamma_{r,q}} - \phi_{r,q} \right) = 0, \quad \forall q, r = 1, 2, \dots, p \quad (2.80)$$

En este caso el error local de truncamiento es de orden $(p+1)$ y viene dado por:

$$LTE = h^{p+1} \sum_{r=p+1} \frac{1}{\sigma_{r,q}} \left(\frac{1}{\gamma_{r,q}} - \phi_{r,q} \right) D_{r,q}(y(t_n)) + O(h^{p+2}) \quad (2.81)$$

Sustituyendo los valores de $\sigma_{r,q}$, $\gamma_{r,q}$, $\phi_{r,q}$ en (2.80) se consiguen las condiciones que tienen que cumplir los coeficientes a_{ij} , b_i , c_i para que el método sea de orden p . Es lo

que hemos hecho para conseguir las 17 condiciones de (2.82) que tiene que cumplir un método Runge-Kutta de orden 5. En la Tabla 2.19 se recogen el número de condiciones a cumplir en cada derivada de orden r de y , así como el número total de condiciones. Y en la Tabla 2.20 se ha recogido la relación entre las etapas de un método Runge-Kutta explícito y su orden. Así, para $p > 4$ no existe ningún método Runge-Kutta explícito de p -etapas y orden p [52].

$$\begin{array}{llll}
 1. \phi_{1,1} = 1 & 5. \phi_{4,1} = \frac{1}{4} & 9. \phi_{5,1} = \frac{1}{5} & 13. \phi_{5,5} = \frac{1}{20} \\
 2. \phi_{2,1} = \frac{1}{2} & 6. \phi_{4,2} = \frac{1}{8} & 10. \phi_{5,2} = \frac{1}{10} & 14. \phi_{5,6} = \frac{1}{20} \\
 3. \phi_{3,1} = \frac{1}{3} & 7. \phi_{4,3} = \frac{1}{12} & 11. \phi_{5,3} = \frac{1}{15} & 15. \phi_{5,7} = \frac{1}{40} \\
 4. \phi_{3,2} = \frac{1}{6} & 8. \phi_{4,4} = \frac{1}{24} & 12. \phi_{5,4} = \frac{1}{30} & 16. \phi_{5,8} = \frac{1}{60} \\
 & & & 17. \phi_{5,9} = \frac{1}{120}
 \end{array} \tag{2.82}$$

Orden	1	2	3	4	5	6	7	8	9	10
Nº de condiciones	1	1	2	4	9	20	48	115	286	719
Nº total de condiciones	1	2	4	8	17	37	85	200	486	1205

Tabla 2.19: Nº de condiciones a cumplir en los métodos Runge-Kutta (órdenes 1-10).

Orden del método	1	2	3	4	5	6	7	8	9	10
Número mínimo de etapas	1	2	3	4	6	7	9	11	12-17	13-17

Tabla 2.20: Relación entre orden y etapas en los métodos Runge-Kutta explícitos [52].

2.3.3. Características de estabilidad

Para estudiar las características de estabilidad de un método Runge-Kutta hay que aplicar el método a la ecuación de test $y' = \lambda y$, obteniéndose una ecuación en diferencias de orden 1:

$$y_{n+1} = R(\hat{h})y_n, \quad \hat{h} = h\lambda \tag{2.83}$$

$R(\hat{h})$ es la función de estabilidad del método y viene dada por [52]:

$$R(\hat{h}) = 1 + \hat{h}b^T (I - \hat{h}A)^{-1} e \tag{2.84}$$

siendo I la matriz identidad de dimensión s y $e = [1, 1, \dots, 1]^T \in \mathbb{R}^s$.

Dekker y Verwer dan una forma alternativa de la función de estabilidad [52]:

$$R(\hat{h}) = \frac{\det(I - \hat{h}A + \hat{h}eb^T)}{\det(I - \hat{h}A)} \quad (2.85)$$

Y de (2.85) se deduce que:

- Cuando el método Runge-Kutta es explícito (A es estrictamente triangular inferior), la matriz $(I - \hat{h}A)$ también es triangular inferior siendo el 1 el elemento que aparece en todas las posiciones de la diagonal principal. Por tanto, $\det(I - \hat{h}A) = 1$ y la función de estabilidad (2.85) se reduce a un polinomio en \hat{h} .
- En los métodos Runge-Kutta explícitos no se puede cumplir la condición de estabilidad absoluta: cuando $|\hat{h}| \rightarrow \infty$, y_{n+1} no está acotado porque $R(\hat{h})$ es un polinomio. Por lo que, los métodos explícitos Runge-Kutta tienen una región de estabilidad absoluta finita.
- En el caso de un método Runge-Kutta de s -etapas y de orden p explícito, siendo $s = p$ (esto sucede cuando $s = 1, 2, 3, 4$), la región de estabilidad siempre es la misma [52] y la función de estabilidad es:

$$R(\hat{h}) = 1 + \sum_{j=1}^s \frac{\hat{h}^j}{j!} \quad (2.86)$$

En la Figura 2.9 se muestran las cuatro regiones de estabilidad para el caso $s = p$.

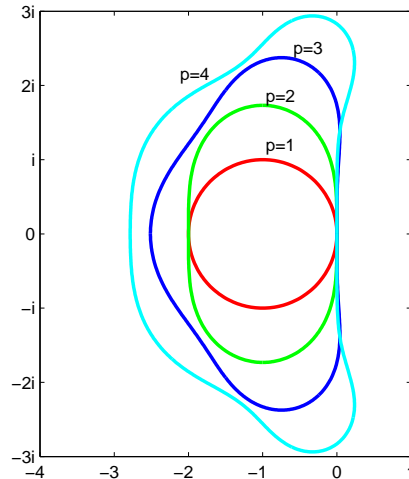


Figura 2.9: Regiones de estabilidad Runge-Kutta con $s = p$ (interior a curva).

- Si el método de Runge-Kutta explícito es de s -etapas y orden p siendo $p < s$ (y esto sucede cuando $p > 4$ en los métodos explícitos), la función de estabilidad es:

$$R(\hat{h}) = 1 + \hat{h} + \frac{\hat{h}^2}{2!} + \frac{\hat{h}^3}{3!} + \dots + \frac{\hat{h}^p}{p!} + \sum_{q=p+1}^s \gamma_q \hat{h}^q \quad (2.87)$$

Los coeficientes γ_q dependen de los coeficientes del método Runge-Kutta que estamos utilizando. Ahora sí que existe una pequeña posibilidad para mejorar la región de estabilidad, pero los intentos para explotar esta posibilidad no producen ningún resultado espectacular [52].

- Cuando el método no es explícito, el determinante de la matriz $(I - \hat{h}A)$ es un polinomio en \hat{h} y la función de estabilidad (2.85) es una función racional en \hat{h} .
- Cuando el método Runge-Kutta no es explícito, es posible que se cumpla la condición de estabilidad absoluta: cuando $|\hat{h}| \rightarrow \infty$, y_{n+1} está acotado y la región de estabilidad absoluta puede ser una región infinita.

Los métodos Runge-Kutta implícitos tienen la ventaja potencial de que comparados con los métodos explícitos, cuentan con menos etapas para conseguir el mismo orden y además su región de estabilidad es mayor que el de los métodos explícitos. La desventaja de los métodos implícitos es la naturaleza implícita de al menos alguna de las etapas, lo que nos obliga a utilizar métodos iterativos.

En adelante trabajaremos con los métodos Runge-Kutta explícitos.

2.3.4. Métodos explícitos anidados

El término principal del error local de truncamiento de los métodos Runge-Kutta es muy costoso de calcular (2.81). Merson (1957) propuso una forma de estimar el error en función de los k_i que ya se han calculado durante el paso. La clave de la idea de Merson se basa en considerar los métodos de Runge-Kutta de orden p y $(p+1)$ que comparten los mismos coeficientes c_i, a_{ij} . A este proceso se le llama anidación. La tabla Butcher correspondiente a un método Runge-Kutta anidado tiene la forma de la Tabla 2.21.

c	A
	\mathbf{b}^T
	$\hat{\mathbf{b}}^T$
	\mathbf{E}^T

Tabla 2.21: Tabla Butcher de los métodos Runge-Kutta anidados.

El método que está definido por \mathbf{c} , A y \mathbf{b}^T es de orden p y el que está definido por \mathbf{c} , A y $\hat{\mathbf{b}}^T$ es de orden $(p+1)$. En los métodos Runge-Kutta anidados las constantes

k_i dadas por la expresión (2.66) son las mismas para ambos métodos (el de orden p y $(p+1)$). La única diferencia entre ambos es la última suma que se efectúa para calcular los términos \hat{y}_{n+1} , y_{n+1} de orden $(p+1)$ y p respectivamente:

$$\hat{y}_{n+1} = y_n + h \sum_{i=1}^s \hat{b}_i k_i, \quad y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i \quad (2.88)$$

El error local de truncamiento se estima mediante la diferencia de los dos valores \hat{y}_{n+1} , y_{n+1} :

$$\hat{y}_{n+1} - y_{n+1} = h \sum_{i=1}^s E_i k_i \quad (2.89)$$

siendo: $E_i = \hat{b}_i - b_i$, de la misma manera que $\mathbf{E}^T = \hat{\mathbf{b}}^T - \mathbf{b}^T$ en la Tabla 2.21.

Cuando utilizamos el valor y_{n+1} calculado con el método de orden p como valor inicial del siguiente paso, el método es de orden p y el método anidado se denota como $(p, p+1)$. Sin embargo, cuando utilizamos el valor \hat{y}_{n+1} calculado con el método de orden $(p+1)$ como valor inicial del siguiente paso, el método es de orden $(p+1)$ y el método anidado se denota por $(p+1, p)$.

Algunos métodos Runge-Kutta anidados son estos:

- Fehlberg ha generado varios pares de Runge-Kutta anidados [26, 27]. El método (4,5) más conocido es el RKF45, donde los coeficientes se eligen para que el error local de truncamiento sea *pequeño*. Es un método de 6 etapas y sus polinomios de estabilidad son:

- Orden 4: $R(\hat{h}) = 1 + \hat{h} + \frac{\hat{h}^2}{2!} + \frac{\hat{h}^3}{3!} + \frac{\hat{h}^4}{4!} + \frac{\hat{h}^5}{104}$
- Orden 5: $R(\hat{h}) = 1 + \hat{h} + \frac{\hat{h}^2}{2!} + \frac{\hat{h}^3}{3!} + \frac{\hat{h}^4}{4!} + \frac{\hat{h}^5}{5!} + \frac{\hat{h}^6}{2080}$

- Otro método Runge-Kutta anidado conocido es el que desarrollaron Dormand y Prince [23] y que se conoce como DOPRI(5,4). Es el método que utiliza la *ode45* de Matlab. Es un método de 7 etapas de tal manera que el vector $\hat{\mathbf{b}}^T$ hace que el método sea de orden 5 y \mathbf{b}^T que sea de orden 4. En la práctica se trata de un método de 6 etapas, ya que el último coeficiente del vector $\hat{\mathbf{b}}^T$ es 0. Se trata de los métodos denominados FSAL (*First Same As Last*), significa que la evaluación de la derivada en la última etapa del paso sirve como evaluación de la primera etapa del siguiente paso. Sus coeficientes están recogidos en la Tabla 2.22 y su región de estabilidad está representada en la Figura 2.10. Los polinomios de estabilidad del par anidado DOPRI(5,4) son estos:

- Orden 4: $R(\hat{h}) = 1 + \hat{h} + \frac{\hat{h}^2}{2} + \frac{\hat{h}^3}{6} + \frac{\hat{h}^4}{24} + \frac{149\hat{h}^5}{16299} + \frac{41\hat{h}^6}{30559} + \frac{\hat{h}^7}{24000}$
- Orden 5: $R(\hat{h}) = 1 + \hat{h} + \frac{\hat{h}^2}{2} + \frac{\hat{h}^3}{6} + \frac{\hat{h}^4}{24} + \frac{\hat{h}^5}{120} + \frac{\hat{h}^6}{600}$

c_i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	a_{i7}
0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
\mathbf{b}^T	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$
$\hat{\mathbf{b}}^T$	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
\mathbf{E}^T	$\frac{71}{57600}$	0	$-\frac{71}{16695}$	$\frac{71}{1920}$	$-\frac{17253}{339200}$	$\frac{22}{525}$	$-\frac{1}{40}$

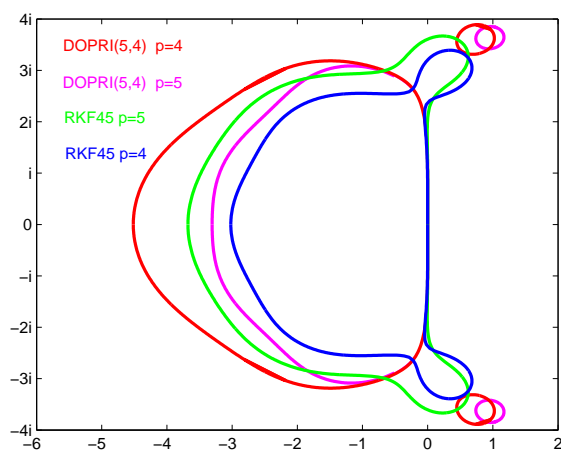
Tabla 2.22: Constantes del método Runge-Kutta DOPRI (5,4) (*ode45*).

Figura 2.10: Regiones de estabilidad RKF45 y DOPRI(5,4) (interior a curva).

Capítulo 3

Métodos lineales multipaso para EDOs de orden 2

No sé lo que pareceré a los ojos del mundo, pero a los míos es como si hubiese sido un muchacho que juega en la orilla del mar y se divierte de tanto en tanto encontrando un guijarro más pulido o una concha más hermosa, mientras el inmenso océano de la verdad se extendía, inexplorado frente a mí.

ISAAC NEWTON

Los sistemas de EDOs de segundo orden que se obtienen tras la semidiscretización de la EDP tipo onda mediante el Método de Elementos Finitos (MEF) presentan mucha rigidez. La resolución de éstos requiere el uso de métodos numéricos con buenas características de estabilidad y disipación numérica controlada (amortiguamiento algorítmico) en el rango de las frecuencias altas.

Algunos métodos desarrollados para el ámbito lineal se presentan en [47]. Entre estos métodos destacan el método Collocation [42], el método de Wilson [68], el HHT- α [43], el método de Houbolt [31], o métodos más recientes como el alfa-generalizado [21].

La presencia de inestabilidades numéricas al resolver problemas no lineales mediante métodos que resultan incondicionalmente estables en el ámbito lineal, ha motivado la búsqueda de métodos numéricos que conservan la energía y/o el momento angular. En este sentido, han sido de interés los esquemas de conservación de energía-momento de Simo y Tarnow [64] y de González [33]; y los métodos *Energy Dissipative, Momentum Conserving*, EDMC, de Armero y Romero [4, 5] que conservan el momento angular, disipan la energía y son capaces de eliminar las frecuencias altas.

En este capítulo estudiaremos algunos métodos de la mecánica clásica computacional desarrollados en el ámbito lineal. Son métodos que resuelven EDOs de segundo orden, sin la reducción previa a orden 1, y que se reducen a métodos multipaso lineales para EDOs de orden 2. Empezaremos recordando la forma de un método lineal multipaso para

EDOs de orden 2 y caracterizando su orden de precisión. A continuación, calcularemos el radio espectral, el amortiguamiento algorítmico y el error relativo en el periodo de estos métodos, lo que permitirá conocer el comportamiento de estos métodos frente a altas frecuencias.

3.1. Métodos lineales multipaso para EDOs de orden 2

Dada una ecuación diferencial de segundo orden de condiciones iniciales del tipo:

$$y'' = f(t, y, y'), \quad y(a) = \eta, y'(a) = \hat{\eta} \quad (3.1)$$

buscamos una solución a la misma en el intervalo $a \leq t \leq b$, siendo a, b finitos. Asumimos que la función f cumple las condiciones de Lipschitz, y llamaremos $y(t)$ a la solución exacta del problema (3.1). Si escribimos (3.1) como un sistema de EDOs de orden 1 y si a éste le aplicamos un método multipaso de \tilde{k} -pasos (2.3), se tiene:

$$\begin{cases} \sum_{j=0}^{\tilde{k}} \tilde{\alpha}_j y_{n+j} = h \sum_{j=0}^{\tilde{k}} \tilde{\beta}_j y'_{n+j} \\ \sum_{j=0}^{\tilde{k}} \tilde{\alpha}_j y'_{n+j} = h \sum_{j=0}^{\tilde{k}} \tilde{\beta}_j f(t_{n+j}, y_{n+j}, y'_{n+j}) \end{cases} \quad (3.2)$$

Eliminando $\{y'_{n+j} : j = 0, 1, \dots, \tilde{k}\}$ en (3.2) resulta [38]:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h^2 \sum_{j=0}^k \beta_j f(t_{n+j}, y_{n+j}), \quad \text{siendo } k = 2\tilde{k} \quad (3.3)$$

La expresión (3.3) corresponde a un método lineal multipaso o de k -pasos para EDOs de segundo orden, que es similar a la expresión de un método lineal multipaso para EDOs de orden 1 (2.3), pero con coeficientes distintos y donde aparece el factor h^2 en lugar del factor h . Esto implica que la acumulación del error local en global es con un factor $1/h^2$ y no con $1/h$ como sucedía en EDOs de orden 1. Es decir, en EDOs de orden 2: $LTE = O(h^{p+2}) \Rightarrow GTE = O(h^p)$.

También son válidas para el caso de métodos lineales multipaso para EDOs de orden 2 las definiciones y expresiones del error global de truncamiento (2.4), asunción de localización (2.5), error local de truncamiento (2.6) y desacoplamiento del sistema para obtener un problema escalar del Capítulo 2.

En el caso de métodos lineales multipaso para EDOs de orden 2 (3.3), se define el operador lineal diferencial de la siguiente manera:

$$\mathcal{L}[y(t); h] = \sum_{j=0}^k [\alpha_j y(t + jh) - h^2 \beta_j y''(t + jh)] \quad (3.4)$$

siendo $y(t)$ una función arbitraria y diferenciable cuantas veces queramos en el intervalo $[a, b]$. Pero si $y(t)$ es una solución de la EDO (3.1), el operador define el error local de truncamiento del método. Entonces, utilizando desarrollos de Taylor de las funciones $y(t)$ e $y''(t)$ alrededor del valor t_n se tiene [38, 51]:

$$\mathcal{L}[y(t); h] = C_0 y(t_n) + C_1 h y'(t_n) + C_2 h^2 y''(t_n) + \dots + C_q h^q y^{(q)}(t_n) + \dots \quad (3.5)$$

siendo las constantes C_i las siguientes:

$$\begin{cases} C_0 = \sum_{i=0}^k \alpha_i \\ C_1 = \sum_{i=0}^k i \alpha_i \\ C_2 = \frac{1}{2!} \left(\sum_{i=0}^k i^2 \alpha_i \right) - \left(\sum_{i=0}^k \beta_i \right) \\ C_q = \frac{1}{q!} \left(\sum_{i=0}^k i^q \alpha_i \right) - \frac{1}{(q-2)!} \left(\sum_{i=0}^k i^{q-2} \beta_i \right), \quad q \geq 3 \end{cases} \quad (3.6)$$

En este caso, el orden del error local de truncamiento es un factor h más fuerte que en el caso de métodos para EDOs de orden 1 [41]. Así, el método numérico dado por (3.3) es de orden p si el error local de truncamiento es $O(h^{p+2})$, es decir, si el error global de truncamiento es $O(h^p)$. Por tanto, si $C_0 = C_1 = \dots = C_p = C_{p+1} = 0$ y $C_{p+2} \neq 0$:

$$LTE_{n+k} = C_{p+2} h^{p+2} y^{(p+2)}(t_n) + O(h^{p+3}) \quad (3.7)$$

Al término $C_{p+2} h^{p+2} y^{(p+2)}(t_n)$ se le llama término principal del error local de truncamiento y a la constante C_{p+2} constante de error. Lo mismo que en el caso de EDOs de orden 1, también existe otra versión de la constante de error para EDOs de segundo orden [38], que viene dada por la expresión:

$$C = \frac{C_{p+2}}{\sigma(1)} \quad (3.8)$$

siendo $\sigma(r)$ el 2º polinomio característico asociado al método y que viene dado por (2.23).

En este capítulo estudiaremos métodos que sirven para resolver sistemas de EDOs de orden 2 de la forma:

$$Md'' + Cd' + F_{int}(d) = F_{ext}, \quad d(0) = d_0, \quad d'(0) = v_0 \quad (3.9)$$

Cuando $F_{int}(d)$ es lineal ($F_{int}(d) = Kd$) (3.9) queda:

$$Md'' + Cd' + Kd = F, \quad d(0) = d_0, \quad d'(0) = v_0 \quad (3.10)$$

utilizando la notación $F = F_{ext}$, con el objeto de simplificar la escritura.

Concretamente, estudiaremos métodos que son reducibles a métodos lineales multipaso para la resolución de EDOs de orden 2 de la forma (3.3). El estudio de estabilidad

de estos métodos consiste en aplicar el método a la EDO de orden 2 escalar lineal homogénea y de coeficientes constantes, que se conoce como ecuación de test:

$$u'' + \omega^2 u = 0 \quad (3.11)$$

Tras aplicar el método a la ecuación de test, se obtiene un sistema de ecuaciones en diferencias de orden 1 del tipo (2.9):

$$U_{n+1} = A(\Omega)U_n \quad (3.12)$$

siendo $\Omega = h\omega$, U_{n+1} y U_n las matrices con las incógnitas en los instantes t_{n+1} y t_n y A el factor de amplificación. El método es estable si su radio espectral ρ , es menor o igual que 1 (estrictamente menor que 1 para raíces múltiples):

$$\rho(A) = \max \{|\lambda_i| : \lambda_i \text{ autovalor de } A\} \leq 1 \quad (3.13)$$

3.2. Método alfa-generalizado

Consideramos la siguiente EDO de orden 2:

$$Md'' + Cd' + Kd = F, \quad d(0) = d_0, \quad d'(0) = v_0 \quad (3.14)$$

Transformamos (3.14) en un sistema de 2 EDOs de orden 1:

$$\begin{cases} y(t) = d(t) \\ v(t) = d'(t) \end{cases} \Rightarrow \begin{cases} y'(t) = d'(t) = v(t) \\ v'(t) = d''(t) = a(t) \end{cases} \quad (3.15)$$

y cumpliéndose (3.14). Si aplicamos el método trapezoidal al sistema (3.15) se tiene:

$$\begin{cases} d_{n+1} = d_n + \frac{v_n + v_{n+1}}{2}h \\ v_{n+1} = v_n + \frac{a_n + a_{n+1}}{2}h \end{cases} \quad (3.16)$$

En (3.16) se observa que el desplazamiento se actualiza utilizando la velocidad constante media de los instantes t_n y t_{n+1} , y que la velocidad se actualiza mediante la aceleración constante media de los instantes t_n y t_{n+1} . Sustituyendo la segunda fórmula de avance de (3.16) en la primera se tiene:

$$d_{n+1} = d_n + \frac{v_n}{2}h + \left(v_n + \frac{a_n + a_{n+1}}{2}h \right) \frac{h}{2} = d_n + v_n h + \left(\frac{a_n}{2} + \frac{a_{n+1}}{2} \right) \frac{h^2}{2} \quad (3.17)$$

La expresión (3.17) corresponde a la fórmula exacta del movimiento uniformemente acelerado de aceleración constante, siendo ésta la media de las aceleraciones de los instantes t_n y t_{n+1} .

En (3.16), podríamos optar por actualizar el desplazamiento ponderando las velocidades de los instantes t_n y t_{n+1} , y actualizar la velocidad ponderando las aceleraciones de los mismos instantes. También se podría reproducir este esquema de ponderaciones en la expresión (3.14). La utilización de este esquema en ambas expresiones (3.14) y (3.16), da lugar a las siguientes posibilidades:

- Generalizamos el método trapezoidal introduciendo un parámetro 2β de ponderación en (3.17) y otro parámetro γ en la segunda expresión de (3.16):

$$\begin{cases} d_{n+1} = d_n + v_n h + ((1 - 2\beta)a_n + 2\beta a_{n+1}) \frac{h^2}{2} \\ v_{n+1} = v_n + ((1 - \gamma)a_n + \gamma a_{n+1}) h \end{cases} \quad (3.18)$$

El sistema biparamétrico (3.18) es el que se utiliza en el método de Newmark que veremos en este capítulo [55] e incluye al método trapezoidal (3.16) considerando $\beta = 0,25$, $\gamma = 0,5$.

- Si además de las ponderaciones de (3.18), también ponderamos la expresión (3.14), exceptuando en esta ponderación las fuerzas de inercia, se tiene:

$$M a_{n+1} + (1 - \alpha) C v_{n+1} + \alpha C v_n + (1 - \alpha) K d_{n+1} + \alpha K d_n = F(t_{n+1-\alpha}) \quad (3.19)$$

siendo $t_{n+1-\alpha} = (1 - \alpha)t_{n+1} + \alpha t_n$. El sistema triparamétrico formado por (3.19) y (3.18), incluye al método de Newmark y es el esquema utilizado en el método HHT- α [43].

- Si manteniendo las ponderaciones de (3.18), también ponderamos la expresión (3.14), pero esta vez ponderando sólo las fuerzas de inercia, se tiene:

$$(1 - \alpha_B) M a_{n+1} + \alpha_B M a_n + C v_{n+1} + K d_{n+1} = F(t_{n+1}) \quad (3.20)$$

También el sistema triparamétrico formado por (3.20) y (3.18), incluye al método de Newmark y es el esquema que utiliza el método de Bossak-Newmark o WBZ- α [70].

- Finalmente, si la ponderación de la expresión (3.14) se hace ponderando de distinta forma las fuerzas de inercia y el resto de los sumandos se obtiene:

$$M a_{n+1-\alpha_m} + C v_{n+1-\alpha_f} + K d_{n+1-\alpha_f} = F(t_{n+1-\alpha_f}) \quad (3.21)$$

donde:

$$\begin{cases} d_{n+1-\alpha_f} = (1 - \alpha_f) d_{n+1} + \alpha_f d_n \\ v_{n+1-\alpha_f} = (1 - \alpha_f) v_{n+1} + \alpha_f v_n \\ a_{n+1-\alpha_m} = (1 - \alpha_m) a_{n+1} + \alpha_m a_n \\ t_{n+1-\alpha_f} = (1 - \alpha_f) t_{n+1} + \alpha_f t_n \end{cases} \quad (3.22)$$

El sistema triparamétrico formado por (3.21) y (3.18), engloba a los métodos citados anteriormente (Newmark, HHT- α y Bossak-Newmark) y es el esquema que

utiliza el método alfa-generalizado [21], el cual viene dado por:

$$\begin{cases} d_{n+1} = d_n + \Delta t v_n + \frac{\Delta t^2}{2} [(1 - 2\beta) a_n + 2\beta a_{n+1}] \\ v_{n+1} = v_n + \Delta t [(1 - \gamma) a_n + \gamma a_{n+1}] \\ M a_{n+1 - \alpha_m} + C v_{n+1 - \alpha_f} + K d_{n+1 - \alpha_f} = F(t_{n+1 - \alpha_f}) \end{cases} \quad (3.23)$$

siendo $\beta, \gamma, \alpha_f, \alpha_m$ parámetros reales, $d' = v$, $d'' = a$ y los valores intermedios (3.22) que representan para $\alpha_f, \alpha_m \in [0, 1]$ la interpolación lineal de los valores extremos de las variables t, d, v, a en el intervalo temporal $[t_n, t_{n+1}]$.

Estudiaremos a continuación el orden de precisión del método alfa-generalizado, que permitirá concluir el orden de precisión de los métodos que derivan de él. Estudiaremos también la estabilidad del método de Newmark, su comportamiento frente a frecuencias altas y la condición para que la disipación de éstas sea máxima.

El primer paso para deducir el orden de precisión es obtener la matriz de amplificación A del método alfa-generalizado. Para ello, aplicamos el método a la ecuación de test (3.11) considerando $\xi \neq 0$, es decir: $u'' + 2\xi\omega u' + \omega^2 u = 0$. Agrupamos los términos del instante t_{n+1} a un lado de las igualdades y los del instante t_n al otro lado:

$$\begin{cases} u_{n+1} - h^2\beta a_{n+1} = u_n + h v_n + \left(\frac{1}{2} - \beta\right) h^2 a_n \\ v_{n+1} - h\gamma a_{n+1} = v_n + h a_n (1 - \gamma) \\ \omega^2 (1 - \alpha_f) u_{n+1} + 2\xi\omega (1 - \alpha_f) v_{n+1} + (1 - \alpha_m) a_{n+1} = -\omega^2 \alpha_f u_n - 2\xi\omega \alpha_f v_n - \alpha_m a_n \end{cases} \quad (3.24)$$

El sistema (3.24) es un sistema de 3 ecuaciones en diferencias de orden 1 que se puede escribir matricialmente de la siguiente manera:

$$\bar{A}_1 U_{n+1} = \bar{A}_2 U_n \Rightarrow U_{n+1} = A U_n \quad (3.25)$$

donde:

$$\begin{cases} U_{n+1} = (u_{n+1}, h v_{n+1}, h^2 a_{n+1})^T, U_n = (u_n, h v_n, h^2 a_n)^T, h = \Delta t, \\ A = \bar{A}_1^{-1} \cdot \bar{A}_2, \bar{A}_1 = \begin{pmatrix} 1 & 0 & -\beta \\ 0 & \frac{1}{h} & -\frac{1}{h}\gamma \\ \omega^2 (1 - \alpha_f) & \frac{2\xi\omega}{h} (1 - \alpha_f) & \frac{1}{h^2} (1 - \alpha_m) \end{pmatrix}, \\ \bar{A}_2 = \begin{pmatrix} 1 & 1 & \frac{1}{2} - \beta \\ 0 & \frac{1}{h} & \frac{1}{h}(1 - \gamma) \\ -\omega^2 \alpha_f & \frac{-2\xi\omega \alpha_f}{h} & \frac{-\alpha_m}{h^2} \end{pmatrix} \end{cases} \quad (3.26)$$

y A es el factor de amplificación del método.

Debemos aplicar sucesivamente las ecuaciones del método en instantes t_{n+i} con $i = 1, 2, 3$, para convertir el sistema (3.25) de 3 ecuaciones en diferencias, en la ecuación en diferencias equivalente de orden 3. De esta forma se obtienen tres sistemas de la forma

(3.25), que se pueden agrupar en un único sistema que está formado por 9 ecuaciones y 12 incógnitas (u_{n+i} , v_{n+i} , a_{n+i} para $i = 0, 1, 2, 3$):

$$\begin{pmatrix} U_{n+1} \\ U_{n+2} \\ U_{n+3} \end{pmatrix} = \left[\begin{array}{c|c|c} A & 0 & 0 \\ \hline 0 & A & 0 \\ \hline 0 & 0 & A \end{array} \right] \begin{pmatrix} U_n \\ U_{n+1} \\ U_{n+2} \end{pmatrix} \quad (3.27)$$

Eliminando 8 ecuaciones y 8 incógnitas en (3.27), obtenemos una única ecuación en diferencias de orden 3 para la incógnita u_n :

$$u_{n+3} + a_1 u_{n+2} + a_2 u_{n+1} + a_3 u_n = 0 \quad (3.28)$$

siendo a_1, a_2, a_3 constantes a determinar, y cuyo polinomio característico:

$$p(r) = r^3 + a_1 r^2 + a_2 r + a_3 \quad (3.29)$$

deberá coincidir con el polinomio característico $\det(A - rI)$ del sistema equivalente (3.25) del que procede. Teniendo en cuenta que el polinomio característico de la matriz A puede expresarse como:

$$\det(A - rI) = r^3 - 2A_1 r^2 + A_2 r - A_3 \quad (3.30)$$

donde A_1, A_2, A_3 son las invariantes de la matriz A :

$$\begin{cases} A_1 = \frac{1}{2} \text{traza}(A) \\ A_2 = \frac{1}{2} \cdot \left((\text{traza}(A))^2 - \text{traza}(A^2) \right) \\ A_3 = \det(A) \end{cases} \quad (3.31)$$

Podemos concluir que: $a_1 = -2A_1$, $a_2 = A_2$, $a_3 = -A_3$. Por lo que (3.28) queda:

$$u_{n+3} - 2A_1 u_{n+2} + A_2 u_{n+1} - A_3 u_n = 0 \quad (3.32)$$

En el caso más general con $\xi \neq 0$ las expresiones de las invariantes (3.31) son muy extensas:

$$\begin{cases} A_1 = \frac{\omega^2 h^2 (-4\beta + 6\beta\alpha_f + 2\gamma + 1 - 2\gamma\alpha_f - \alpha_f) + 2\xi\omega h (2 - 2\alpha_f - 4\gamma + 6\gamma\alpha_f) - 4 + 6\alpha_m}{4(\omega^2 h^2 (-\beta + \beta\alpha_f) + \xi\omega h (-2\gamma + 2\gamma\alpha_f) - 1 + \alpha_m)} \\ A_2 = \frac{\omega^2 h^2 (-2\beta + 6\beta\alpha_f + 2\gamma - 1 - 4\gamma\alpha_f) + 2\xi\omega h (-2\gamma + 2 - 4\alpha_f + 6\gamma\alpha_f) - 2 + 6\alpha_m}{2(\omega^2 h^2 (-\beta + \beta\alpha_f) + \xi\omega h (-2\gamma + 2\gamma\alpha_f) - 1 + \alpha_m)} \\ A_3 = \frac{\omega^2 h^2 (-2\gamma\alpha_f + \alpha_f + 2\beta\alpha_f) + 2\xi\omega h (2\gamma\alpha_f - 2\alpha_f) + 2\alpha_m}{2(\omega^2 h^2 (-\beta + \beta\alpha_f) + \xi\omega h (-2\gamma + 2\gamma\alpha_f) - 1 + \alpha_m)} \end{cases} \quad (3.33)$$

Y para $\xi = 0$ toman estos valores:

$$\begin{cases} A_1 = \frac{\omega^2 h^2 (\beta(3\alpha_f - 2) + (\gamma + \frac{1}{2})(1 - \alpha_f)) - 2 + 3\alpha_m}{2(-1 + \alpha_m + \omega^2 h^2 (-\beta + \beta\alpha_f))} \\ A_2 = \frac{\omega^2 h^2 (-2\gamma\alpha_f + 3\beta\alpha_f + \gamma - \beta - \frac{1}{2}) - 1 + 3\alpha_m}{-1 + \alpha_m + \omega^2 h^2 (-\beta + \beta\alpha_f)} \\ A_3 = \frac{\omega^2 h^2 (-\gamma\alpha_f + \frac{1}{2}\alpha_f + \beta\alpha_f) + \alpha_m}{-1 + \alpha_m + \omega^2 h^2 (-\beta + \beta\alpha_f)} \end{cases} \quad (3.34)$$

Sustituyendo (3.34) en (3.32) observamos que el método alfa-generalizado corresponde a un método multipaso para EDOs de orden 2 del tipo (3.3):

$$\frac{\sum_{i=0}^3 \alpha_i u_{n+i}}{\delta} = -\frac{\omega^2 h^2 \sum_{i=0}^3 \beta_i u_{n+i}}{\delta} \implies \sum_{i=0}^3 \alpha_i u_{n+i} = -\omega^2 h^2 \sum_{i=0}^3 \beta_i u_{n+i} \quad (3.35)$$

$$\text{donde: } \begin{cases} \alpha_0 = \alpha_m, & \beta_0 = -\gamma\alpha_f + \frac{1}{2}\alpha_f + \beta\alpha_f \\ \alpha_1 = 1 - 3\alpha_m, & \beta_1 = 2\gamma\alpha_f - 3\beta\alpha_f - \gamma + \beta + \frac{1}{2} \\ \alpha_2 = -2 + 3\alpha_m, & \beta_2 = \beta(3\alpha_f - 2) + (\gamma + \frac{1}{2})(1 - \alpha_f) \\ \alpha_3 = 1 - \alpha_m, & \beta_3 = \beta - \beta\alpha_f \\ \delta = -1 + \alpha_m + \omega^2 h^2(-\beta + \beta\alpha_f) \end{cases} \quad (3.36)$$

Podemos aplicar directamente las condiciones (3.6) y (3.7) y anular las constantes C_0 , C_1 , C_2 y C_3 para que el orden de precisión sea 2. En este caso no obstante, haremos el desarrollo completo considerando la ecuación de test:

$$u'' = -\omega^2 u \quad (3.37)$$

Utilizando la definición del error local de truncamiento tenemos:

$$\begin{aligned} LTE &= u(t_{n+3}) - u_{n+3}^* \\ &= u(t_{n+3}) - [2A_1 u(t_{n+2}) - A_2 u(t_{n+1}) + A_3 u(t_n)] \\ &= u(t_{n+3}) - 2A_1 u(t_{n+2}) + A_2 u(t_{n+1}) - A_3 u(t_n) \end{aligned} \quad (3.38)$$

donde u_{n+3}^* se ha calculado haciendo la asunción de localización $u(t_{n+i}) = u_{n+i}$ para $i = 0, 1, 2$. Teniendo en cuenta (3.35), la ecuación (3.38) se puede escribir como:

$$LTE = \frac{\sum_{i=0}^3 \alpha_i u(t_{n+i}) + \omega^2 h^2 \sum_{i=0}^3 \beta_i u(t_{n+i})}{\delta} \quad (3.39)$$

Y sustituyendo la expresión $u''(t_{n+i}) = -\omega^2 u(t_{n+i})$ en (3.39) llegamos a la ecuación:

$$LTE = \frac{\sum_{i=0}^3 \alpha_i u(t_{n+i}) - h^2 \sum_{i=0}^3 \beta_i u''(t_{n+i})}{\delta} \quad (3.40)$$

Utilizaremos los siguientes desarrollos de Taylor:

$$\begin{cases} u(t_{n+1}) = u(t_n) + hu'(t_n) + \frac{h^2}{2!}u''(t_n) + \frac{h^3}{3!}u'''(t_n)... \\ u(t_{n+2}) = u(t_n) + 2hu'(t_n) + 2h^2u''(t_n) + \frac{4h^3}{3}u'''(t_n)... \\ u(t_{n+3}) = u(t_n) + 3hu'(t_n) + \frac{9}{2}h^2u''(t_n) + \frac{9}{2}h^3u'''(t_n)... \end{cases} \quad (3.41)$$

Derivamos dos veces las ecuaciones (3.41) obteniendo:

$$\begin{cases} u''(t_{n+1}) = u''(t_n) + hu'''(t_n) + \frac{h^2}{2!}u^{(iv)}(t_n)... \\ u''(t_{n+2}) = u''(t_n) + 2hu'''(t_n) + 2h^2u^{(iv)}(t_n) + \frac{4h^3}{3}u^{(v)}(t_n)... \\ u''(t_{n+3}) = u''(t_n) + 3hu'''(t_n) + \frac{9}{2}h^2u^{(iv)}(t_n) + \frac{9}{2}h^3u^{(v)}(t_n)... \end{cases} \quad (3.42)$$

Sustituyendo las expresiones (3.41) y (3.42) en el numerador de la ecuación (3.40) y considerando únicamente el numerador, se tiene:

$$\begin{aligned} & \alpha_0 u(t_n) + \alpha_1 \left(u(t_n) + hu'(t_n) + \frac{h^2}{2!} u''(t_n) + \frac{h^3}{3!} u'''(t_n) \dots \right) \\ & + \alpha_2 \left(u(t_n) + 2hu'(t_n) + 2h^2 u''(t_n) + \frac{4h^3}{3} u'''(t_n) \dots \right) + \alpha_3 \left(u(t_n) + 3hu'(t_n) + \frac{9}{2} h^2 u''(t_n) + \frac{9}{2} h^3 u'''(t_n) \dots \right) \\ & - h^2 \left(\beta_0 u''(t_n) + \beta_1 \left(u''(t_n) + hu'''(t_n) + \frac{h^2}{2!} u^{(iv)}(t_n) \dots \right) + \beta_2 \left(u''(t_n) + 2hu'''(t_n) + 2h^2 u^{(iv)}(t_n) \dots \right) \right. \\ & \left. + \beta_3 \left(u''(t_n) + 3hu'''(t_n) + \frac{9}{2} h^2 u^{(iv)}(t_n) \dots \right) \right) \quad (3.43) \end{aligned}$$

Reordenamos términos de igual orden en la ecuación (3.43):

$$\begin{aligned} & u(t_n) (\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3) + hu'(t_n) (\alpha_1 + 2\alpha_2 + 3\alpha_3) \\ & + h^2 u''(t_n) \left(\frac{1}{2!} (\alpha_1 + 4\alpha_2 + 9\alpha_3) - (\beta_0 + \beta_1 + \beta_2 + \beta_3) \right) \\ & + h^3 u'''(t_n) \left(\frac{1}{3!} (\alpha_1 + 8\alpha_2 + 27\alpha_3) - (\beta_1 + 2\beta_2 + 3\beta_3) \right) + O(h^4) \quad (3.44) \end{aligned}$$

Y la expresión (3.44) se puede expresar de esta manera:

$$C_0 u(t_n) + C_1 hu'(t_n) + C_2 h^2 u''(t_n) + \dots + C_q h^q u^{(q)}(t_n) + O(h^{q+1}) \quad (3.45)$$

donde las constantes C_i son:

$$\begin{cases} C_0 = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 \\ C_1 = \alpha_1 + 2\alpha_2 + 3\alpha_3 \\ C_2 = \frac{1}{2!} (\alpha_1 + 2^2\alpha_2 + 3\alpha_3) - (\beta_0 + \beta_1 + \beta_2 + \beta_3) \\ C_q = \frac{1}{q!} (\alpha_1 + 2^q\alpha_2 + 3^q\alpha_3) - \frac{1}{(q-2)!} (\beta_1 + 2^{q-2}\beta_2 + 3^{q-2}\beta_3), \quad q \geq 3 \end{cases} \quad (3.46)$$

Por lo que, el error local de truncamiento se expresa como:

$$LTE = \frac{C_0 u(t_n) + C_1 hu'(t_n) + C_2 h^2 u''(t_n) + \dots + C_q h^q u^{(q)}(t_n) + O(h^{q+1})}{\delta} \quad (3.47)$$

Aplicando lo que hemos visto en la Sección 3.1 para métodos del tipo (3.3), el método es de orden p si las constantes $C_0 = C_1 = \dots = C_p = C_{p+1} = 0$ y $C_{p+2} \neq 0$. En el caso del método alfa-generalizado, el método será de orden 2 si $C_0 = C_1 = C_2 = C_3 = 0$ y $C_4 \neq 0$. Sustituyendo los valores de (3.36) en (3.46) tenemos:

$$\begin{cases} C_0 = \alpha_m + 1 - 3\alpha_m - 2 + 3\alpha_m + 1 - \alpha_m = 0 \\ C_1 = 1 - 3\alpha_m - 4 + 6\alpha_m + 3 - 3\alpha_m = 0 \\ C_2 = \frac{1}{2} (1 - 3\alpha_m - 8 + 12\alpha_m + 9 - 9\alpha_m) - (-\gamma\alpha_f + \frac{1}{2}\alpha_f + \beta\alpha_f + 2\gamma\alpha_f - 3\beta\alpha_f \\ \quad -\gamma + \beta + \frac{1}{2} + 3\beta\alpha_f + \gamma - \gamma\alpha_f - 2\beta + \frac{1}{2} - \frac{1}{2}\alpha_f + \beta - \beta\alpha_f) = 0 \\ C_3 = -\gamma - \alpha_m + \alpha_f + \frac{1}{2} \\ C_4 = \frac{1}{24} (50 - 36\alpha_m) - \frac{1}{2} (-2\gamma\alpha_f + 3\gamma + 2\beta + \frac{5}{2} - 2\alpha_f) \end{cases} \quad (3.48)$$

Por lo que, en la expresión (3.47) del error local de truncamiento se anulan las constantes C_i para $i = 0, 1, 2$, resultando:

$$LTE = \frac{C_3}{\delta} h^3 u'''(t_n) + O\left(\frac{h^4}{\delta}\right) \quad (3.49)$$

El método alfa-generalizado es de orden 1 si $C_3 \neq 0$, y de orden 2 cuando $C_3 = 0$. Esto sucede cuando $\gamma = 1/2 - \alpha_m + \alpha_f$ y en este caso, el error local de truncamiento es:

$$LTE = \frac{C_4}{\delta} h^4 u^{(iv)}(t_n) + O\left(\frac{h^5}{\delta}\right) \quad (3.50)$$

y la constante C_4 viene dada por:

$$C_4 = \frac{1}{12} - \frac{1}{2}\alpha_f + \gamma\alpha_f - \beta \quad (3.51)$$

Cuando hay amortiguamiento, es decir $\xi \neq 0$, la expresión del error local de truncamiento (3.39) se convierte en:

$$LTE = \frac{\boxed{\sum_{i=0}^3 \alpha_i u(t_{n+i}) - h^2 \sum_{i=0}^3 \beta_i u''(t_{n+i})} + h2\xi\omega \left(\sum_{i=0}^3 (\gamma_i u(t_{n+i}) - \beta_i h u'(t_{n+i})) \right)}{\delta} \quad (3.52)$$

$$\text{siendo: } \begin{cases} \gamma_0 = (\gamma - 1)\alpha_f \\ \gamma_1 = -3\gamma\alpha_f + 2\alpha_f + \gamma - 1 \\ \gamma_2 = -2\gamma + 3\gamma\alpha_f + 1 - \alpha_f \\ \gamma_3 = (1 - \alpha_f)\gamma \end{cases} \quad (3.53)$$

Los dos sumandos que en la expresión (3.52) aparecen dentro del cuadro son los dos sumandos de (3.40) que ya hemos calculado. Consideraremos únicamente el tercer sumando de (3.52):

$$h2\xi\omega \left(\sum_{i=0}^3 (\gamma_i u(t_{n+i}) - \beta_i h u'(t_{n+i})) \right) \quad (3.54)$$

Sustituimos en (3.54) los desarrollos de Taylor dados por (3.41) y sus primeras derivadas:

$$\begin{aligned} & h2\xi\omega \left[\gamma_0 u(t_n) - \beta_0 h u'(t_n) + \gamma_1 \left(u(t_n) + h u'(t_n) + \frac{h^2}{2!} u''(t_n) + \frac{h^3}{3!} u'''(t_n) + \dots \right) \right. \\ & - \beta_1 h \left(u'(t_n) + h u''(t_n) + \frac{h^2}{2!} u'''(t_n) + \dots \right) + \gamma_2 \left(u(t_n) + 2h u'(t_n) + 2h^2 u''(t_n) + \frac{4h^3}{3} u'''(t_n) \dots \right) \\ & - \beta_2 h \left(u'(t_n) + 2h u''(t_n) + 2h^2 u'''(t_n) \dots \right) + \gamma_3 \left(u(t_n) + 3h u'(t_n) + \frac{9}{2} h^2 u''(t_n) + \frac{9}{2} h^3 u'''(t_n) \dots \right) \\ & \left. - \beta_3 h \left(u'(t_n) + 3h u''(t_n) + \frac{9}{2} h^2 u'''(t_n) \dots \right) \right] \end{aligned}$$

Reordenamos términos en la expresión anterior y llegamos a:

$$\begin{aligned}
& 2h\xi\omega [u(t_n)(\gamma_0 + \gamma_1 + \gamma_2 + \gamma_3) + hu'(t_n)((\gamma_1 + 2\gamma_2 + 3\gamma_3) - (\beta_0 + \beta_1 + \beta_2 + \beta_3)) \\
& \quad + h^2u''(t_n)\left(\frac{1}{2!}(\gamma_1 + 2^2\gamma_2 + 3^2\gamma_3) - (\beta_1 + 2\beta_2 + 3\beta_3)\right) \\
& \quad + h^3u'''(t_n)\left(\frac{1}{3!}(\gamma_1 + 2^3\gamma_2 + 3^3\gamma_3) - \frac{1}{2!}(\beta_1 + 2^2\beta_2 + 3^2\beta_3)\right) \\
& \quad \quad \quad + O(h^4)] \quad (3.55)
\end{aligned}$$

Y la expresión (3.55) se puede expresar de esta manera:

$$2h\xi\omega \left(D_0u(t_n) + D_1hu'(t_n) + D_2h^2u''(t_n) + \dots + D_qh^qu^{(q)}(t_n) + O(h^{q+1}) \right) \quad (3.56)$$

$$\text{siendo: } \begin{cases} D_0 = \gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 \\ D_1 = (\gamma_1 + 2\gamma_2 + 3\gamma_3) - (\beta_0 + \beta_1 + \beta_2 + \beta_3) \\ D_2 = \frac{1}{2!}(\gamma_1 + 2^2\gamma_2 + 3^2\gamma_3) - (\beta_1 + 2\beta_2 + 3\beta_3) \\ D_3 = \frac{1}{3!}(\gamma_1 + 2^3\gamma_2 + 3^3\gamma_3) - \frac{1}{2!}(\beta_1 + 2^2\beta_2 + 3^2\beta_3) \end{cases} \quad (3.57)$$

Sustituyendo los valores de (3.36) y (3.53) en la expresión (3.57), tenemos que $D_i = 0$ para $i = 0, 1, 2, 3$. Por tanto, el hecho de aplicar el método para $\xi \neq 0$ no altera las condiciones de orden a las que hemos llegado en el caso de que no hay amortiguamiento, ya que los sumandos adicionales provenientes del caso en el que hay amortiguamiento son nulos. Entonces, queda justificada la condición $\gamma = 1/2 - \alpha_m + \alpha_f$ para que el método alfa-generalizado sea de orden 2. \square

El método alfa-generalizado es incondicionalmente estable cuando se cumplen las siguientes relaciones [21]:

$$\alpha_m \leq \alpha_f \leq \frac{1}{2}, \quad \beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m) \quad (3.58)$$

Se consigue disipación máxima de las frecuencias altas cuando los autovalores de la matriz de amplificación A se convierten en reales y esto sucede cuando:

$$\beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2 \quad (3.59)$$

*Observación: El método alfa-generalizado es de orden 3 cuando $C_1 = C_2 = C_3 = C_4 = 0$ y esto sucede cuando: $\gamma = \frac{1}{2} - \alpha_m + \alpha_f$ y $\beta = \frac{1}{12} + \alpha_f(\alpha_f - \alpha_m)$.

3.2.1. Método de Newmark

Cuando en el método alfa-generalizado hacemos $\alpha_f = \alpha_m = 0$, se consigue el método de Newmark [55], que viene dado por las siguientes relaciones:

$$\begin{cases} Ma_{n+1} + Cv_{n+1} + Kd_{n+1} = F_{n+1} \\ d_{n+1} = d_n + \Delta t v_n + \frac{\Delta t^2}{2} [(1 - 2\beta) a_n + 2\beta a_{n+1}] \\ v_{n+1} = v_n + \Delta t [(1 - \gamma) a_n + \gamma a_{n+1}] \end{cases} \quad (3.60)$$

En este caso las invariantes de la matriz A (3.31) son estas en el caso general $\xi \neq 0$:

$$\begin{cases} A_1 = \frac{1}{2} \cdot \frac{\omega^2 h^2 (-2\gamma - 1 + 4\beta) + 2\xi \omega h (-2 + 4\gamma) + 4}{2 + 2\beta \omega^2 h^2 + 4\xi \omega h \gamma} \\ A_2 = \frac{\omega^2 h^2 (-2\gamma + 2\beta + 1) + 2\xi \omega h (-2 + 2\gamma) + 2}{2 + 2\beta \omega^2 h^2 + 4\xi \omega h \gamma} \\ A_3 = 0 \end{cases} \quad (3.61)$$

Y el polinomio característico (3.30) queda:

$$\det(A - \lambda I) = \lambda^3 - 2A_1 \lambda^2 + A_2 \lambda = 0 \implies \lambda(\lambda^2 - 2A_1 \lambda + A_2) = 0 \quad (3.62)$$

Particularizando las condiciones de orden (3.48) del método alfa-generalizado a este caso, el método de Newmark resulta de orden 2 cuando $\gamma = 1/2$ [47], siendo su constante de error:

$$C = \frac{C_4}{\sigma(1)} = \frac{1}{12} - \beta \quad (3.63)$$

*Observación: El método podría ser de orden 3 cuando $C_1 = C_2 = C_3 = C_4 = 0$, debiéndose cumplir para ello $\gamma = 1/2$, $\beta = 1/12$. Pero en este caso, como lo veremos a continuación, el método no es incondicionalmente estable.

Para hacer el estudio de estabilidad del método de Newmark es necesario hallar los autovalores de (3.62) que resultan:

$$\begin{cases} \lambda_1 = A_1 + \sqrt{A_1^2 - A_2} \\ \lambda_2 = A_1 - \sqrt{A_1^2 - A_2} \\ \lambda_3 = 0 \end{cases} \quad (3.64)$$

Y sustituyendo los valores de las invariantes (3.61) en (3.64) tenemos:

$$\begin{cases} \lambda_1 = \frac{1}{2} \cdot \frac{\Omega^2 (-2\gamma - 1 + 4\beta) + 2\xi \Omega (-2 + 4\gamma) + 4 + K}{2 + 2\beta \Omega^2 + 4\xi \Omega \gamma} \\ \lambda_2 = \frac{1}{2} \cdot \frac{\Omega^2 (-2\gamma - 1 + 4\beta) + 2\xi \Omega (-2 + 4\gamma) + 4 - K}{2 + 2\beta \Omega^2 + 4\xi \Omega \gamma} \\ \lambda_3 = 0 \end{cases} \quad (3.65)$$

donde $\Omega = \omega h$ y K viene dado por:

$$K = \Omega [\Omega^2 (1 + 4\gamma + 4\gamma^2 - 16\beta) + \xi \Omega (-16\gamma + 8) + (16\xi^2 - 16)]^{1/2} \quad (3.66)$$

El método será estable si su radio espectral es menor o igual que 1:

$$\rho(A) = \max \{|\lambda_1|, |\lambda_2|, |\lambda_3|\} \leq 1 \quad (3.67)$$

La condición necesaria para que se cumpla (3.67) es $\gamma \geq \frac{1}{2}$. En este caso, $\gamma \geq \frac{1}{2}$, el método es incondicionalmente estable si se verifica $\frac{1}{2} \leq \gamma \leq 2\beta$. Si $\gamma \geq \frac{1}{2}$ y $\gamma > 2\beta$, el tamaño de paso deberá cumplir la condición:

$$\omega h \leq \Omega_{crit} = \frac{\xi \left(\gamma - \frac{1}{2}\right) + \left[\frac{\gamma}{2} - \beta + \xi^2 \left(\gamma - \frac{1}{2}\right)^2\right]^{1/2}}{\left(\frac{\gamma}{2} - \beta\right)} \quad (3.68)$$

para que el método resulte estable [47].

En la Figura 3.1 hemos representado los autovalores (3.65) en función de $\Omega = \omega h$ en el plano complejo para $\gamma = 1/2, \beta = 1/4$. Éstos caen en la circunferencia de centro $(0,0)$ y radio $R = 1$, es decir, su módulo es igual que la unidad y el método resulta incondicionalmente estable en este caso. En la Figura 3.2 se han dibujado los autovalores (imagen izquierda) y el radio espectral (imagen derecha) en un caso en el que el método de Newmark es inestable, $\gamma = 0,4, \beta = 0,6$. En este caso el módulo de los autovalores es mayor que la unidad. Y en la Figura 3.3 se han representado los autovalores y el radio espectral para un caso en el que el método es condicionalmente estable, $\gamma = 0,5, \beta = 0,2$. También se ha marcado el valor Ω_{crit} a partir del cual el método se hace inestable.

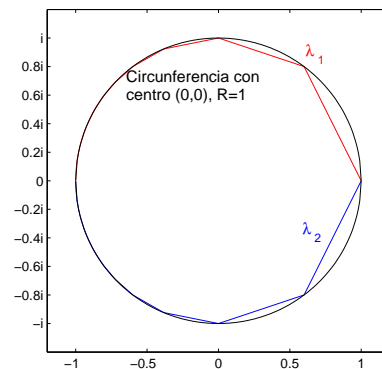


Figura 3.1: Autovalores del método de Newmark $\gamma = 1/2, \beta = 1/4$.

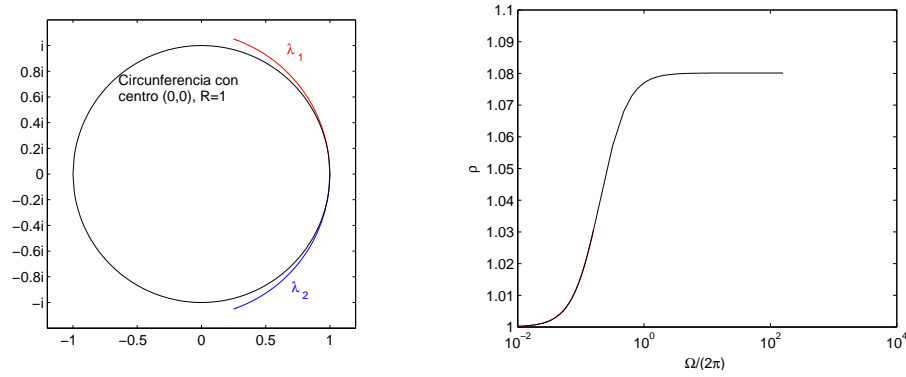


Figura 3.2: Autovalores y radio espectral del método de Newmark $\gamma = 0,4, \beta = 0,6$.

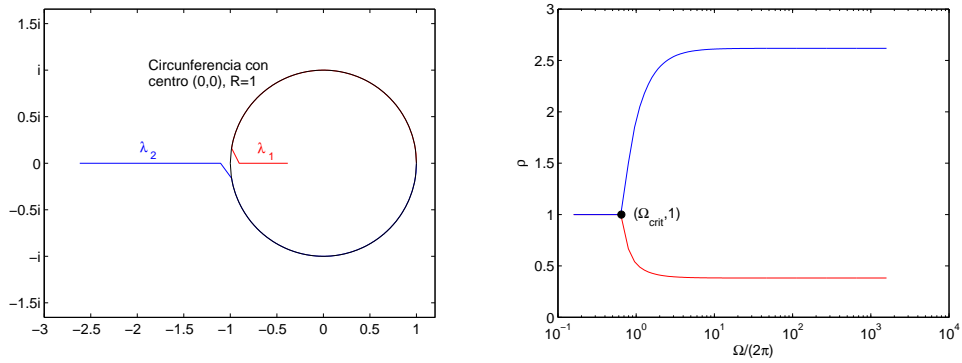


Figura 3.3: Autovalores y radio espectral del método de Newmark $\gamma = 0,5, \beta = 0,2$.

En el método de Newmark se consiguen autovalores reales cuando se anula la parte imaginaria de éstos. Es decir, cuando $K = 0$:

$$K = 0 \Rightarrow \Omega^2 (1 + 4\gamma + 4\gamma^2 - 16\beta) + \xi\Omega (-16\gamma + 8) + (16\xi^2 - 16) = 0 \quad (3.69)$$

Despejando Ω de la expresión (3.69) se obtiene el valor de Ω donde los autovalores se convierten en reales, es decir, el valor donde sucede la bifurcación:

$$\Omega_{bif} = \frac{\xi \frac{(\gamma - \frac{1}{2})}{2} \pm \left[\frac{(\gamma + \frac{1}{2})^2}{4} - \beta + \xi^2 (\beta - \frac{\gamma}{2}) \right]^{1/2}}{\frac{(\gamma + \frac{1}{2})^2}{4} - \beta} \quad (3.70)$$

Tras la bifurcación, los dos autovalores son reales; el módulo de uno de ellos crece y el del otro decrece. En consecuencia, el radio espectral crece.

Para calcular los valores de β, γ en los que la disipación es máxima, hacemos que la bifurcación se dé en $\Omega \rightarrow \infty$. Es decir, planteamos para qué valores de β, γ sucede

$\Omega_{bif} \rightarrow \infty$. Y esto sucede cuando el denominador de (3.70) es cero:

$$\beta = \frac{(\gamma + \frac{1}{2})^2}{4} \quad (3.71)$$

Todo lo que hemos explicado queda resumido en la Figura 3.4, donde también se define una zona estable y de disipación de frecuencias altas que se justifica a continuación.

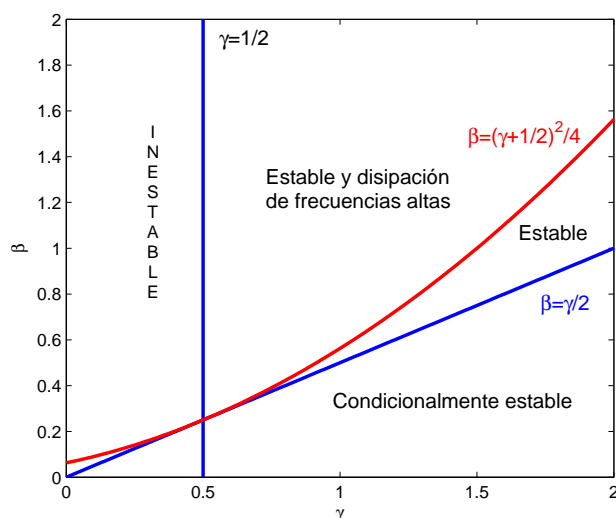


Figura 3.4: Estabilidad del método de Newmark.

En la Figura 3.5 se han dibujado los autovalores y el radio espectral de un método de Newmark en el que se da disipación de las altas frecuencias, $\gamma = 0,6$, $\beta = 0,3025$. En este caso, los autovalores son siempre complejos conjugados. Manteniendo el valor de $\gamma = 0,6$ y modificando levemente el valor del parámetro β ($\beta = 0,3$), tenemos un método de Newmark en el que no existe disipación de las frecuencias altas. En la Figura 3.6 se han dibujado los autovalores y el radio espectral para este segundo caso. Al principio los autovalores son complejos conjugados y en un instante la parte imaginaria de éstos se hace cero, es el punto donde se da la bifurcación. A partir de aquí los autovalores son reales, y en la recta real uno de ellos tiende hacia a la izquierda y el otro hacia la derecha.

En el caso del método de Newmark de orden 2 ($\gamma = 1/2$), el valor mínimo de β que retiene la estabilidad incondicional es $\beta = \gamma/2 = 1/4$. Además, $\beta = 1/4$ es el que consigue máxima disipación (3.71). En este caso, el método de Newmark que se tiene es el método trapezoidal y a pesar de que verifica la condición (3.71), el radio espectral cumple $\rho = 1$ con independencia del valor de las frecuencias (ver Figura 3.9).

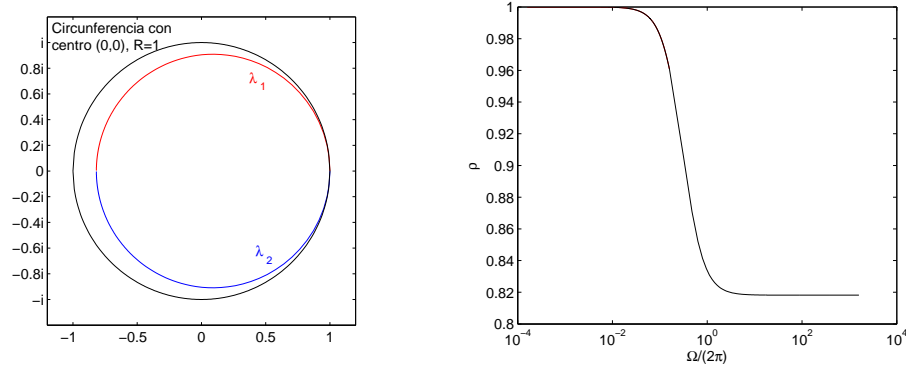


Figura 3.5: Autovalores y radio espectral del método de Newmark $\gamma = 0,6, \beta = 0,3025$.

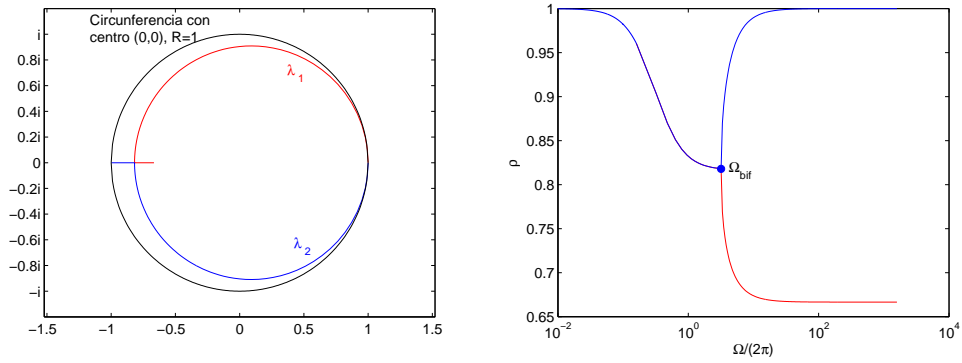


Figura 3.6: Autovalores y radio espectral del método de Newmark $\gamma = 0,6, \beta = 0,3$.

3.2.2. Método alfa

Considerando $\alpha_m = 0$ en el método alfa-generalizado, se obtiene el método α de Hilber-Hughes-Taylor, también conocido como método HHT- α [43]. Este método es una modificación al método de Newmark, con el objeto de conseguir disipación de las altas frecuencias manteniendo el orden de precisión 2 y las condiciones de estabilidad. En él se mantienen las primeras dos ecuaciones de (3.23) y la tercera ecuación adopta esta forma:

$$Ma_{n+1} + (1 - \alpha) Cv_{n+1} + \alpha Cv_n + (1 - \alpha) Kd_{n+1} + \alpha Kd_n = F(t_{n+1-\alpha}) \quad (3.72)$$

siendo: $t_{n+1-\alpha} = (1 - \alpha)t_{n+1} + \alpha t_n$

El orden de este método se puede deducir utilizando (3.48) y resulta de orden 2 cuando $\gamma = \frac{1}{2} + \alpha$.

Si además los parámetros se seleccionan de tal forma que $\alpha \in [0, \frac{1}{3}]$ y $\beta = \frac{(1+\alpha)^2}{4}$, el método HHT- α resulta incondicionalmente estable y se consigue disipación de frecuencias

altas. En las Figuras 3.7 y 3.8 se han dibujado los autovalores (imagen izquierda) y los radios espectrales (imagen derecha) de dos métodos HHT- α para $\alpha = 0,3$ y $\alpha = 0,05$ respectivamente. Dos de los autovalores son complejos conjugados, $\lambda_{1,2}$, y el tercero es real, λ_3 . El módulo del autovalor real es más pequeño que el módulo de los autovalores conjugados, $|\lambda_3| < |\lambda_{1,2}|$. Es lo que se denomina raíz espuria (*spurious root* en inglés), lo que significa *raíz no válida* del polinomio característico.

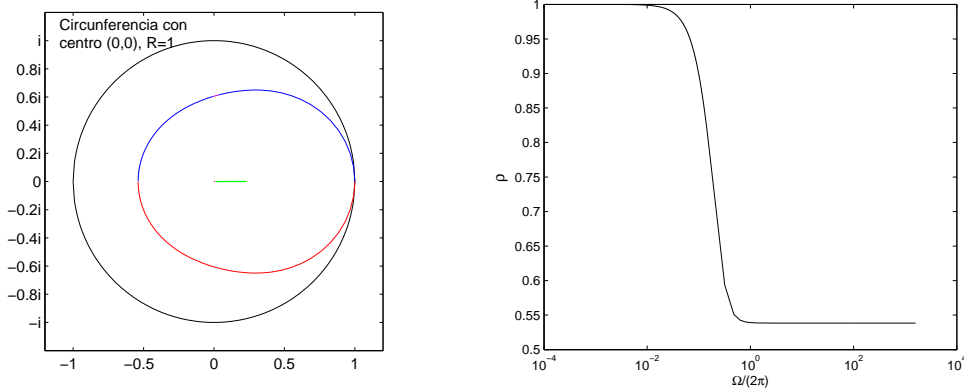


Figura 3.7: Autovalores y radio espectral del método HHT- $\alpha = 0,3$.

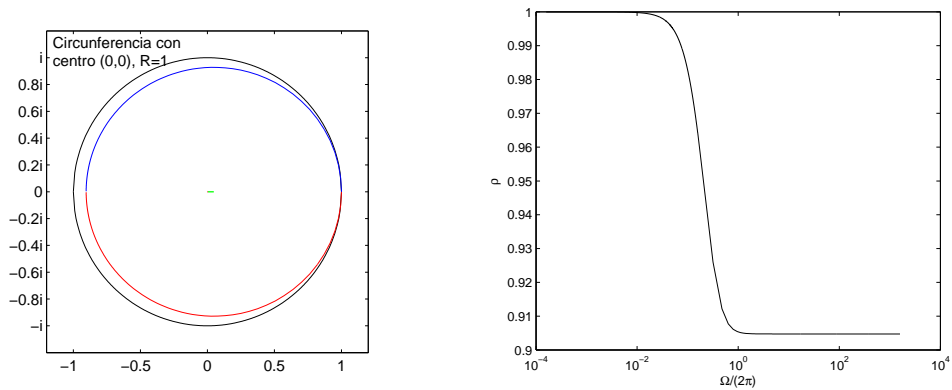


Figura 3.8: Autovalores y radio espectral del método HHT- $\alpha = 0,05$.

3.3. Método de Houbolt

El método de Houbolt viene dado por las siguientes expresiones:

$$\begin{cases} Ma_{n+1} + Cv_{n+1} + Kd_{n+1} = F_{n+1} \\ a_{n+1} = \frac{2d_{n+1} - 5d_n + 4d_{n-1} - d_{n-2}}{\Delta t^2} \\ v_{n+1} = \frac{11d_{n+1} - 18d_n + 9d_{n-1} - 2d_{n-2}}{6\Delta t} \end{cases} \quad (3.73)$$

Es un método de orden 2, incondicionalmente estable [31] y con máxima disipación de las frecuencias altas (ver Figura 3.9).

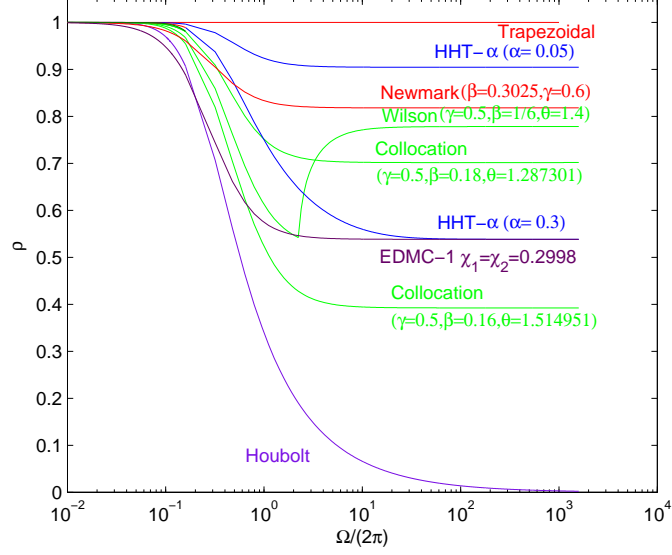


Figura 3.9: Radios espectrales de algunos métodos.

3.4. Método Collocation

El método Collocation viene dado por las siguientes ecuaciones:

$$\begin{cases} Ma_{n+\theta} + Cv_{n+\theta} + Kd_{n+\theta} = F_{n+\theta} \\ a_{n+\theta} = (1 - \theta) a_n + \theta a_{n+1} \\ F_{n+\theta} = (1 - \theta) F_n + \theta F_{n+1} \\ d_{n+\theta} = d_n + \theta \Delta t v_n + \frac{(\theta \Delta t)^2}{2} [(1 - 2\beta) a_n + 2\beta a_{n+\theta}] \\ v_{n+\theta} = v_n + \theta \Delta t [(1 - \gamma) a_n + \gamma a_{n+\theta}] \end{cases} \quad (3.74)$$

Si $\theta = 1$ el método se convierte en el método de Newmark y si $\beta = \frac{1}{6}$, $\gamma = \frac{1}{2}$ se obtiene el método de Wilson. Cuando $\gamma = \frac{1}{2}$ el método es mínimamente de orden 2, pudiéndose conseguir orden 3 con $\beta = \frac{1}{12} - \frac{\theta(\theta-1)}{2}$.

El método es de segundo orden e incondicionalmente estable si los parámetros γ , θ y β cumplen [42]:

$$\gamma = \frac{1}{2}, \quad \theta \geq 1, \quad \frac{\theta}{2(\theta+1)} \geq \beta \geq \frac{2\theta^2 - 1}{4(2\theta^3 - 1)} \quad (3.75)$$

Otros métodos de características similares fueron propuestos por Zienkiewicz [71] y analizados por Wood en [69]. Se trata de una familia de métodos en la que se utiliza la información de otro punto adicional que en los métodos de Newmark, HHT- α o alfa-generalizado y unas funciones de peso. De esta manera es posible conseguir orden 4 y estabilidad condicional.

En la Figura 3.9 se han dibujado los radios espectrales correspondientes a varios métodos que hemos visto en este capítulo.

3.5. Métodos que disipan la energía y conservan el momento

Los métodos descritos hasta este momento fueron desarrollados sobre todo para la integración temporal de problemas lineales. Aunque se trata de métodos con buenas características de estabilidad en el ámbito lineal, pueden dar lugar a inestabilidades en el ámbito no lineal. Estas inestabilidades generalmente son fruto del crecimiento incontrolado de la energía del sistema discreto y ésta ha sido la motivación del desarrollo de distintos algoritmos que conservan la energía y/o el momento angular. En [4, 5] se presentan los métodos *Energy Dissipative*, *Momentum Conserving*, EDMC-1 y EDMC-2, que funcionan bien en el ámbito no lineal. Son métodos que conservan el momento angular, disipan la energía y son capaces de eliminar las frecuencias altas.

El método EDMC-1 adaptado para la resolución del problema (3.14) viene dado por:

$$\begin{cases} M(v_{n+1} - v_n) + C(d_{n+1} - d_n) + Kh(\beta_1^+ d_{n+1} + \beta_1^- d_n) = hF \\ d_{n+1} = d_n + \beta_2^+ hv_{n+1} + \beta_2^- hv_n \end{cases} \quad (3.76)$$

siendo: $\begin{cases} \beta_1^+ = \frac{1}{2}(1 + \chi_1), & \beta_1^- = \frac{1}{2}(1 - \chi_1) \\ \beta_2^+ = \frac{1}{2}(1 + \chi_2), & \beta_2^- = \frac{1}{2}(1 - \chi_2) \end{cases}$ y χ_1, χ_2 parámetros reales.

Es un método de orden de precisión 1 y que consigue eliminar las altas frecuencias cuando $\chi_1 = \chi_2$. El método EDMC-1 que resulta al considerar $\chi_1 = \chi_2 = 0,299$ tiene el mismo ρ_∞ que el método HHT- $\alpha = 0,3$, ver Figura 3.9.

El método EDMC-2 adaptado para la resolución del problema (3.14) viene dado por las siguientes expresiones:

$$\begin{cases} M(v_{n+1} - v_n) + C(d_{n+1} - d_n) + \frac{1}{2}Kh(d_{n+1} + \tilde{d}_n) = hF \\ d_{n+1} = d_n + \frac{1}{2}h(v_{n+1} + \tilde{v}_n) \end{cases} \quad (3.77)$$

siendo \tilde{d}_n y \tilde{v}_n :

$$\begin{cases} M(\tilde{v}_n - v_n) + Kh\alpha(\tilde{d}_n - d_{n+1}) = hF \\ \tilde{d}_n = d_n + h\alpha(\tilde{v}_n - v_{n+1}) \end{cases} \quad (3.78)$$

y siendo α un parámetro real. Se trata un método de orden 2 con $\rho_\infty = 0$, pero donde es posible controlar el amortiguamiento algorítmico variando el valor del parámetro α .

3.6. Otros parámetros

Además del error local de truncamiento, existen otras medidas de precisión como son el amortiguamiento algorítmico ($\bar{\xi}$) y el error relativo en el periodo [47]. La solución exacta del problema SDOF homogéneo (3.11) continuo viene dado por:

$$d(t) = e^{-\xi\omega t} (d_0 \cos \omega_d t + c \sin \omega_d t) \quad (3.79)$$

siendo ω_d la frecuencia natural con amortiguamiento:

$$\begin{cases} c = \frac{v_0 + \xi\omega d_0}{\omega_d} \\ \omega_d = \omega \sqrt{1 - \xi^2} \end{cases} \quad (3.80)$$

Escribiremos de manera similar a (3.79) la solución discreta que se obtiene tras aplicar un método numérico:

$$d_n = e^{-\bar{\xi}\bar{\omega}t_n} (d_0 \cos \bar{\omega}_d t_n + \bar{c} \sin \bar{\omega}_d t_n) \quad (3.81)$$

donde \bar{c} es una constante a determinar y $\bar{\omega}_d = \bar{\omega} \sqrt{1 - \bar{\xi}^2}$.

El valor t_n de la expresión (3.81) se puede escribir como $t_n = h \cdot n$ siendo $h = \Delta t$. Por lo que (3.81) también se puede escribir como:

$$d_n = e^{-\bar{\xi}\bar{\Omega}n} (d_0 \cos \bar{\Omega}_d n + \bar{c} \sin \bar{\Omega}_d n) \quad (3.82)$$

siendo $\bar{\Omega} = \bar{\omega}h$, $\bar{\Omega}_d = \bar{\omega}_d h$ y $\bar{\Omega}_d = \bar{\Omega} \sqrt{1 - \bar{\xi}^2}$.

Comparando las expresiones (3.79) y (3.81), observamos que $\bar{\xi}$ y $\bar{\omega}$ son los homólogos o equivalentes algorítmicos de ξ y ω . Calcularemos los valores de los parámetros $\bar{\xi}$ y $\bar{\omega}$ en función de los autovalores de la matriz de amplificación. Los métodos numéricos que hemos visto en este capítulo tienen una matriz de amplificación 3×3 y por tanto tres autovalores: uno real y dos complejos conjugados. Esto significa que el problema SDOF homogéneo tiene soluciones del tipo $d_n = \sum_{j=1}^3 c_j \lambda_j^n$, siendo λ_j autovalor de la matriz de amplificación.

Cuando $|\lambda_3| < |\lambda_{1,2}|$, la solución del problema queda:

$$d_n \approx \sum_{j=1}^2 c_j \lambda_j^n \quad (3.83)$$

siendo $\lambda_{1,2}$ los autovalores conjugados de la matriz de amplificación: $\lambda_{1,2} = a(\Omega) \pm b(\Omega) i$.

Este es el caso de los métodos que hemos visto en este capítulo, los cuales tienen un autovalor espurio que cumple $|\lambda_3| < |\lambda_{1,2}|$. Desarrollaremos la expresión (3.83) operando en coordenadas polares:

$$\begin{cases} r = |\lambda_{1,2}| = \sqrt{a^2 + b^2} \\ \varphi_{1,2} = \arg(\lambda_{1,2}) = \pm \arctg\left(\frac{b}{a}\right) \end{cases} \quad (3.84)$$

Y teniendo en cuenta (3.84), se tiene:

$$d_n \approx \sum_{j=1}^2 c_j \lambda_j^n = \sum_{j=1}^2 c_j (r e^{i\varphi_j})^n = \sum_{j=1}^2 c_j r^n e^{i\varphi_j n} = r^n (\tilde{c}_1 \cos \varphi_1 n + \tilde{c}_2 \sin \varphi_1 n) \quad (3.85)$$

siendo $\tilde{c}_1 = \frac{c_1 + c_2}{2}$ y $\tilde{c}_2 = \frac{c_1 - c_2}{2i}$. Igualando las expresiones (3.82) y (3.85) tenemos:

$$\begin{cases} r^n = e^{-\bar{\xi}\bar{\Omega}n} \\ \varphi n = \bar{\Omega}_d n \end{cases} \Rightarrow \begin{cases} \sqrt{a^2 + b^2} = e^{-\bar{\xi}\bar{\Omega}} \\ \bar{\Omega}_d = \arct\left(\frac{b}{a}\right) \end{cases} \Rightarrow \begin{cases} \bar{\xi} = -\frac{\ln(a^2 + b^2)}{2\bar{\Omega}} \\ \bar{\Omega}\sqrt{1 - \bar{\xi}^2} = \arct\left(\frac{b}{a}\right) \end{cases} \quad (3.86)$$

El sistema (3.86) es un sistema de dos ecuaciones y dos incógnitas ($\bar{\xi}$ y $\bar{\Omega}$). Queremos llegar a expresiones que tienen la forma: $\begin{cases} \bar{\xi} = \bar{\xi}(\Omega) \\ \frac{T - \bar{T}}{T} = G(\Omega) \end{cases}$

Para ello, despejamos $\bar{\Omega}$ de la segunda expresión de (3.86), y sustituyéndolo en la primera expresión se tiene el valor de $\bar{\xi}$:

$$\bar{\xi} = \frac{\frac{\ln(a^2 + b^2)}{2 \cdot \arct\left(\frac{b}{a}\right)}}{\sqrt{1 + \left(\frac{\ln(a^2 + b^2)}{2 \cdot \arct\left(\frac{b}{a}\right)}\right)^2}} = \bar{\xi}(\Omega) \quad (3.87)$$

Una vez obtenido $\bar{\xi}$ calculamos $\bar{\Omega}$:

$$\bar{\Omega} = \arct\left(\frac{b}{a}\right) \sqrt{1 + \left(\frac{\ln(a^2 + b^2)}{2 \cdot \arct\left(\frac{b}{a}\right)}\right)^2} = \bar{\Omega}(\Omega) \quad (3.88)$$

Y utilizando la expresión (3.88) de $\bar{\Omega}$, se puede calcular el error relativo en el periodo:

$$\frac{\bar{T} - T}{T} = \frac{\Omega}{\bar{\Omega}} - 1 \Rightarrow \frac{\bar{T} - T}{T} = \frac{\omega}{\bar{\omega}} - 1 \quad (3.89)$$

En las Figuras 3.10 y 3.11 se han dibujado el amortiguamiento algorítmico y el error relativo en el periodo de algunos métodos.

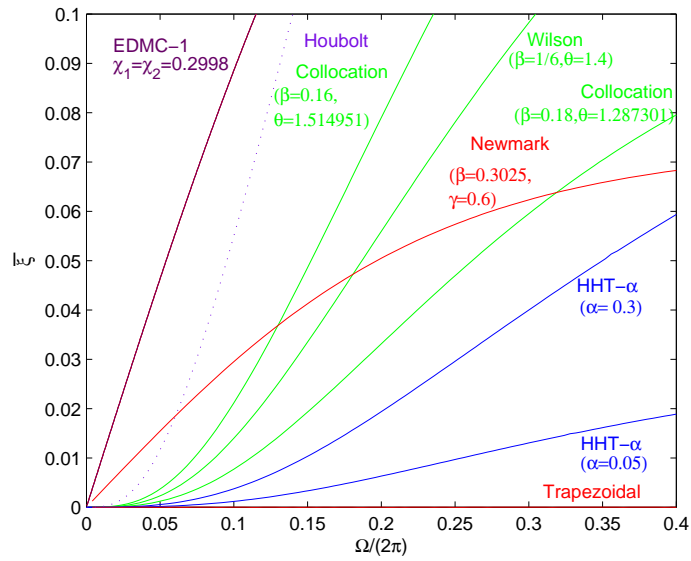


Figura 3.10: Amortiguamiento algorítmico de algunos métodos.

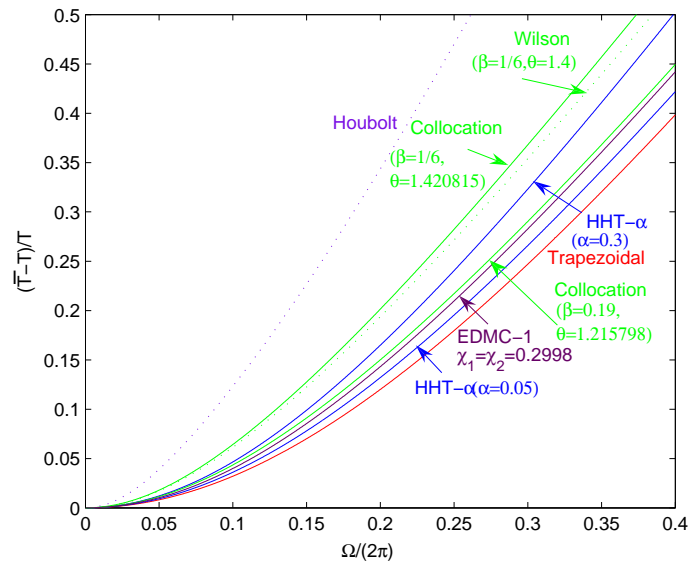


Figura 3.11: Error relativo en el periodo de algunos métodos.

Capítulo 4

ode45, ode15s

Debe haber un mundo ideal, una especie de paraíso matemático donde todo sucede como en los libros de texto.

BERTRAND RUSSELL

Matlab tiene implementados distintos algoritmos o funciones que permiten hallar numéricamente la solución de EDOs de valor inicial [2, 32, 60]. Dos de estas funciones son la *ode45* y la *ode15s*, ambas de paso variable. La *ode45* se caracteriza por ser apta para problemas no rígidos y la *ode15s* para los rígidos [1], aunque esta última no haya funcionado según lo esperado en la EDP onda del ejemplo 2 estudiado en el Capítulo 1. En este capítulo intentaremos dar respuesta a este hecho, estudiando los métodos numéricos en los que se basan las funciones *ode45* y *ode15s*, los estimadores del error que utilizan, en función de qué cambian de tamaño de paso, etc.

4.1. ode45

En el Capítulo 2 ya se ha dicho que la *ode45* está basada en el método Runge-Kutta anidado DOPRI(5,4) [23], cuya tabla Butcher viene dada por 2.22. El vector \mathbf{b}^T hace que el método sea de orden 4, mientras que con el vector $\hat{\mathbf{b}}^T$ el método es de orden 5. La *ode45* avanza con el resultado obtenido con el método de orden 5, pero precisa calcular los resultados de ambos órdenes para poder calcular la estimación del error.

■ Resultado obtenido con el método DOPRI(5,4) orden 5: $\hat{y}_{n+1} = y_n + h \sum_{i=1}^s \hat{b}_i k_i$ (4.1)

■ Resultado obtenido con el método DOPRI(5,4) orden 4: $y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i$ (4.2)

siendo $k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j)$, $i = 1, 2, \dots, s$.

4.1.1. Estimación del error en la ode45

El control del error global se realiza mediante el control de una estimación del error local [59, 65]. La estimación del error local que se utiliza frecuentemente es la diferencia entre dos soluciones numéricas calculadas con métodos de orden consecutivo. El objetivo suele ser que esta diferencia sea menor que una tolerancia definida.

La *ode45* utiliza un estimador del error local que es la diferencia entre el valor numérico obtenido con el método de orden 5 (\hat{y}_{n+1}) y el obtenido con el de orden 4 (y_{n+1}), y como se demuestra en [61] esta diferencia coincide con el error local de truncamiento del método de orden 4:

$$\begin{aligned} est &= \hat{y}_{n+1} - y_{n+1} = \\ &= LTE_4 + O(h^{5+1}) \end{aligned} \quad (4.3)$$

siendo $LTE_4 = O(h^{4+1})$ el error local de truncamiento del método de orden 4 y el término dominante en (4.3). Sustituyendo las expresiones de \hat{y}_{n+1} e y_{n+1} , (4.1) y (4.2) respectivamente, en (4.3) el estimador queda:

$$est = \hat{y}_{n+1} - y_{n+1} = h \sum_{i=1}^s (\hat{b}_i - b_i) k_i \quad (4.4)$$

Una ventaja de los métodos Runge-Kutta anidados es que con pocas operaciones se consiguen los resultados correspondientes a los métodos de órdenes $(p, p+1)$ y esto hace que un estimador del error local basado en la diferencia de ambos sea *barato* computacionalmente.

La tolerancia se define de esta forma:

$$tol^i = Atol^i + |y^i| Rtol, \quad i = 1, 2, \dots, m \quad (4.5)$$

siendo **Atol** y *Rtol* especificados por el usuario y m la dimensión del sistema de EDOs $y' = f(t, y)$ que estamos resolviendo.

La tolerancia será absoluta si $Rtol = 0$, relativa si $Atol^i = 0$ o una combinación de ambas. La tolerancia relativa *Rtol* es un escalar y la absoluta **Atol** un vector, que se especifica para cada componente del vector solución del sistema de EDOs.

Una vez especificados los dos datos **Atol** y *Rtol*, podemos calcular y_{min}^i , que es un valor mínimo de y significativo en cada componente:

$$Atol^i = Rtol \cdot y_{min}^i \quad (4.6)$$

De esta manera, la expresión (4.5) se escribe:

$$tol^i = (y_{min}^i + |y^i|) Rtol, \quad i = 1, 2, \dots, m \quad (4.7)$$

debiéndose cumplir la condición de tolerancia:

$$\|est\| \leq \|tol\| \quad (4.8)$$

La norma utilizada por defecto está basada en la componente máxima:

$$\|est\| = \|\hat{y}_{n+1} - y_{n+1}\| = \max_{i=1,2,\dots,m} |\hat{y}_{n+1}^i - y_{n+1}^i| \quad (4.9)$$

El hecho de definir el vector tolerancia **tol** componente a componente permite escribir la condición (4.8) de la siguiente manera:

$$\max_{i=1,2,\dots,m} \left(\frac{|\hat{y}_{n+1}^i - y_{n+1}^i|}{y_{min}^i + |y_{n+1}^i|} \right) \leq Rtol \quad (4.10)$$

Las expresiones (4.8) y (4.10) son equivalentes. En (4.10) se considera la componente máxima normalizada, cuyo valor absoluto tiene que ser menor que la tolerancia relativa *Rtol*. En lo que sigue denotaremos por $\|est\|$ el valor absoluto de la componente máxima normalizada que debe cumplir:

$$\|est\| \leq Rtol \quad (4.11)$$

Si denotamos $E_i = (\hat{b}_i - b_i)$ y $f_i = k_i$, la expresión (4.4) queda:

$$est = h \sum_{i=1}^s E_i \cdot f_i = h \cdot E \cdot f \quad (4.12)$$

siendo: $E = (E_1, E_2, \dots, E_s)$ y $f = \begin{pmatrix} f_1^1 & f_1^2 & \dots & f_1^m \\ f_2^1 & f_2^2 & \dots & f_2^m \\ \vdots & \dots & \ddots & \vdots \\ f_s^1 & f_s^2 & \dots & f_s^m \end{pmatrix}$. Ver Sección 2.3.4.

Y teniendo en cuenta (4.12), la expresión (4.10) queda:

$$h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{y_{min}^i + |y_{n+1}^i|} \right) \leq Rtol \Rightarrow \|est\| \leq Rtol \quad (4.13)$$

Observación. La *ode45* no utiliza exactamente el valor $\|est\|$ de la expresión (4.13) para calcular la norma de la estimación del error, sino que utiliza otra que la vamos a denotar con un subíndice M , $\|est_M\|$, y que viene dada por:

$$\|est_M\| = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \right) \quad (4.14)$$

y que debe cumplir $\|est_M\| \leq Rtol$.

Se puede verificar que la utilización de (4.14) es correcta, ya que ésta bien es mayor o igual que la expresión del estimador dada por (4.13):

$$\|est\| \leq \|est_M\| \quad (4.15)$$

o bien, ambos estimadores, (4.13) y (4.14), están acotados por los mismos dos valores:

$$\begin{cases} h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_n^i| + y_{min}^i} \right) \leq \|est\| \leq h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i|} \right) \\ h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_n^i| + y_{min}^i} \right) \leq \|est_M\| \leq h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i|} \right) \end{cases} \quad (4.16)$$

Demostración. Demostraremos las desigualdades (4.15) y (4.16). Y lo haremos para los tres posibles casos que se pueden dar:

- Caso $\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i) = |y_{n+1}^i|$. En este caso se tiene:

$$\|est_M\| = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \right) = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i|} \right)$$

Como $|y_{n+1}^i| \leq |y_n^i| + y_{min}^i$, cogiendo las inversas tenemos:

$$\underbrace{h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i| + y_{min}^i} \right)}_{\|est\|} \leq \underbrace{h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i|} \right)}_{\|est_M\|} \quad (4.17)$$

Por lo que queda demostrada la relación (4.15).

- Caso $\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i) = y_{min}^i$. En este caso se tiene:

$$\|est_M\| = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \right) = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{y_{min}^i} \right)$$

Como $y_{min}^i \leq |y_{n+1}^i| + y_{min}^i$, cogiendo las inversas tenemos:

$$\underbrace{h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i| + y_{min}^i} \right)}_{\|est\|} \leq \underbrace{h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{y_{min}^i} \right)}_{\|est_M\|} \quad (4.18)$$

Por lo que también en este caso queda demostrada la relación (4.15).

- En tercer lugar consideramos $\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i) = |y_n^i|$. En este caso el valor del estimador de Matlab es este:

$$\|est_M\| = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \right) = h \cdot \max_{i=1,2,\dots,m} \left(\frac{|(E \cdot f)^i|}{|y_n^i|} \right)$$

Demostraremos que en este caso $\|est_M\|$ está acotado:

Como $|y_{n+1}^i| \leq |y_n^i| \leq |y_n^i| + y_{min}^i$, cogiendo las inversas se cumple:

$$h \cdot \max_i \left(\frac{|(E \cdot f)^i|}{|y_n^i| + y_{min}^i} \right) \leq \underbrace{h \cdot \max_i \left(\frac{|(E \cdot f)^i|}{|y_n^i|} \right)}_{\|est_M\|} \leq h \cdot \max_i \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i|} \right) \quad (4.19)$$

Por lo que $\|est_M\|$ queda acotado como se ha dicho en (4.16).

También $\|est\|$ está acotado por los mismos valores que $\|est_M\|$. Como $|y_{n+1}^i| \leq |y_{n+1}^i| + y_{min}^i \leq |y_n^i| + y_{min}^i$, cogiendo las inversas se cumple:

$$h \cdot \max_i \left(\frac{|(E \cdot f)^i|}{|y_n^i| + y_{min}^i} \right) \leq \underbrace{h \cdot \max_i \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i| + y_{min}^i} \right)}_{\|est\|} \leq h \cdot \max_i \left(\frac{|(E \cdot f)^i|}{|y_{n+1}^i|} \right) \quad (4.20)$$

Y también $\|est\|$ queda acotado como se ha dicho en (4.16).

De esta manera han quedado demostradas las desigualdades (4.15) y (4.16). \square

4.1.2. Estimación del error local vs. regiones de estabilidad ode45

Si quisiéramos controlar totalmente el error global, también deberíamos analizar el factor de amplificación A (relacionado éste con la región de estabilidad), asegurando que su radio espectral es menor o igual que la unidad. La *ode45* sólo calcula la estimación del error local y no hace ningún cálculo adicional para que el factor de amplificación sea inferior a la unidad. Es decir, sólo controla:

$$\|est_M\| \leq Rtol \quad (4.21)$$

Hemos estudiado si existe alguna relación entre la estimación del error local de la *ode45* y las regiones de estabilidad del método DOPRI(5,4). Cuando aplicamos el método DOPRI(5,4) a la ecuación de test $y' = \lambda y$, se obtienen los valores de \hat{y}_{n+1} e y_{n+1} en función de los polinomios característicos correspondientes a cada orden y en función del valor y_n :

- Orden 5: $\hat{y}_{n+1} = R_5(\hat{h}) \cdot y_n$
- Orden 4: $y_{n+1} = R_4(\hat{h}) \cdot y_n$

donde $\hat{h} = h\lambda$, y R_5 y R_4 quedaron definidos en el Capítulo 2 Sección 2.3.4. Por tanto, en este caso, la estimación del error (4.4) queda:

$$est = \hat{y}_{n+1} - y_{n+1} = \left(R_5(\hat{h}) - R_4(\hat{h}) \right) \cdot y_n \quad (4.22)$$

Y la condición (4.21) resulta:

$$\|est_M\| = \max_{i=1,2,\dots,m} \left(\frac{|y_n^i| |R_5 - R_4|}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \right) \leq Rtol \quad (4.23)$$

Como se cumple $|y_n^i| \leq \max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)$, se verifica:

$$\frac{1}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \leq \frac{1}{|y_n^i|}, \quad \text{para } |y_n^i| > 0 \quad (4.24)$$

Y teniendo en cuenta (4.24) el estimador $\|est_M\|$ verifica:

$$\begin{aligned} \|est_M\| &= \max_{i=1,2,\dots,m} \left(\frac{|y_n^i| |R_5 - R_4|}{\max(|y_n^i|, |y_{n+1}^i|, y_{min}^i)} \right) \\ &\leq \max_{i=1,2,\dots,m} \left(\frac{|y_n^i| |R_5 - R_4|}{|y_n^i|} \right) = |R_5 - R_4| \end{aligned} \quad (4.25)$$

Por lo que se tiene:

$$\|est_M\| \leq |R_5 - R_4| \quad (4.26)$$

Y si $|R_5 - R_4| \leq Rtol$ entonces se cumplirá:

$$\|est_M\| \leq Rtol \quad (4.27)$$

La cuestión que ahora nos planteamos es si existe alguna relación entre $|R_5 - R_4|$ y la región de estabilidad de orden 5 del método DOPRI(5,4).

Para responder a esta pregunta hemos dibujado en el plano complejo el lugar geométrico de los puntos \hat{h} que cumplen $|R_5 - R_4| = Rtol$. En la Figura 4.1 se pueden ver las curvas correspondientes a distintos valores de $Rtol$. La parte interior de estas curvas es donde se cumple $|R_5 - R_4| \leq Rtol$.

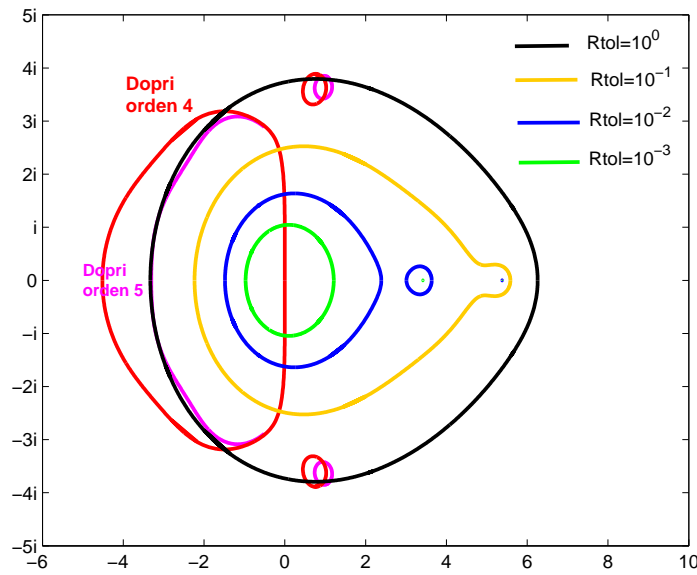


Figura 4.1: Regiones $|R_5 - R_4| \leq Rtol$ del método DOPRI(5,4).

Se puede apreciar que cuanto menor es la constante $Rtol$ más próxima al origen resulta la curva $|R_5 - R_4| = Rtol$, siendo menor el área $|R_5 - R_4| \leq Rtol$. En esta figura se han superpuesto las curvas de las regiones de estabilidad DOPRI(5,4). Se aprecia que en el semiplano negativo, las regiones $|R_5 - R_4| \leq Rtol$ caen dentro de la región de estabilidad del método DOPRI(5,4) de orden 5, para valores de $Rtol$ menores que 10^{-2} . Se puede precisar más diciendo que existe una constante $a \in (10^{-2}, 10^{-1})$ para la cual la región $|R_5 - R_4| \leq a$ del semiplano negativo, cae dentro de la región de estabilidad del método DOPRI(5,4) de orden 5. Y a partir de este valor a , no toda la región $|R_5 - R_4| \leq a$ cae dentro de la región de estabilidad del método de orden 5, siendo estas zonas de inestabilidad *pequeñas*. Así, se cumple que para valores de $Rtol \leq a$:

$$\hat{h} \in \{\text{Región donde } |R_5 - R_4| \leq Rtol\} \Rightarrow \hat{h} \in \{\text{Región de estab. DOPRI(5,4) orden 5}\} \quad (4.28)$$

$\|est_M\|$ cumple la desigualdad (4.26), pero la desigualdad que controla la *ode45* es (4.27), por lo que no existe un control estricto del factor de amplificación (es decir, control de la posición de \hat{h} en la región de estabilidad) en el código. Se realizaría un control estricto del error global haciendo $|R_5 - R_4| \leq Rtol$, pero esto implicaría un gran coste computacional.

4.1.3. Control del tamaño de paso

Tras avanzar de t_{n-1} a t_n con tamaño h , el estimador del error local DOPRI(5,4) (4.3) es éste:

$$est_n = LTE_4 = y(t_n + h) - y_{n+1}^* = h^5 \phi(t_n) + O(h^6) \quad (4.29)$$

donde $\phi(t_n)$ es la función que multiplica a h^{p+1} en la expresión (2.81) e y_{n+1}^* se ha calculado haciendo la asunción de localización (2.5).

Supongamos que hemos dado un paso bueno, $\|est_M\| \leq Rtol$, y que queremos calcular el tamaño del siguiente paso (de t_n a t_{n+1}). Éste se calcula multiplicando el tamaño de paso actual h por una constante a la que vamos a llamar σ . El siguiente tamaño de paso será $\sigma \cdot h$ y el error local que se cometerá en ese paso será [61]:

$$est_{n+1} = (\sigma \cdot h)^5 \phi(t_{n+1}) + O((\sigma h)^6) \quad (4.30)$$

Utilizando desarrollos de Taylor se tiene:

$$\phi(t_{n+1}) = \phi(t_n) + h\phi'(t_n) + O(h^2) \quad (4.31)$$

Por lo que la expresión (4.30) queda:

$$est_{n+1} = (\sigma \cdot h)^5 \phi(t_{n+1}) + O((\sigma h)^6) = \sigma^5 h^5 \phi(t_n) + O(h^6) = \sigma^5 est_n + O(h^6) \quad (4.32)$$

El mayor tamaño de paso que pasará el test del error, corresponde a elegir σ de tal forma que $\|est_{n+1}\| \approx Rtol$:

$$\|est_{n+1}\| \approx Rtol \Rightarrow \sigma^5 \|est_n\| \approx Rtol \quad (4.33)$$

Y el tamaño de paso a dar será:

$$\sigma \cdot h = h \left(\frac{Rtol}{\|est_n\|} \right)^{1/5} \quad (4.34)$$

Cuando el paso dado ha resultado fallido, también se utiliza (4.34) para calcular con qué tamaño de paso repetiremos el paso de t_{n-1} a t_n .

Se utilizan constantes como 0,8 o 0,9 para garantizar que el paso que se va a dar va a ser bueno [61]. En concreto, en el caso de la *ode45* el tamaño de paso (4.34) se multiplica por la constante 0,8. A continuación damos más detalles sobre el cálculo del tamaño de paso. Y a partir de ahora denotaremos por $est = est_n$ la estimación del error en el paso que acabamos de dar y denotaremos simplemente por $\|est\|$ la norma de la estimación del error que calcula Matlab.

4.1.3.1. Primer tamaño de paso

Durante muchos años, al código se le ha proporcionado el primer tamaño de paso [38]. Tanto la *ode45* como la *ode15s* tienen esta opción mediante *htry*. De todas formas, un primer tamaño de paso que diese lugar a un paso fallido sería reparado fácilmente mediante el control del tamaño de paso del código. En el caso de la *ode45*, se define $absh = \min(hmax, htspan)$ para concretar el primer tamaño de paso, siendo *htspan* la longitud del intervalo de integración y *hmax* el valor máximo para el tamaño de paso. También se definen *hmin* (menor tamaño de paso) y un factor *rh* de la siguiente manera:

$$hmin = 16 \cdot eps(t) = \begin{cases} 16 \cdot |t| \cdot eps, & t \neq 0 \\ 16 \cdot 4,9407 \cdot 10^{-324}, & t = 0 \end{cases} \quad (4.35)$$

$$rh = \frac{1,25 \cdot \|y'_0\|}{\|y_0\| \cdot Rtol^{1/5}} \quad (4.36)$$

siendo: $eps = 2,2204 \cdot 10^{-16}$. Una vez definidos los valores de *absh* y *rh*, se controla el producto de ambos y se redefine el valor de *absh*:

$$absh = \begin{cases} absh, & absh \cdot rh \leq 1 \\ \frac{1}{rh}, & absh \cdot rh > 1 \end{cases} \quad (4.37)$$

Finalmente, se asegura que el *absh* obtenido mediante (4.37) es mayor que *hmin* haciendo: $absh = \max(absh, hmin)$. Y el primer tamaño de paso será *absh*.

Cuando al código se le suministra el primer tamaño de paso *htry*, éste únicamente se asegura de que el valor *htry* se encuentra en el intervalo [*hmin*, *hmax*] haciendo: $absh = \min(hmax, \max(hmin, htry))$.

4.1.3.2. Nuevo tamaño de paso tras h bueno

Tanto cuando el paso ha sido bueno como malo, el siguiente tamaño de paso se calcula utilizando (4.34). De (4.34) se deduce que el siguiente tamaño de paso, h_{nuevo} , depende del valor que haya tenido el estimador del error en el paso actual, del tamaño de paso actual *h* y de la tolerancia relativa *Rtol* que estamos utilizando. Cuando el paso dado ha sido bueno, es decir $\|est\| \leq Rtol$, la expresión que utiliza la *ode45* para calcular el nuevo tamaño de paso es:

$$h_{nuevo} = \begin{cases} 5 \cdot h, & \|est\| \leq Rtol \cdot 1,0486 \cdot 10^{-4} \\ 0,8 \cdot h \cdot \left(\frac{Rtol}{\|est\|}\right)^{1/5}, & Rtol \cdot 1,0486 \cdot 10^{-4} < \|est\| \leq Rtol \end{cases} \quad (4.38)$$

Es decir, para valores del estimador menores o iguales que $Rtol \cdot 1,0486 \cdot 10^{-4}$ el nuevo tamaño de paso es 5 veces el tamaño de paso actual y para valores del estimador

comprendidos en el intervalo $(Rtol \cdot 1,0486 \cdot 10^{-4}, Rtol]$ se utiliza la segunda opción de la expresión (4.38), obteniéndose $h_{nuevo} = 0,8 \cdot h$ cuando la estimación del error coincide con la tolerancia, $\|est\| = Rtol$.

Pero la *ode45* no utiliza de forma explícita la expresión (4.38), sino que define una variable llamada *temp* que depende de la estimación del error en el paso actual (*est*) y de la tolerancia relativa (*Rtol*):

$$temp = 1,25 \left(\frac{\|est\|}{Rtol} \right)^{1/5} \quad (4.39)$$

La *ode45* define el siguiente tamaño de paso, h_{nuevo} , en función de la variable *temp* y del tamaño de paso actual h , distinguiéndose dos casos:

$$h_{nuevo} = \begin{cases} 5 \cdot h, & temp \leq 0,2 \\ \frac{h}{temp}, & temp > 0,2 \end{cases} \quad (4.40)$$

Hemos calculado la relación que tiene que cumplir la estimación en función de *Rtol* para que la variable $temp = 0,2$, resultando $\|est\| = Rtol \cdot 1,0486 \cdot 10^{-4}$. Así, la expresión (4.40) que tenemos en función de *temp*, se puede escribir en función de la estimación del error $\|est\|$ como lo hemos hecho en (4.38).

Como estamos analizando el caso en el que el paso dado ha sido bueno, es decir $\|est\| \in [0, Rtol]$, el parámetro *temp* cumple $temp \in [0, 5/4]$ y teniendo en cuenta (4.38), el siguiente tamaño de paso cumple la relación:

$$0,8 \cdot h \leq h_{nuevo} \leq 5 \cdot h \quad (4.41)$$

El nuevo tamaño de paso también depende de la tolerancia relativa *Rtol*. Así, para una misma estimación del error, el nuevo tamaño de paso será mayor cuanto mayor sea la tolerancia utilizada.

4.1.3.3. Nuevo tamaño de paso tras h fallido

Si el paso que hemos dado no cumple el requisito de tolerancia, es decir $\|est\| > Rtol$, hay que repetir el paso con un tamaño de paso menor que el anterior. La expresión que utiliza la *ode45* para repetir un paso es:

$$h_{nuevo} = \begin{cases} 0,8 \cdot h \cdot \left(\frac{Rtol}{\|est\|} \right)^{1/5}, & Rtol < \|est\| \leq Rtol \cdot 32768 \\ h \cdot 0,1, & \|est\| \geq Rtol \cdot 32768 \end{cases} \quad (4.42)$$

Si éste de nuevo resulta fallido, los consecutivos pasos se calculan dividiendo entre 2 el valor del tamaño de paso actual h hasta dar con un tamaño de paso que cumpla el requisito $\|est\| \leq Rtol$.

Para hacer esto las expresiones que utiliza la *ode45* son estas:

$$h_{nuevo} = \begin{cases} \max\left(hmin, \max\left(h \cdot 0, 1, \frac{h}{temp}\right)\right), & \text{primer intento tras h fallido} \\ 0, 5 \cdot h, & \text{a partir del 2º intento} \end{cases} \quad (4.43)$$

donde *temp* viene dado por (4.39). La primera expresión de (4.43) es con la que se prueba tras hacer un fallo. Si esta repetición resulta fallida se utiliza la segunda expresión de (4.43).

La igualdad $0, 1 \cdot h = \frac{h}{temp}$ se cumple cuando $\|est\| = Rtol \cdot 32768$. De esta manera, partiendo de (4.43) se llega a la expresión (4.42).

Juntando las expresiones (4.38) y (4.42), se puede escribir el tamaño de paso nuevo tanto para los pasos buenos como para los fallidos en función de la estimación del error, tratándose de una función continua:

$$h_{nuevo} = \begin{cases} 5 \cdot h, & \|est\| \leq Rtol \cdot 1,0486 \cdot 10^{-4} \\ \frac{h}{temp}, & Rtol \cdot 1,0486 \cdot 10^{-4} < \|est\| \leq Rtol \\ \frac{h}{temp}, & Rtol < \|est\| \leq Rtol \cdot 32768 \\ h \cdot 0, 1, & \|est\| \geq Rtol \cdot 32768 \end{cases} \quad (4.44)$$

Las dos últimas expresiones de (4.44) se sustituyen por $h/2$ cuando estamos probando por segunda vez o posteriores tras un paso fallido. Las constantes 0, 1, 0, 2 y 5 son las que tiene implementadas la *ode45* para calcular el siguiente tamaño de paso. Si en lugar de estas utilizáramos otras, la expresión (4.44) obviamente cambiaría.

4.2. ode15s

La *ode15s* está basada en los métodos BDF [30] y además de éstos también tiene implementados los métodos NDF (ver Sección 2.2.5). Es con estos últimos con los que resuelve por defecto. La “s” del final indica que la *ode15s* sirve para resolver ecuaciones diferenciales stiff (rígidas) [61]. Es una función de orden variable ($k = 1, 2, \dots, 5$), que empieza trabajando en orden $k = 1$ y al que se le puede proporcionar el orden máximo al que queremos que resuelva.

4.2.1. Aspectos computacionales ode15s

Recordamos las expresiones de los métodos BDF y NDF dadas por (2.40) y (2.59) respectivamente:

- Expresión de los métodos BDF: $\sum_{j=1}^k \nabla^j y_{n+k} = h f_{n+k}$.

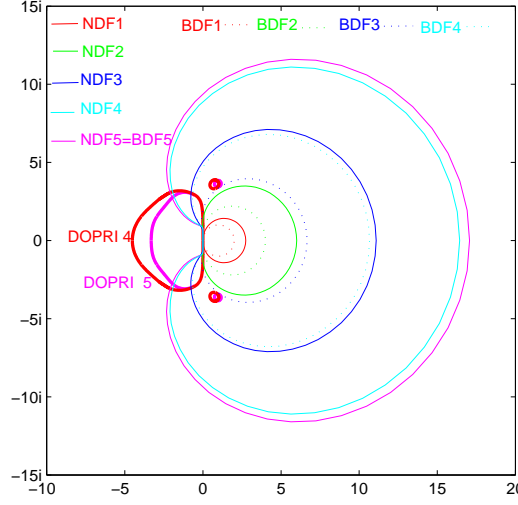


Figura 4.2: Regiones de estabilidad DOPRI(5,4), BDF y NDF.

- Expresión de los métodos NDF: $\sum_{j=1}^k \frac{1}{j} \nabla^j y_{n+k} = hf_{n+k} + \kappa \gamma_k \nabla^{k+1} y_{n+k}$.

En [62] se da una forma alternativa común para escribir la parte izquierda de las expresiones de estos dos métodos:

$$\sum_{j=1}^k \frac{1}{j} \nabla^j y_{n+k} = \gamma_k \left(y_{n+k} - y_{n+k}^{(0)} \right) + \sum_{j=1}^k \gamma_j \nabla^j y_{n+k-1} \quad (4.45)$$

$$\text{siendo: } \begin{cases} \gamma_j = \sum_{l=1}^j \frac{1}{l} \\ y_{n+k}^{(0)} = \sum_{j=0}^k \nabla^j y_{n+k-1} = \nabla^0 y_{n+k-1} + \nabla y_{n+k-1} + \dots + \nabla^k y_{n+k-1} \\ y_{n+k} - y_{n+k}^{(0)} = \nabla^{k+1} y_{n+k} \end{cases} \quad (4.46)$$

Teniendo en cuenta (4.45), los métodos BDF y NDF se escriben de esta manera:

$$(1 - \kappa) \gamma_k \left(y_{n+k} - y_{n+k}^{(0)} \right) + \sum_{j=1}^k \gamma_j \nabla^j y_{n+k-1} = hf_{n+k} \quad (4.47)$$

siendo $\kappa = 0$ para las BDF, y en el caso de las NDF se han recogido en la Tabla 2.5 los valores κ .

Dividiendo la expresión (4.47) entre la constante $(1 - \kappa)\gamma_k$ se obtiene:

$$\left(y_{n+k} - y_{n+k}^{(0)} \right) + \frac{\sum_{j=1}^k \gamma_j \nabla^j y_{n+k-1}}{(1 - \kappa) \gamma_k} = \frac{hf(t_{n+k}, y_{n+k})}{(1 - \kappa) \gamma_k} \quad (4.48)$$

La expresión (4.48) es una ecuación no lineal en y_{n+k} que se puede resolver utilizando el método de Newton. Definimos $d = y_{n+k} - y_{n+k}^{(0)}$, de donde se tiene $y_{n+k} = d + y_{n+k}^{(0)}$. Sustituyendo estas dos igualdades en (4.48) se obtiene una ecuación no lineal en d :

$$d + \frac{\sum_{j=1}^k \gamma_j \nabla^j y_{n+k-1}}{(1-\kappa)\gamma_k} = \frac{hf(t_{n+k}, d + y_{n+k}^{(0)})}{(1-\kappa)\gamma_k} \quad (4.49)$$

Simplificamos la notación de (4.49) utilizando una variable ψ que sólo depende de los valores anteriores:

$$\psi = \frac{1}{(1-\kappa)\gamma_k} \sum_{j=1}^k \gamma_j \nabla^j y_{n+k-1} \quad (4.50)$$

Y llegamos a la siguiente ecuación:

$$d + \psi - \frac{hf(t_{n+k}, d + y_{n+k}^{(0)})}{(1-\kappa)\gamma_k} = 0 \quad (4.51)$$

Para hallar la solución de una ecuación del tipo $f(x) = 0$ por el método de Newton, se parte de un valor $x^{(0)}$ y se itera hasta que $f(x^i + \Delta x) < tol$. Así, para la iteración i se tiene:

$$\Delta x^i = -\frac{f'(x^i)}{f(x^i)} \quad (4.52)$$

Cuando se trata de una función de varias variables y se quiere hallar $R(\mathbf{x}) = 0$, la expresión (4.52) se convierte en:

$$\Delta \mathbf{x}^i = -J^{-1}(\mathbf{x}^i)R(\mathbf{x}^i) \quad (4.53)$$

siendo: $J = \frac{\partial R}{\partial \mathbf{x}}$.

Para el sistema de ecuaciones dado por (4.51) se tiene:

$$\begin{cases} R(d^i) = d^i + \psi - \frac{hf(t_{n+k}, d^i + y_{n+k}^{(0)})}{(1-\kappa)\gamma_k} \\ J(d^i) = \frac{\partial R(d^i)}{\partial d} = I - \frac{h}{(1-\kappa)\gamma_k} \cdot \frac{\partial f(t_{n+k}, d^i + y_{n+k}^{(0)})}{\partial d} \end{cases} \quad (4.54)$$

Y aplicando (4.53) se obtiene el valor de $\Delta^{(i)}$:

$$\Delta^{(i)} = -J^{-1}(d^i)R(d^i) \quad (4.55)$$

Una vez calculado $\Delta^{(i)}$, se actualiza $d^{(i+1)}$: $d^{(i+1)} = d^{(i)} + \Delta^{(i)}$. El último valor de $d^{(i+1)}$ que se obtiene corresponde a la diferencia hacia atrás de orden $(k+1)$. Si este último valor viene dado por $d^{(f+1)}$, al finalizar la iteración se tiene el valor de y_{n+k} que buscábamos:

$$y_{n+k} = y_{n+k}^{(0)} + d^{(f+1)}$$

En la *ode15s*, hay que destacar el hecho de que el jacobiano de la función f ($J = \frac{\partial f}{\partial y}$), no se calcula ni en cada iteración, ni en cada paso. Se calcula una vez al inicio y no se vuelve a calcular hasta que la convergencia en el método de Newton Raphson resulta demasiado lenta.

4.2.1.1. La matriz de diferencias

La *ode15s* está diseñada tomando como base las diferencias hacia atrás de las fórmulas BDF y NDF. Así, se define una matriz *dif* con las diferencias hacia atrás que se calculan durante el proceso. Esta matriz consta de tantas filas como dimensión tiene la EDO a resolver y el número de columnas es 2 unidades mayor que el orden máximo que hayamos definido. Por ejemplo, si el orden máximo al que se va a resolver es $k = 4$, la matriz de diferencias tiene $4 + 2 = 6$ columnas. Si queremos dar un paso de t_{n+k-1} a t_{n+k} para hallar y_{n+k} en orden k , se parte de la matriz de diferencias almacenada al final del paso anterior (paso de t_{n+k-2} a t_{n+k-1}):

$$dif_{n+k-1}(h) = (\nabla y_{n+k-1} \quad \nabla^2 y_{n+k-1} \quad \dots \quad \nabla^k y_{n+k-1} \quad \nabla^{k+1} y_{n+k-1} \quad \nabla^{k+2} y_{n+k-1}) \quad (4.56)$$

siendo $\nabla^j y_{n+k-1} = \nabla(\nabla^{j-1} y_{n+k-1})$, $\nabla y_{n+k-1} = y_{n+k-1} - y_{n+k-2}$.

Las diferencias hacia atrás también satisfacen esta igualdad:

$$\nabla^{j+1} y_{n+k-1} = \nabla^j y_{n+k-1} - \nabla^j y_{n+k-2} \quad (4.57)$$

Cada vez que se da un paso hay que actualizar la matriz de diferencias (4.56). Por lo que tras finalizar el paso de t_{n+k-1} a t_{n+k} necesitamos tener $dif_{n+k}(h)$. Tras calcular y_{n+k} se tiene la diferencia hacia atrás de orden $(k+1)$, $\nabla^{k+1} y_{n+k}$. Y son este valor y la matriz almacenada hasta este momento (4.56) las que se utilizan en la actualización. El proceso es el siguiente:

$$\begin{cases} dif_{n+k,k+2}(h) = \nabla^{k+1} y_{n+k} - dif_{n+k-1,k+1}(h) \\ dif_{n+k,k+1}(h) = \nabla^{k+1} y_{n+k} \\ dif_{n+k,j}(h) = dif_{n+k-1,j}(h) + dif_{n+k,j+1}(h), \text{ para } j = k, (k-1), \dots, 1 \end{cases} \quad (4.58)$$

En la expresión (4.58) hemos denotado por $dif_{n+k,i}(h)$ la columna i -ésima de la matriz $dif_{n+k}(h)$. La primera fórmula de (4.58) significa que en la columna $(k+2)$ de la matriz $dif_{n+k}(h)$ se introduce la diferencia entre $\nabla^{k+1} y_{n+k}$ y la columna $(k+1)$. Como $dif_{n+k-1,k+1}(h) = \nabla^{k+1} y_{n+k-1}$ y teniendo en cuenta la propiedad (4.57) se tiene:

$$dif_{n+k,k+2}(h) = \nabla^{k+1} y_{n+k} - \nabla^{k+1} y_{n+k-1} = \nabla^{k+2} y_{n+k} \quad (4.59)$$

La segunda fórmula de la expresión (4.58) significa que el valor $\nabla^{k+1} y_{n+k}$ se introduce en la columna $(k+1)$ de la matriz $dif_{n+k}(h)$. Tras realizar las dos primeras evaluaciones

de (4.58), sólo se han actualizado las columnas $(k+1)$ y $(k+2)$ de la matriz de diferencias:

$$dif_{n+k}(h) = (\nabla y_{n+k-1} \quad \nabla^2 y_{n+k-1} \quad \dots \quad \nabla^k y_{n+k-1} \quad \nabla^{k+1} y_{n+k} \quad \nabla^{k+2} y_{n+k}) \quad (4.60)$$

La tercera fórmula de (4.58) dice que en la columna j de la matriz $dif_{n+k}(h)$ se introduce la suma de lo que había anteriormente en esa columna y lo que hay ahora en la columna $(j+1)$, siendo $j = k, (k-1), \dots, 2, 1$. Este proceso empieza en $j = k$ y se repite hasta $j = 1$. La primera operación que se hace es ésta:

$$dif_{n+k,k}(h) = \nabla^k y_{n+k-1} + \nabla^{k+1} y_{n+k} \quad (4.61)$$

De nuevo, teniendo en cuenta (4.57) la igualdad (4.61) resulta:

$$\nabla^k y_{n+k-1} + \nabla^{k+1} y_{n+k} = \nabla^k y_{n+k} \quad (4.62)$$

Y después de esto la matriz $dif_{n+k}(h)$ que tenemos es:

$$dif_{n+k}(h) = (\nabla y_{n+k-1} \quad \nabla^2 y_{n+k-1} \quad \dots \quad \nabla^k y_{n+k} \quad \nabla^{k+1} y_{n+k} \quad \nabla^{k+2} y_{n+k}) \quad (4.63)$$

Repitiendo el proceso para $j = (k-1), \dots, 1$ en este orden, se consigue actualizar la matriz de diferencias que queda en función de las diferencias hacia atrás en t_{n+k} :

$$dif_{n+k}(h) = (\nabla y_{n+k} \quad \nabla^2 y_{n+k} \quad \dots \quad \nabla^k y_{n+k} \quad \nabla^{k+1} y_{n+k} \quad \nabla^{k+2} y_{n+k}) \quad (4.64)$$

4.2.2. Estimación del error en la ode15s

La *ode15s* utiliza el error local de truncamiento como estimador del error. Para el método de orden k éste viene dado por:

$$est \approx LTE = Ch^{k+1}y^{k+1}(t_n) + O(h^{k+2}) \quad (4.65)$$

siendo C la constante de error. La función *ode15s* utiliza las constantes de error normalizadas de los métodos BDF y NDF, que vienen dadas por (2.47) y (2.60) respectivamente.

Al tener disponibles las diferencias hacia atrás de distintos órdenes, se utilizan éstas para hallar una aproximación del valor $y^{k+1}(t_n)$ de la expresión (4.65). Para ello se procede de esta manera:

- Se aproxima $y(t)$ utilizando el polinomio interpolador regresivo de Newton que pasa por los $(k+2)$ puntos $\{(t_{n+i}, y_{n+i}) : i = -1, 0, 1, 2, \dots, k\}$:

$$y(t) \approx Q(t) = y_{n+k} + \sum_{j=1}^{k+1} \nabla^j y_{n+k} \frac{1}{j!h^j} \prod_{m=0}^{j-1} (t - t_{n+k-m}) \quad (4.66)$$

- Se calcula la derivada $(k + 1)$ -ésima de (4.66). Nótese que el grado del polinomio (4.66) es $(k + 1)$ y que es el grado que corresponde al subíndice $j = k + 1$ del sumatorio de (4.66):

$$\begin{aligned} \nabla^{k+1} y_{n+k} \frac{1}{(k+1)! h^{k+1}} \prod_{m=0}^k (t - t_{n+k-m}) = \\ \nabla^{k+1} y_{n+k} \frac{1}{(k+1)! h^{k+1}} (t - t_{n+k})(t - t_{n+k-1}) \dots (t - t_n) \end{aligned} \quad (4.67)$$

Al derivar $(k + 1)$ veces (4.66), excepto el sumando de mayor grado (4.67) el resto dan cero. La derivada $(k + 1)$ -ésima del sumando de mayor grado y por tanto la derivada $(k + 1)$ -ésima de (4.66) es ésta:

$$Q^{(k+1)}(t) = \nabla^{k+1} y_{n+k} \frac{1}{(k+1)! h^{k+1}} (k+1)! = \nabla^{k+1} y_{n+k} \frac{1}{h^{k+1}} \quad (4.68)$$

Y se tiene:

$$y^{(k+1)}(t) \approx Q^{(k+1)}(t) = \nabla^{k+1} y_{n+k} \frac{1}{h^{k+1}} \quad (4.69)$$

Sustituyendo la aproximación (4.69) y la constante de error correspondiente en (4.65), tenemos la expresión del estimador del error de la *ode15s* en función de diferencias hacia atrás:

$$LTE \approx C \cdot \nabla^{k+1} y_{n+k} = est \quad (4.70)$$

Esta diferencia hacia atrás está disponible en la posición $(k + 1)$ de la matriz (4.64). Si se utiliza la norma basada en la componente máxima, el paso dado con la *ode15s* será bueno si el estimador cumple:

$$\|est\| = C \cdot \max_{i=1,2,\dots,m} \left(\frac{|\nabla^{k+1} y_{n+k}^i|}{|y_{n+k-1}^i| + |y_{n+k}^i|} \right) \leq Rtol \quad (4.71)$$

Lo mismo que con la *ode45*, el estimador del error que utiliza la *ode15s* no es el dado por (4.71), sino que es este otro que lo denotaremos por $\|est_M\|$ y que es mayor que $\|est\|$ o ambos están acotados por los mismos valores (sirve la demostración que se ha hecho para la *ode45*):

$$\|est_M\| = C \cdot \max_{i=1,2,\dots,m} \left(\frac{|\nabla^{k+1} y_{n+k}^i|}{|y_{n+k-1}^i|, |y_{n+k}^i|, y_{min}^i} \right) \leq Rtol \quad (4.72)$$

4.2.3. Estimador del error local vs. regiones de estabilidad *ode15s*

El estimador del error que utiliza la *ode15s* no depende de $\hat{h} = h\lambda$ y no guarda relación alguna con las regiones de estabilidad de las BDF. Debido a que los ángulos de $A(\alpha)$ estabilidad son bastante grandes para la *ode15s* (consultar Tabla 2.5), la *ode15s* está diseñada bajo la hipótesis de $A(90^\circ)$ estabilidad y sólo se controla la estimación del error de cada paso. Esto puede conllevar la inestabilización del resultado si hay varios valores λh que caen fuera de la región de estabilidad, como posteriormente lo veremos en la Sección 4.3.2.

4.2.4. Control del tamaño de paso

4.2.4.1. Primer tamaño de paso

La *ode15s* da el primer paso utilizando el mismo criterio que utiliza la *ode45*. Únicamente cambia un poco la definición del tamaño de paso mínimo *hmin*:

$$hmin = 16 \cdot eps \cdot abs(t) = \begin{cases} 16 \cdot |t| \cdot eps, & t \neq 0 \\ 0, & t = 0 \end{cases} \quad (4.73)$$

Esta nueva expresión de *hmin* es muy parecida a la expresión (4.35) que teníamos para la *ode45*, con la excepción de que antes el valor de *hmin* correspondiente a $t = 0$ era un valor muy próximo a 0 y ahora es exactamente 0.

4.2.4.2. Nuevo tamaño de paso tras *h* bueno

Si estamos resolviendo en orden k y si el paso dado ha sido bueno, esto es $\|est\| \leq Rtol$, la *ode15s* calcula el tamaño de paso que correspondería a orden k , y le llama h_{opt} . Éste depende del tamaño de paso actual h y se calcula de esta manera:

$$h_{opt} = \begin{cases} 10 \cdot h, & \|est\| \leq Rtol \cdot A_k \\ \frac{h}{1,2} \left(\frac{Rtol}{\|est\|} \right)^{1/(k+1)}, & \|est\| > Rtol \cdot A_k \end{cases} \quad (4.74)$$

con $A_k = \left(\frac{0,1}{1,2} \right)^{(k+1)}$. Los valores de A_k correspondientes a distintos órdenes se han recogido en la Tabla 4.1.

Para ello, la *ode15s* define una variable *temp*:

$$temp = 1,2 \left(\frac{\|est\|}{Rtol} \right)^{1/(k+1)} \quad (4.75)$$

Y define el tamaño de paso en función del valor de $temp$:

$$h_{opt} = \begin{cases} 10 \cdot h, & temp \leq 0,1 \\ \frac{h}{temp}, & temp > 0,1 \end{cases} \quad (4.76)$$

Para llegar a la expresión (4.74), hemos calculado el valor de la estimación en función de $Rtol$ para que la variable $temp = 0,1$, resultando:

$$\|est\| = Rtol \cdot \left(\frac{0,1}{1,2}\right)^{(k+1)} \quad (4.77)$$

La *ode15s* dispone de un contador que cuenta los pasos que se han dado sin cambiar de orden ni de tamaño de paso. Para orden k , mínimamente hay que dar $(k + 2)$ pasos consecutivos con el mismo orden y tamaño de paso. Esto garantiza que para un mismo orden y tamaño de paso tendremos la matriz de diferencias completa, ya que ésta tiene $(k + 2)$ columnas. A partir de que se hayan dado estos $(k + 2)$ pasos obligatorios, la *ode15s* realiza varios cálculos. Por una parte, calcula cuál sería el siguiente tamaño de paso si siguiésemos trabajando con el mismo orden k , y por otra, calcula los tamaños de paso que tendríamos si pasáramos a trabajar en orden $(k - 1)$ (cuando $k > 1$) o en orden $(k + 1)$ (siempre que no hayamos alcanzado el orden máximo definido). Para calcular los nuevos tamaños de paso de órdenes $(k - 1)$ y $(k + 1)$, se necesitan las estimaciones del error del paso que se ha dado suponiendo que éste se hubiese dado en órdenes $(k - 1)$ y $(k + 1)$. Las estimaciones del error de órdenes $(k - 1)$ y $(k + 1)$ son las diferencias hacia atrás de órdenes k y $(k + 2)$ respectivamente, multiplicadas por las constantes de error correspondientes. Estas diferencias hacia atrás se encuentran en las columnas k y $(k + 2)$ de la matriz (4.64) respectivamente.

Denotaremos por *errkm1* la estimación del error que corresponde al método de orden $(k - 1)$. La *ode15s* define $temp_{k-1} = 1,3 \cdot \left(\frac{\|errkm1\|}{Rtol}\right)^{1/k}$ y en función del valor de $temp_{k-1}$ define el tamaño de paso *hkm1* que se daría si pasáramos a trabajar en orden $(k - 1)$:

$$hkm1 = \begin{cases} 10 \cdot h, & est \leq Rtol \cdot A_{k-1} \\ \frac{h}{temp_{k-1}}, & est > Rtol \cdot A_{k-1} \end{cases} \quad (4.78)$$

siendo $A_{k-1} = \left(\frac{0,1}{1,3}\right)^k$. Los valores de A_{k-1} para órdenes $2 \leq k \leq 5$ se han recogido en la Tabla 4.1.

De similar manera, denotaremos por *errkp1* la estimación del error del método de orden $(k + 1)$. Se define $temp_{k+1} = 1,4 \cdot \left(\frac{\|errkp1\|}{Rtol}\right)^{1/(k+2)}$ y el tamaño de paso *hkp1* que se tendría si pasáramos a trabajar en orden $(k + 1)$:

$$hkp1 = \begin{cases} 10 \cdot h, & est \leq Rtol \cdot A_{k+1} \\ \frac{h}{temp_{k+1}}, & est > Rtol \cdot A_{k+1} \end{cases} \quad (4.79)$$

con $A_{k+1} = \begin{pmatrix} 0,1 \\ 1,4 \end{pmatrix}^{(k+2)}$. De nuevo, se han recogido en la Tabla 4.1 los valores de A_{k+1} para órdenes $1 \leq k \leq 4$.

orden k	A_k	A_{k-1} (<i>errkm1</i>)	A_{k+1} (<i>errkp1</i>)
1	$6,944 \cdot 10^{-3}$	—	$3,644 \cdot 10^{-4}$
2	$5,787 \cdot 10^{-4}$	$5,917 \cdot 10^{-3}$	$2,603 \cdot 10^{-5}$
3	$4,823 \cdot 10^{-5}$	$4,552 \cdot 10^{-4}$	$1,859 \cdot 10^{-6}$
4	$4,019 \cdot 10^{-6}$	$3,501 \cdot 10^{-5}$	$1,328 \cdot 10^{-7}$
5	$3,349 \cdot 10^{-7}$	$2,693 \cdot 10^{-6}$	—

Tabla 4.1: Valores de las constantes A_k , A_{k-1} y A_{k+1} .

Una vez que se tienen los tamaños de paso correspondientes a órdenes k , $(k-1)$ y $(k+1)$, el proceso que se sigue hasta determinar el siguiente tamaño de paso es éste:

1. Se empieza comparando el tamaño de paso correspondiente a orden $(k-1)$ (*hkm1*), con el que corresponde a orden k (h_{opt}). Si $hkm1 > h_{opt}$, se guarda en h_{opt} el valor *hkm1* y se coge el orden que le corresponde a éste, es decir: $k_{nuevo} = (k-1)$. Sino, se mantiene el valor de h_{opt} y $k_{nuevo} = k$.
2. A continuación se compara el tamaño de paso correspondiente a orden $(k+1)$ (*hkp1*), con h_{opt} . Si $hkp1 > h_{opt}$, se guarda en h_{opt} el valor *hkp1* y se aumenta en una unidad el orden k_{nuevo} que hasta este momento se había guardado.
3. Por último, se compara el valor de h_{opt} con el valor del último tamaño de paso h y si $h_{opt} > h$, el siguiente paso se dará con tamaño h_{opt} y orden k_{nuevo} . Sino, se sigue trabajando con el mismo orden y tamaño de paso del último paso: en orden k y tamaño de paso h .

El hecho de que se tengan que cumplir las condiciones anteriores, evita que se cambie continuamente de tamaño de paso y de orden.

4.2.4.3. Nuevo tamaño de paso tras h fallido

Si no se cumple el requisito de tolerancia establecido, es decir $\|est\| > Rtol$, hay que repetir el paso. Se calcula el siguiente tamaño de paso que correspondería a orden k al que el código llama h_{opt} :

$$h_{opt} = \begin{cases} \max \left(hmin, h \cdot \max \left(0, 1, 0,833 \cdot \left(\frac{Rtol}{\|est\|} \right)^{\frac{1}{(k+1)}} \right) \right), & \text{primer intento h fallido} \\ 0,5 \cdot h, & \text{a partir del 2º intento} \end{cases} \quad (4.80)$$

En el caso en que $k > 1$ y si estamos en el primer intento tras un paso fallido, también se calcula el tamaño de paso que correspondería si pasáramos a trabajar a orden $(k - 1)$. Esto se hace utilizando una fórmula parecida a la primera expresión de (4.80):

$$hkm1 = \max \left(hmin, h \cdot \max \left(0, 1, 0,769 \cdot \left(\frac{Rtol}{\|errkm1\|} \right)^{1/k} \right) \right) \quad (4.81)$$

siendo $errkm1$ el estimador del error correspondiente a orden $(k - 1)$.

Si resulta que $hkm1 > h_{opt}$, el orden en el que se dará el siguiente paso será $(k - 1)$ y el siguiente tamaño de paso será el mínimo entre el tamaño de paso actual h y $hkm1$, no permitiendo aumentar el tamaño de paso actual tras un paso fallido:

$$h_{nuevo} = \min(h, hkm1) \quad (4.82)$$

Si este paso también resultara fallido, los siguientes pasos se darían dividiendo entre 2 el tamaño de paso actual.

4.2.4.4. Actualización de la matriz de diferencias

El cambio de tamaño de paso requiere una adecuación de la matriz de diferencias y esto resulta costoso [62]. Supongamos que se ha utilizado un tamaño de paso constante h por lo menos en los últimos $(k + 1)$ pasos ($(k + 2)$ es el requisito del contador) y que la integración ha alcanzado un instante temporal t_{n+k} . Por tanto, tenemos disponibles valores y_{n+j} en instantes $t_{n+k} - jh$ para $j = 0, 1, \dots, k$. El polinomio interpolador que pasa por estos puntos y que ha quedado definido en la Sección 2.2.1 mediante (2.29) es:

$$P(t) = y_{n+k} + \sum_{j=1}^k \nabla^j y_{n+k} \frac{1}{j!h^j} \prod_{m=0}^{j-1} (t - t_{n+k-m})$$

Los valores $\nabla^j y_{n+k}$ para $j = 1, \dots, k$ están almacenados en la matriz de diferencias de donde cogemos las primeras k columnas. Denotaremos por D esta matriz:

$$D = (\nabla y_{n+k} \quad \nabla^2 y_{n+k} \quad \nabla^3 y_{n+k} \quad \dots \quad \nabla^k y_{n+k}) \quad (4.83)$$

El hecho de cambiar a un tamaño de paso nuevo, $h^* \neq h$, significa que tenemos que calcular los valores de $P(t)$ en los puntos $t_{n+k-j}^* = t_{n+k} - jh^*$ para $j = 0, 1, \dots, k$ para formar:

$$D^* = (\nabla y_{n+k}^* \quad \nabla^2 y_{n+k}^* \quad \nabla^3 y_{n+k}^* \quad \dots \quad \nabla^k y_{n+k}^*) \quad (4.84)$$

La evaluación del polinomio $P(t)$ con tamaños de paso h o h^* nos conduce a la siguiente igualdad:

$$P(t) = y_{n+k} + \sum_{j=1}^k \nabla^j y_{n+k} \frac{1}{j! h^j} \prod_{m=0}^{j-1} (t - t_{n+k-m}) = y_{n+k} + \sum_{j=1}^k \nabla^j y_{n+k}^* \frac{1}{j! (h^*)^j} \prod_{m=0}^{j-1} (t - t_{n+k-m}^*) \quad (4.85)$$

Simplificando el factor y_{n+k} en la igualdad (4.85) y evaluando la igualdad en los valores $t = t_{n+k-r}^*$ para $r = 1, \dots, k$ se tienen k ecuaciones de la forma:

$$\sum_{j=1}^k \nabla^j y_{n+k} \frac{1}{j! h^j} \prod_{m=0}^{j-1} (t_{n+k-r}^* - t_{n+k-m}) = \sum_{j=1}^k \nabla^j y_{n+k}^* \frac{1}{j! (h^*)^j} \prod_{m=0}^{j-1} (t_{n+k-r}^* - t_{n+k-m}^*) \quad (4.86)$$

Y teniendo en cuenta las siguientes igualdades:

$$\begin{cases} t_{n+k-r}^* - t_{n+k-m} = (t_{n+k} - rh^*) - (t_{n+k} - mh) = mh - rh^* \\ t_{n+k-r}^* - t_{n+k-m}^* = (t_{n+k} - rh^*) - (t_{n+k} - mh^*) = mh^* - rh^* \end{cases} \quad (4.87)$$

la expresión (4.86) queda:

$$\sum_{j=1}^k \nabla^j y_{n+k} R_{jr} = \sum_{j=1}^k \nabla^j y_{n+k}^* U_{jr} \quad (4.88)$$

siendo U y R matrices de dimensiones $(k \times k)$, con valores:

$$\begin{aligned} R_{jr} &= \frac{1}{j!} \prod_{m=0}^{j-1} (m - r \frac{h^*}{h}) \\ U_{jr} &= \frac{1}{j!} \prod_{m=0}^{j-1} (m - r) \end{aligned} \quad (4.89)$$

Escribimos la expresión (4.88) matricialmente:

$$\sum_{j=1}^k \nabla^j y_{n+k} R_{jr} = \sum_{j=1}^k \nabla^j y_{n+k}^* U_{jr} \Rightarrow D \cdot R = D^* \cdot U \Rightarrow D^* = D \cdot R \cdot U^{-1} \quad (4.90)$$

Como la matriz U cumple que su inversa es ella misma [62], la expresión (4.90) queda:

$$D^* = D \cdot R \cdot U \quad (4.91)$$

Y la función *ode15s* utiliza (4.91) para adecuar la matriz de diferencias tras cambiar de tamaño de paso.

4.3. Aplicación de *ode45* y *ode15s* a ejemplos sencillos

En esta sección analizaremos el funcionamiento de las funciones *ode45* y *ode15s* en algunas EDOs sencillas. Consideraremos dos tipos de EDOs: EDOs de orden 1 con autovalores reales y EDOs de orden 2 con autovalores imaginarios puros. Veremos la posición que ocupan los valores $\hat{h} = \lambda h$ en las regiones de estabilidad correspondientes a los métodos de cada una de las funciones (DOPRI(5,4) y BDF/NDF).

4.3.1. Resolución de EDOs de orden 1

Consideramos la siguiente EDO de orden 1 de condición inicial:

$$y' = \lambda \cdot y, \quad y(0) = 1, \quad \text{siendo } \lambda < 0 \quad (4.92)$$

Hemos hallado la solución de (4.92) en el intervalo $T = [0, 10]$ utilizando las funciones *ode45* y *ode15s* (casos NDF y BDF), para distintos valores de λ y utilizando los valores establecidos por defecto en cada una de las funciones (*Rtol*, etc.). Los pasos que ha dado cada función se han recogido en la Tabla 4.2.

λ	<i>ode45</i>	<i>ode15s</i> NDF	<i>ode15s</i> BDF
-1	13	42	42
-2	19	55	55
-4	24	62	64
-8	36	68	70
-20	73	75	74
-100	314	80	78

Tabla 4.2: Número de pasos dado por la *ode45* y la *ode15s* en el problema (4.92).

La función *ode45* resulta más eficiente con los autovalores más pequeños, y la *ode15s* con los autovalores *grandes* (en ambos casos, BDF y NDF). Hemos analizado los valores $\lambda = -1$, en el que la *ode45* es más eficiente, y $\lambda = -100$ donde la más eficiente resulta la *ode15s*. En cada uno de estos dos casos hemos calculado el tamaño de paso y la estimación del error de ambas funciones. En el caso de la *ode15s* también hemos calculado el orden al que se da cada paso. Y en el caso de la *ode45* hemos dibujado la posición de los valores \hat{h} en el plano complejo, viendo si éstos caen en la región de estabilidad del método. No se ha dibujado la posición \hat{h} en las regiones de estabilidad de los métodos BDF o NDF, ya que \mathbb{R}^- pertenece a las regiones de estabilidad de éstas. Las gráficas que se muestran en esta sección corresponden a utilizar los métodos NDF en la función *ode15s*.

Para $\lambda = -1$ y en el caso de la *ode45*, todos los valores \hat{h} caen dentro de la región de estabilidad DOPRI(5,4) de orden 5, Figura 4.3. En la imagen izquierda de la Figura 4.4 se ve la regularidad que presenta la estimación del error, y en consecuencia los tamaños

de paso (imagen derecha de la Figura 4.4). En el caso de la *ode15s*, la disminución de error no siempre conlleva aumento del tamaño de paso (Figura 4.4), ya que sólo se plantea el cambio de tamaño de paso cuando el contador de la *ode15s* ha alcanzado el valor mínimo ($k+2$). Al no ser iguales las definiciones de tamaño de paso de un código y otro, los tamaños de paso de la función *ode15s* son menores que los de la *ode45*, aunque la estimación del error de la función *ode15s* es más pequeña que la estimación de la *ode45* en algunos instantes, ver Figura 4.4.

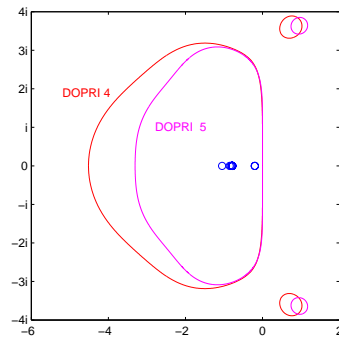


Figura 4.3: Posición \hat{h} de (4.92) en la región de estab. DOPRI(5,4), con $\lambda = -1$.

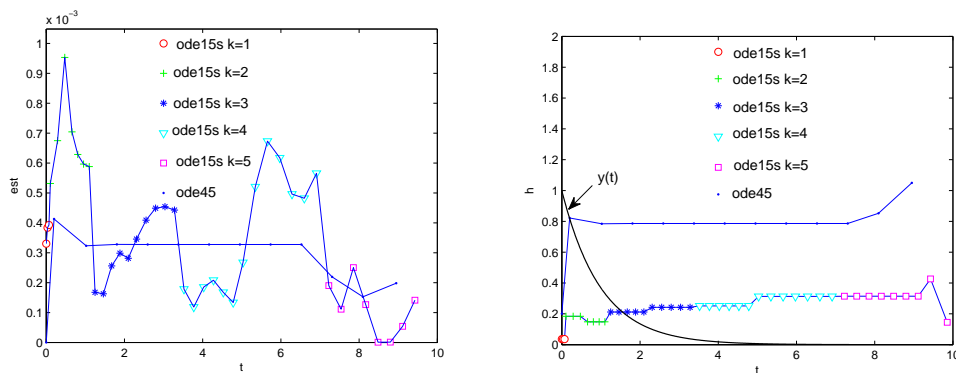


Figura 4.4: Estimación del error y tamaños de paso en (4.92), con $\lambda = -1$.

La función *ode15s* sólo plantea cambio de orden o de tamaño de paso cuando se han dado mínimamente ($k+2$) pasos con el mismo orden k y tamaño de paso h . En la imagen izquierda de la Figura 4.5 hemos dibujado el valor del contador tras finalizar cada paso. En la imagen derecha de la misma Figura, hemos superpuesto los tamaños de paso de la *ode15s* (imagen derecha Figura 4.4) y el valor del contador multiplicado por 10^{-1} , con el objeto de representar ambas cosas en una misma figura. Observamos que efectivamente la *ode15s* no cambia ni de orden ni de tamaño de paso mientras que el contador tiene un valor menor que ($k+2$).

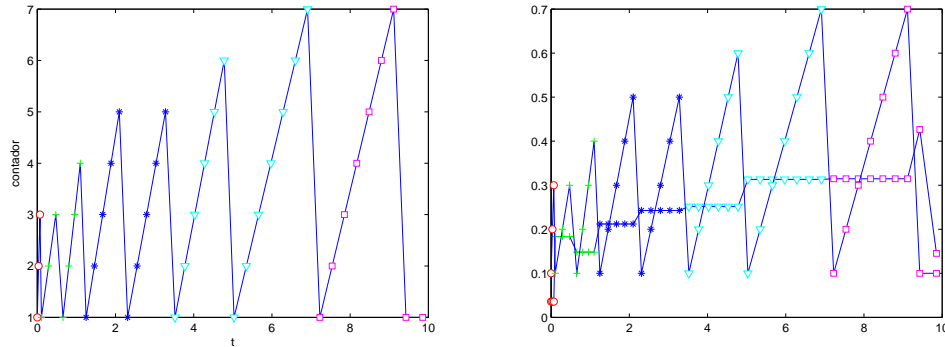


Figura 4.5: Funcionamiento del contador *ode15s* en (4.92), con $\lambda = -1$.

Una vez que el contador de la *ode15s* alcanza el valor mínimo establecido, la función plantea la posibilidad de cambio de orden o/y tamaño de paso. Sucede esto en los puntos 1–8 que se han marcado en la Figura 4.6. Si la función está trabajando en orden k y tamaño de paso h , calcula el siguiente tamaño de paso que le corresponde a este mismo orden y le llama h_{opt} . Si $k > 1$, la función también calcula el error que se habría cometido en el paso actual si hubiésemos estado trabajando en orden $(k - 1)$ y calcula el siguiente tamaño de paso para este orden, al que llama $hkm1$. Cuando k es menor que el orden máximo establecido a la función *ode15s*, también se calcula el error que se habría cometido en el paso actual trabajando en orden $(k + 1)$ y calcula el siguiente tamaño de paso para este orden, $hkp1$. La función empieza comparando $hkm1$ y h_{opt} . Si $hkm1 > h_{opt}$ guarda en h_{opt} el tamaño de paso $hkm1$ y coge el orden que corresponde a este tamaño de paso, es decir, $(k - 1)$. A continuación compara $hkp1$ y h_{opt} . Si $hkp1 > h_{opt}$ guarda en h_{opt} el tamaño de paso $hkp1$ y suma una unidad al orden que se tenía. Además, el tamaño de paso resultante de estas dos comparaciones tiene que ser mayor que el tamaño de paso actual h para que se cambie de tamaño de paso o/y orden. Sino, se mantienen el orden y el tamaño de paso anteriores.

En la imagen derecha de la Figura 4.6 hemos dibujado la estimación del error durante la integración (estimación que corresponde a orden k) y las estimaciones del error para órdenes $(k - 1)$ y $(k + 1)$ cuando el contador ya ha alcanzado el valor $(k + 2)$. En los puntos 1–8, el siguiente tamaño de paso se ha elegido de esta manera:

- En el punto 1 se está trabajando en orden $k = 1$ y sólo se ha calculado la estimación del error y el tamaño de paso para orden $(k + 1) = 2$, $hkp1$. Como éste resulta más grande que h_{opt} y más grande que el tamaño de paso al que se estaba trabajando (h), el siguiente tamaño de paso es $hkp1$ y se dará en orden $k = 2$.
- En los puntos marcados con 2, 4 y 7, sucede que $hkp1 > h_{opt}$ y también $hkp1 > h$. Por lo que el siguiente tamaño de paso es $hkp1$ y éste se dará en orden $(k + 1)$.
- En los puntos marcados con 3 y 5, no se cumplen ni $hkm1 > h_{opt}$ ni $hkp1 > h_{opt}$. Como $h_{opt} > h$, se cambia de tamaño de paso pero no de orden.

- En el punto 6, $hkp1 < h_{opt}$ y $h_{opt} < h$. Por tanto, se mantienen el orden y el tamaño de paso al que se venía trabajando.
- En el punto 8 se ha alcanzado el orden máximo $k = 5$, por lo que no se puede calcular $hkp1$. No se cumple $hkm1 > h_{opt}$, y como $h_{opt} > h$, se cambia el tamaño de paso pero no el orden.

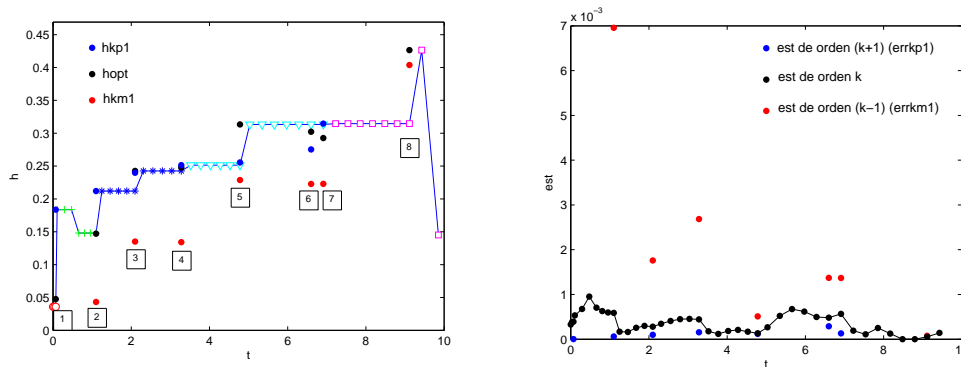


Figura 4.6: Siguiete tamaño de paso tras paso bueno en *ode15s* para (4.92), $\lambda = -1$.

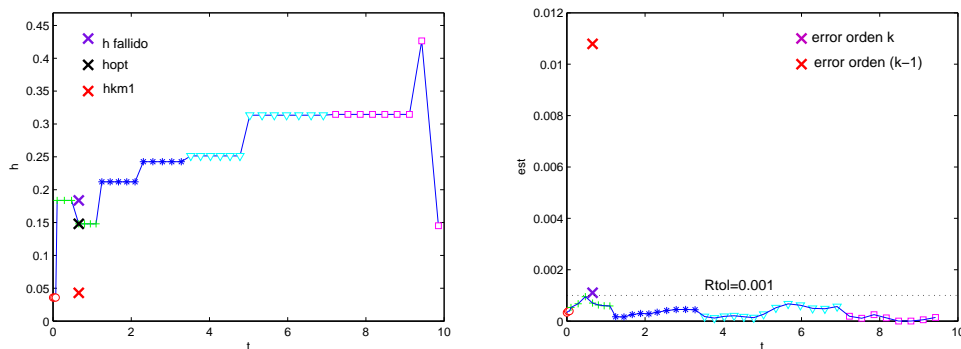


Figura 4.7: Siguiete tamaño de paso tras paso fallido en *ode15s* para (4.92), $\lambda = -1$.

En la Figura 4.7, se han representado los pasos buenos y los fallidos de la *ode15s* al resolver el problema (4.92) con $\lambda = -1$. Ha habido un paso fallido ($\|est\| > 0,001$). En el primer intento tras un paso fallido, la *ode15s* calcula el tamaño de paso que corresponde a orden $(k - 1)$, $hkm1$, y el que corresponde a orden k , h_{opt} . Si $hkm1 > h_{opt}$ el siguiente tamaño de paso se dará en orden $(k - 1)$ y el tamaño de paso será el mínimo entre $hkm1$ y el tamaño de paso actual h . Tras el paso que se ha fallado, como $h_{opt} > hkm1$ y $h_{opt} < h$, el siguiente paso se da con h_{opt} en el orden que se venía trabajando.

En el caso $\lambda = -100$ y *ode45*, algunos valores \hat{h} caen fuera de la región de estabilidad de orden 5, Figura 4.8. Los pasos que da la *ode45* son muy regulares, ver Figura 4.9, y están estrechamente unidos con la estimación del error, Figura 4.10: cuanto menor es el error más grande es el paso. En el caso de la *ode15s*, la estimación del error disminuye

mucho al avanzar en el tiempo y esto hace que el tamaño de paso aumente, Figuras 4.9 y 4.10. Se podría decir que los pasos más grandes de la *ode15s* suceden cuando la función solución no tiene muchas variaciones (en este caso valores próximos a 0).

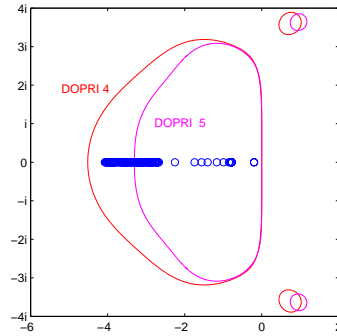


Figura 4.8: Posición \hat{h} de (4.92) en la región de estab. DOPRI(5,4), con $\lambda = -100$.

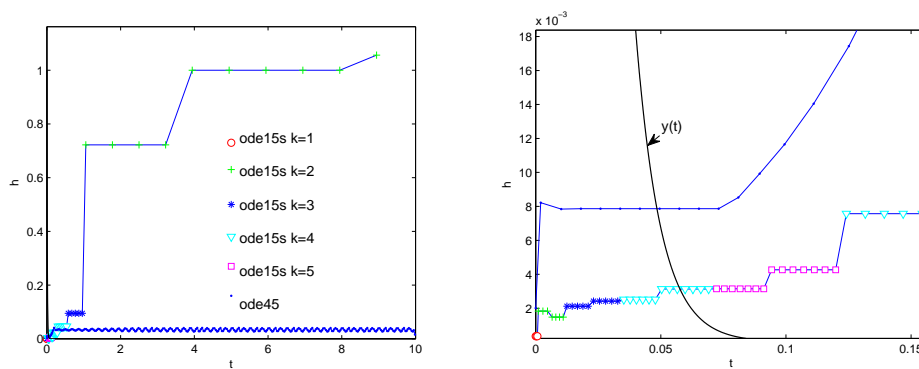


Figura 4.9: Tamaños de paso en (4.92), con $\lambda = -100$ (dcha. detalle).

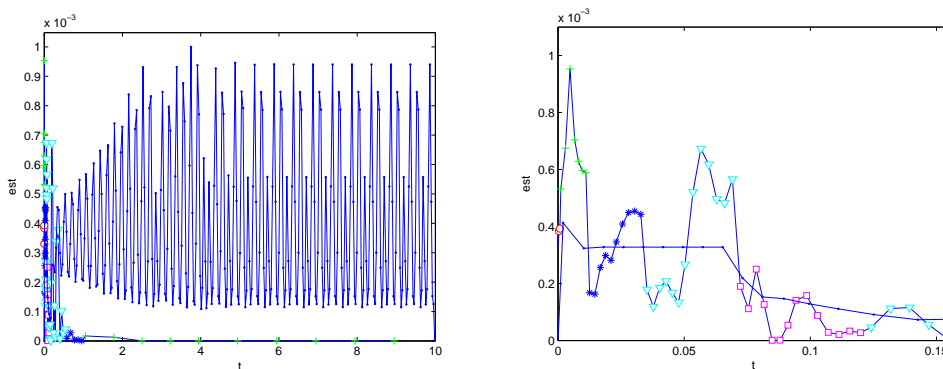


Figura 4.10: Estimación del error en (4.92), con $\lambda = -100$ (dcha. detalle).

4.3.2. Resolución de EDOs de orden 2

Consideramos la siguiente EDO de orden 2 de condición inicial:

$$y'' = \lambda^2 \cdot y, \quad (y(0), y'(0)) = (1, 0), \quad \text{siendo } \lambda^2 < 0 \quad (4.93)$$

Reducimos (4.93) a un sistema de 2 EDOs de orden 1:

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \lambda^2 & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix}, \quad (y(0), z(0)) = (1, 0) \quad (4.94)$$

Los autovalores del problema (4.94) son $\pm |\lambda| i$. Hemos resuelto (4.94) en el intervalo $T = [0, 10]$, utilizando las dos funciones *ode45* y *ode15s* (casos BDF y NDF), para distintos valores de λ^2 y utilizando los valores establecidos por defecto.

$\pm \lambda i$	<i>ode45</i>	<i>ode15s</i> NDF	<i>ode15s</i> BDF
$\pm 1i$	18	46	51
$\pm 2i$	31	81	82
$\pm 4i$	57	151	146
$\pm 8i$	109	291	281
$\pm 20i$	262	711	689
$\pm 100i$	1281	3515	3406

Tabla 4.3: Número de pasos dado por la *ode45* y la *ode15s* en el problema (4.94).

Hemos analizado más en detalle los valores $\lambda^2 = -1$ y $\lambda^2 = -100^2$, calculando para cada uno de estos dos casos los tamaños de paso requeridos por cada función y la estimación del error en cada paso, y se han dibujado las posiciones que ocupan los valores \hat{h} en el plano complejo. En el caso de la *ode15s* hemos especificado el orden al que se ha dado cada paso. En este análisis, se ha utilizado el método BDF en la función *ode15s*.

En el caso $\lambda^2 = -1$, los resultados obtenidos con ambas funciones han sido muy parecidos. Para la *ode45*, tanto los tamaños de paso como los valores del estimador son muy regulares, Figura 4.11. Además, todos los valores de \hat{h} caen dentro de la región de estabilidad de orden 5, Figura 4.12. En el caso de la *ode15s*, los tamaños de paso son menores y la estimación del error generalmente es mayor que la de la *ode45*, Figura 4.11. Los valores \hat{h} de la *ode15s* se han dibujado en la Figura 4.13, y exceptuando los \hat{h} que corresponden a los pasos dados en órdenes $k = 1$, $k = 2$ (que caen en la región de estabilidad por ser A-estables) y $k = 5$ (aunque éste no es A-estable), el resto de valores cae fuera de la región de estabilidad correspondiente, ver Figura 4.14.

En la Figura 4.15 se han dibujado el contador y los tamaños de paso. De nuevo se ve que no se cambia ni de orden ni de tamaño de paso mientras que el contador no alcanza

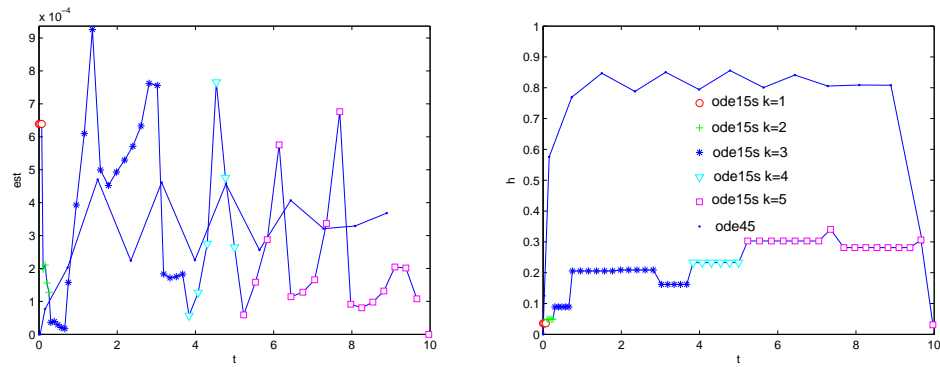


Figura 4.11: Estimación del error y tamaños de paso en (4.94), con $\lambda^2 = -1$.

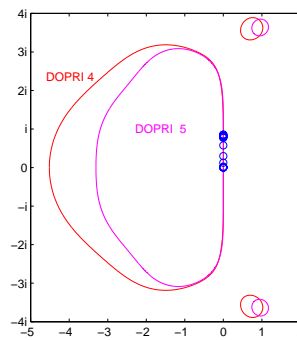


Figura 4.12: Posición \hat{h} de (4.94) en la región de estab. DOPRI(5,4), con $\lambda^2 = -1$.

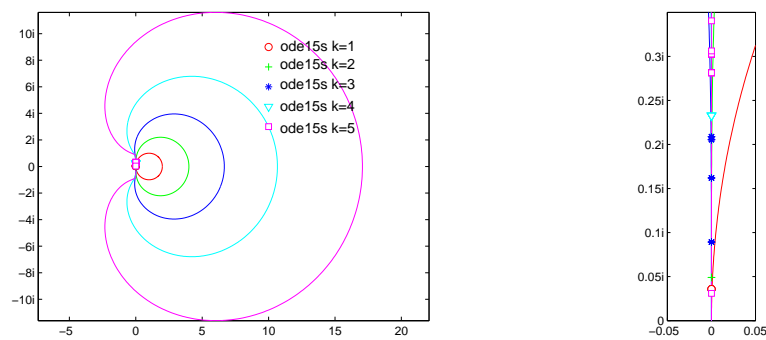


Figura 4.13: Posición \hat{h} de (4.94) en región de estab. BDF, con $\lambda^2 = -1$ (dcha. detalle).

el valor $(k + 2)$. En la Figura 4.16 se han representado los cambios de orden y/o tamaño de paso que se han dado tras un paso bueno y en la Figura 4.17 la repetición del paso tras un paso fallido (ha habido dos pasos fallidos).

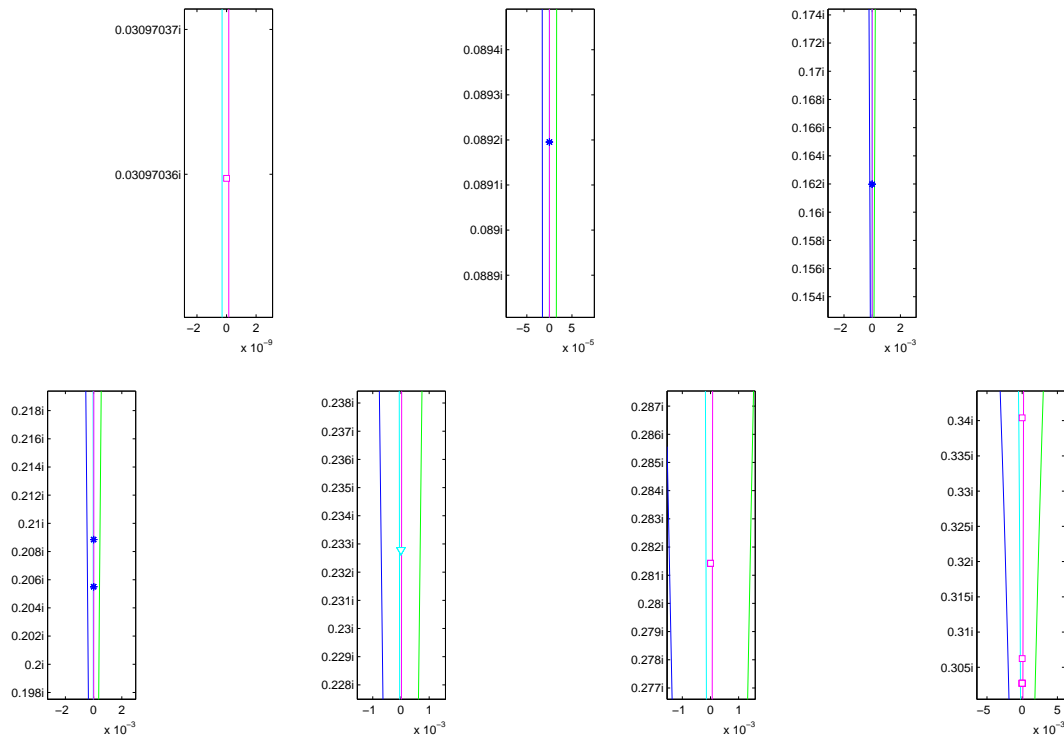


Figura 4.14: Posición \hat{h} de (4.94) en región de estab. BDF, (detalle de la Figura 4.13).

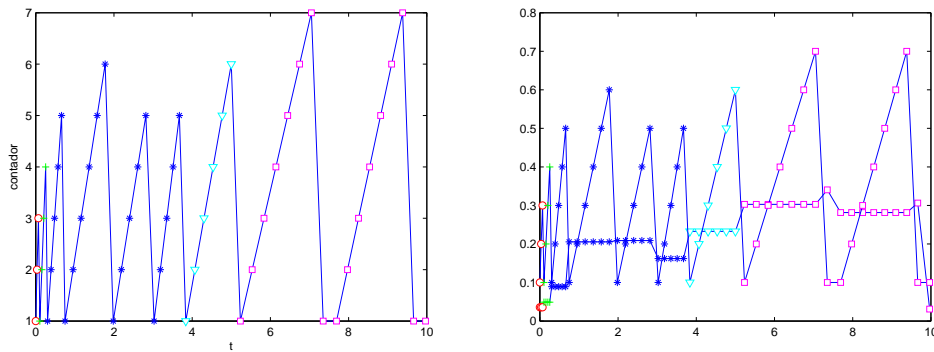


Figura 4.15: Funcionamiento del contador *ode15s* en (4.94), con $\lambda^2 = -1$.

En el caso de $\lambda^2 = -100^2$, hemos resuelto el problema (4.94) para dos valores distintos de la tolerancia relativa: 10^{-2} y 10^{-3} . Cuando $Rtol = 10^{-2}$, la *ode15s* da 1973 pasos (caso BDF) y la solución pierde mucha precisión, frente a los 768 pasos que da la *ode45* proporcionando una mejor solución. Con $Rtol = 10^{-3}$, la *ode15s* da 3406 pasos y la *ode45* 1281 pasos. De nuevo, la solución calculada con la *ode15s* es peor que la calculada con la *ode45*, Figuras 4.18 y 4.19.

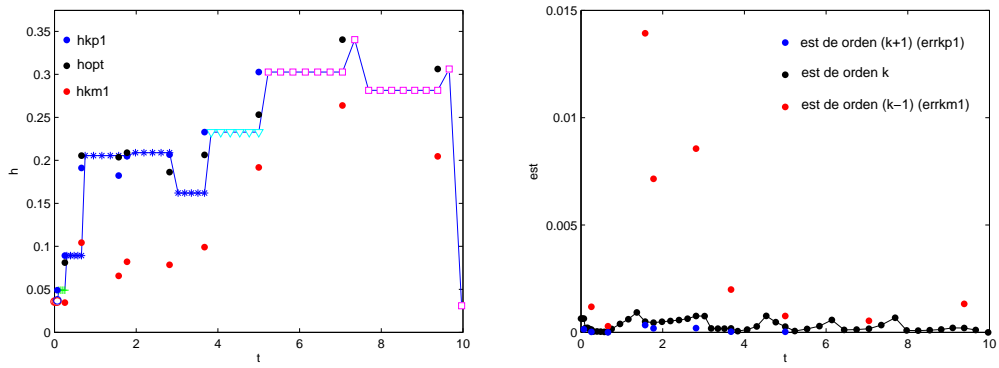


Figura 4.16: Siguiete tamaño de paso tras paso bueno en *ode15s* para (4.94), $\lambda^2 = -1$.

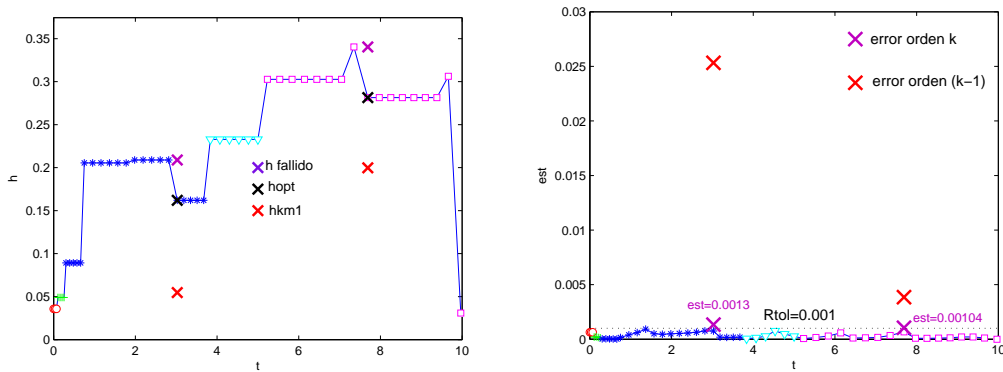


Figura 4.17: Siguiete tamaño de paso tras paso fallido en *ode15s* para (4.94), $\lambda^2 = -1$.

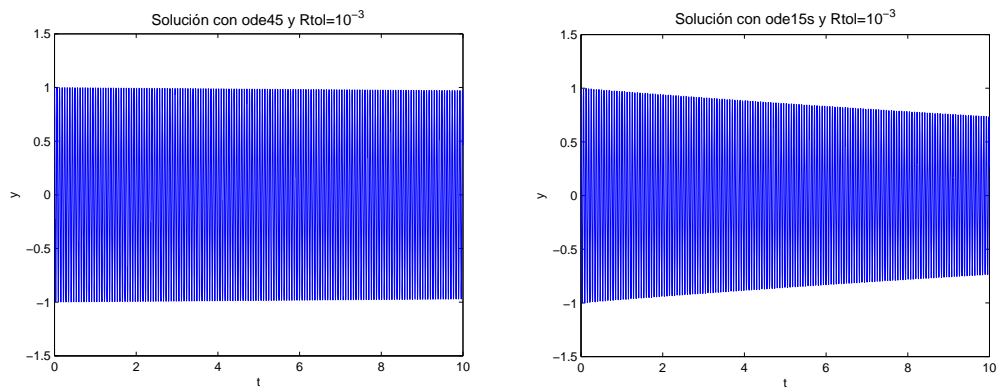


Figura 4.18: Soluciones de (4.94), $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$.

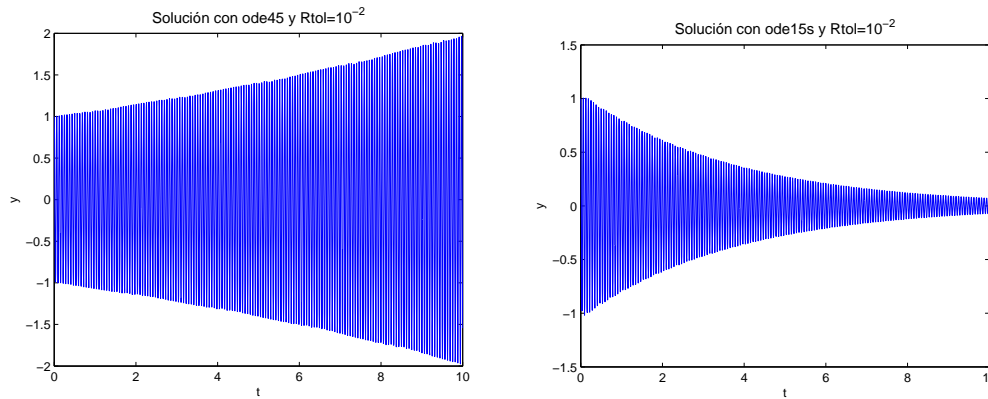


Figura 4.19: Soluciones de (4.94), $\lambda^2 = -100^2$ y $Rtol=10^{-2}$.

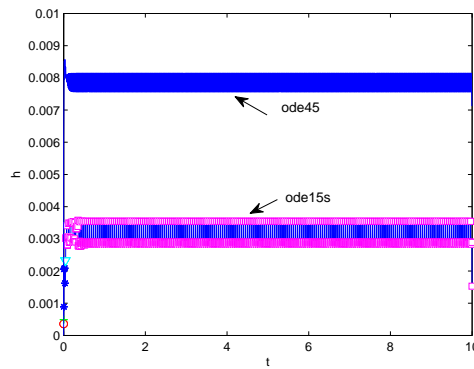


Figura 4.20: Tamaños de paso para el problema (4.94), $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$.

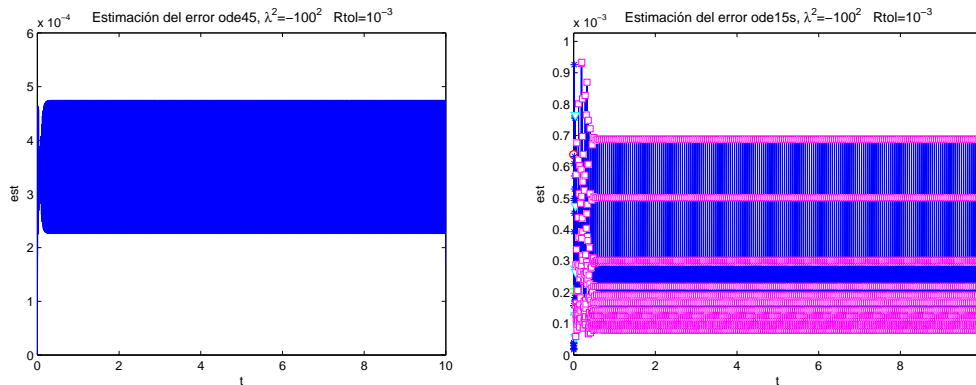


Figura 4.21: Estimación del error en (4.94), con $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$.

Las Figuras 4.20 y 4.21 muestran los tamaños de paso y las estimaciones del error. Aunque la media de las estimaciones del error de la *ode45* es 1,64 veces mayor que la de la *ode15s* ($est_{45} = 3,4778 \cdot 10^{-4}$ y $est_{15s} = 2,1186 \cdot 10^{-4}$), los tamaños de paso de la *ode45* resultan 2,66 veces mayores. Los valores \hat{h} de la función *ode45* caen dentro de

la región de estabilidad DOPRI(5,4) de orden 5, imagen derecha de la Figura 4.22. Los valores \hat{h} correspondientes a la función *ode15s* se han representado en la Figura 4.23. En la parte superior izquierda de esta figura se han dibujado las posiciones de todos los \hat{h} y el resto de imágenes muestran la posición de cada \hat{h} respecto de la región de estabilidad correspondiente a cada orden. A excepción de las que corresponden a órdenes $k = 1, 2$ y 5, el resto caen fuera de la región de estabilidad que les corresponde.

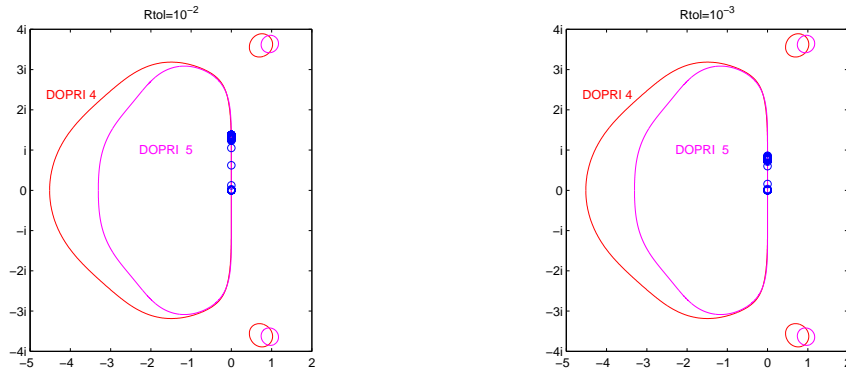


Figura 4.22: Posición \hat{h} de (4.94) en la región de estab. DOPRI(5,4), con $\lambda^2 = -100^2$.

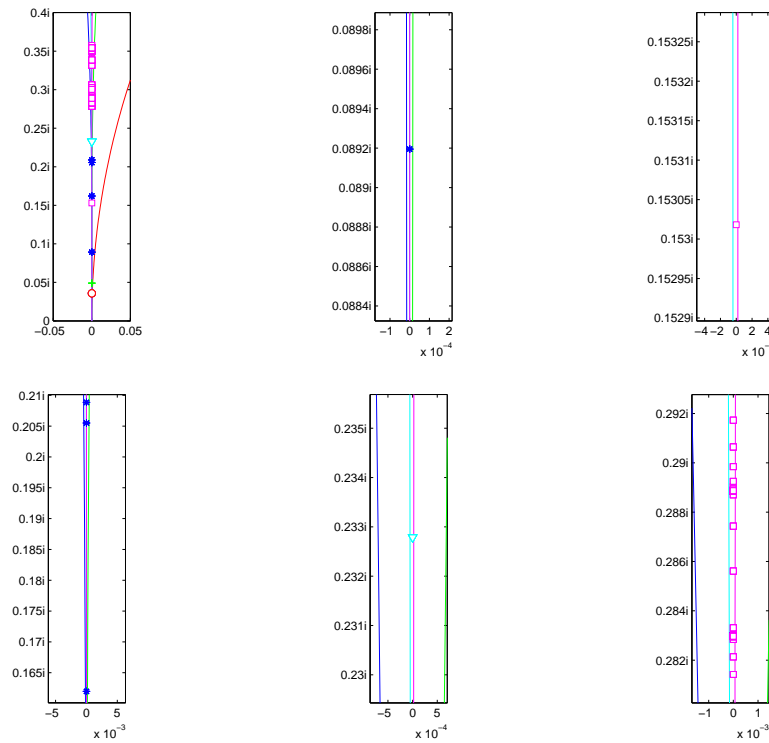


Figura 4.23: Posición \hat{h} de (4.94) en región de estab. BDF, $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$.

En el caso $Rtol = 10^{-2}$, se han dibujado en las Figuras 4.24 y 4.25 los tamaños de paso y las estimaciones del error. En este caso, la estimación del error de la *ode15s* es 1,50 veces mayor que el de la *ode45* ($est_{45} = 3,2557 \cdot 10^{-3}$ y $est_{15s} = 4,8863 \cdot 10^{-3}$), y los tamaños de paso de la *ode15s* resultan 2,57 veces menores.

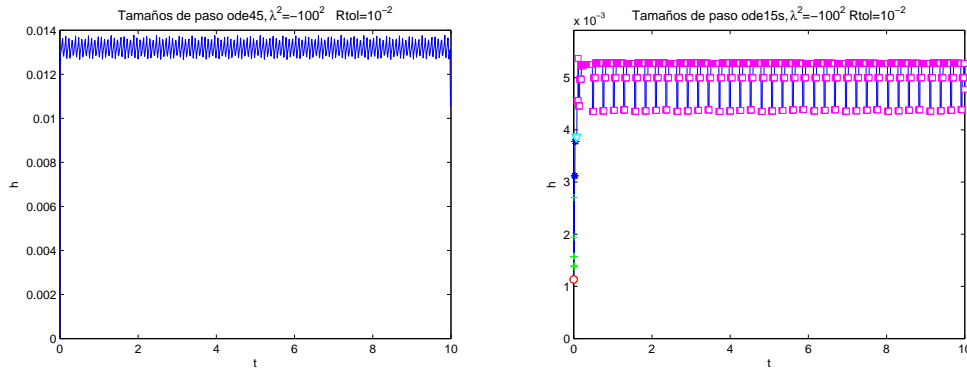


Figura 4.24: Tamaños de paso para el problema (4.94), $\lambda^2 = -100^2$ y $Rtol = 10^{-2}$.

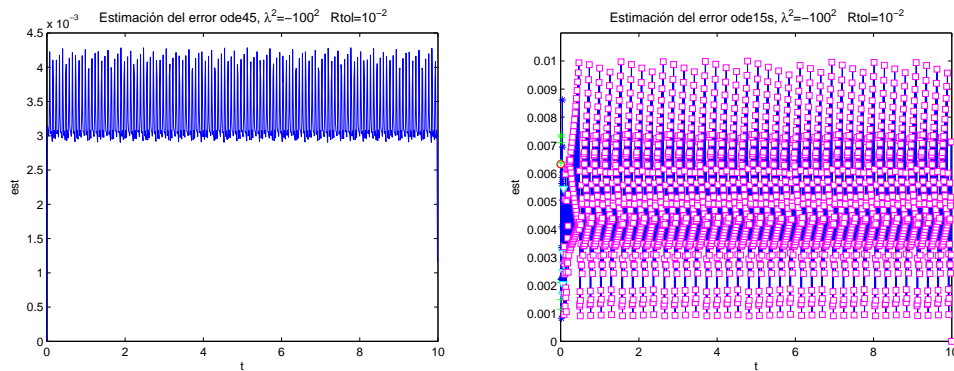


Figura 4.25: Estimación del error en (4.94), con $\lambda^2 = -100^2$ y $Rtol = 10^{-2}$.

En el caso de la función *ode45*, los valores \hat{h} caen todos dentro de la región de estabilidad DOPRI(5,4) de orden 5 (imagen izquierda de la Figura 4.22). En el caso de la función *ode15s*, a excepción de las que corresponden a órdenes $k = 1, 2$ y 5 , el resto caen fuera de la región de estabilidad, Figura 4.26. La solución obtenida con la *ode15s* empeora porque las posiciones de \hat{h} están más alejadas del origen que en el caso $Rtol = 10^{-3}$, compararlo con la Figura 4.23.



Figura 4.26: Posición \hat{h} de (4.94) en región de estab. BDF, $\lambda^2 = -100^2$ y $Rtol = 10^{-2}$.

4.3.3. Estimación del error en la ode15s

Con el objeto de entender mejor la falta de eficacia de la función *ode15s* al resolver el problema (4.94) (tanto con BDFs como NDFs) y teniendo en cuenta que los tamaños de paso dependen de las estimaciones del error, hemos analizado lo que sucede con las estimaciones del error de (4.94). La norma que utiliza la *ode15s* en la estimación del error puede ser de dos tipos: la basada en la componente máxima (es la que utiliza por defecto) y la norma euclídea. En las estimaciones del error de los ejemplos anteriores sólo se ha utilizado la norma basada en la componente máxima y esta vez hemos calculado la estimación del error utilizando ambas normas. Así, hemos resuelto (4.94) para $\lambda^2 = -1$, $\lambda^2 = -2$ y $\lambda^2 = -4$ utilizando las NDFs en la función *ode15s*.

Las Figuras 4.27 y 4.28 muestran las componentes solución del problema (4.94) con $\lambda^2 = -1$ (imagen izquierda) y las componentes del estimador del error (imagen derecha), calculado éste con la norma basada en la componente máxima. En las imágenes de la derecha se han superpuesto el valor absoluto de las componentes solución multiplicado por la constante 10^{-4} , con el fin de representar ambos (estimador y solución) en la misma figura. Resulta difícil sacar alguna conclusión sólo de estas dos gráficas.

En las Figuras 4.29 y 4.30 hemos dibujado las componentes solución del problema (4.94) con $\lambda^2 = -2$ y las componentes del estimador con la norma basada en la componente máxima. Las Figuras 4.31 y 4.32 muestran lo mismo para $\lambda^2 = -4$. Podemos concluir que a partir de un momento las componentes del estimador presentan un carácter periódico, parecido al que presentan las componentes de la solución y que el estimador es mínimo cerca de los puntos con tangencia horizontal en la función solución.

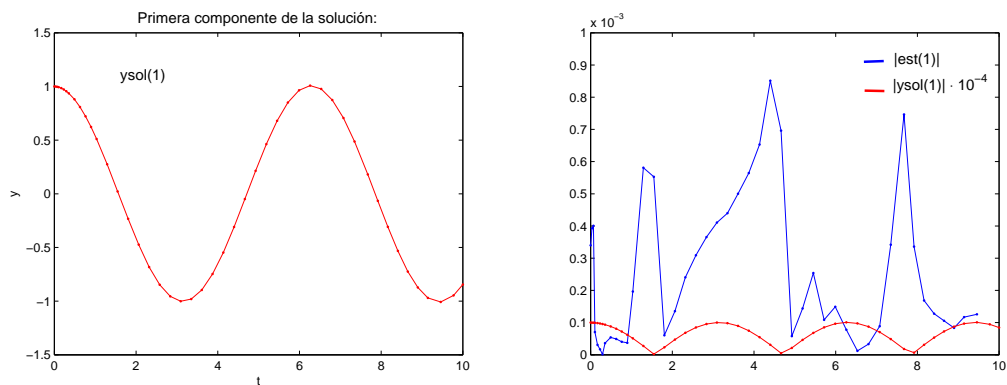


Figura 4.27: Solución y estimación del error en la 1ª componente para (4.94), $\lambda^2 = -1$.

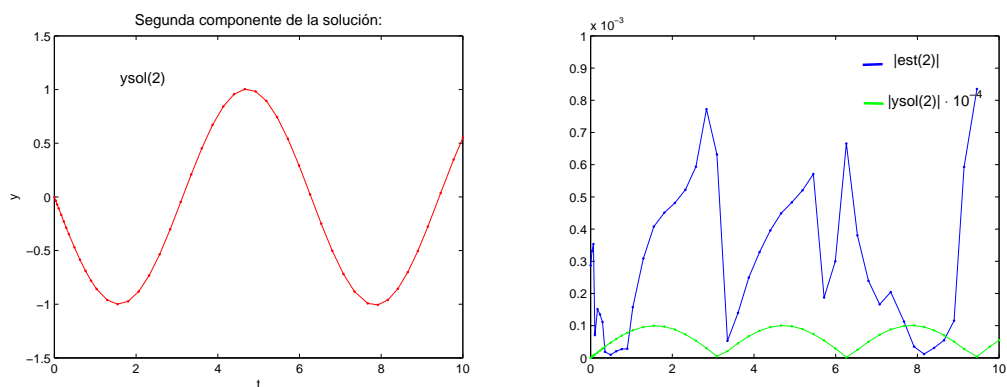


Figura 4.28: Solución y estimación del error en la 2ª componente para (4.94), $\lambda^2 = -1$.

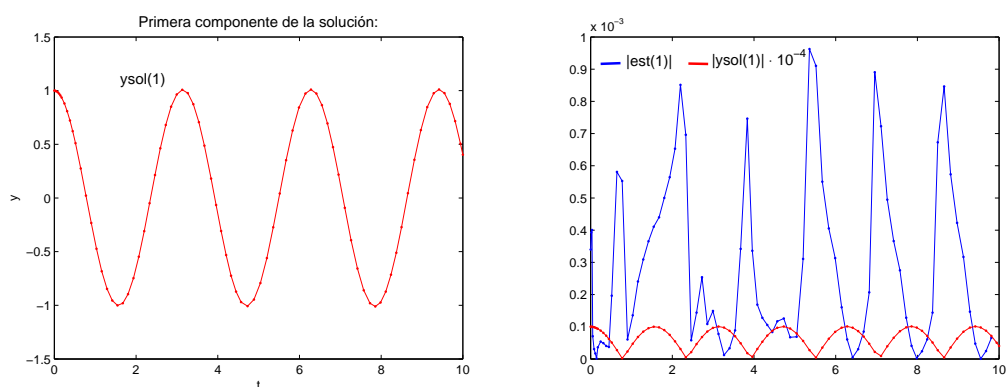


Figura 4.29: Solución y estimación del error en la 1ª componente para (4.94), $\lambda^2 = -2$.

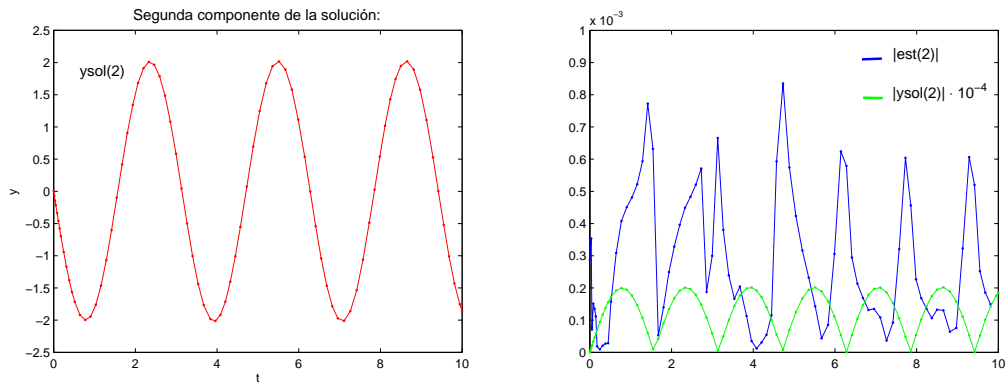


Figura 4.30: Solución y estimación del error en la 2ª componente para (4.94), $\lambda^2 = -2$.

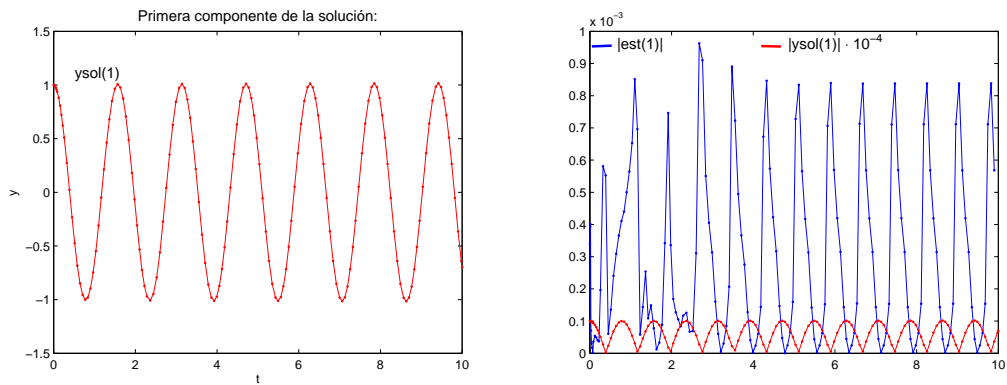


Figura 4.31: Solución y estimación del error en la 1ª componente para (4.94), $\lambda^2 = -4$.

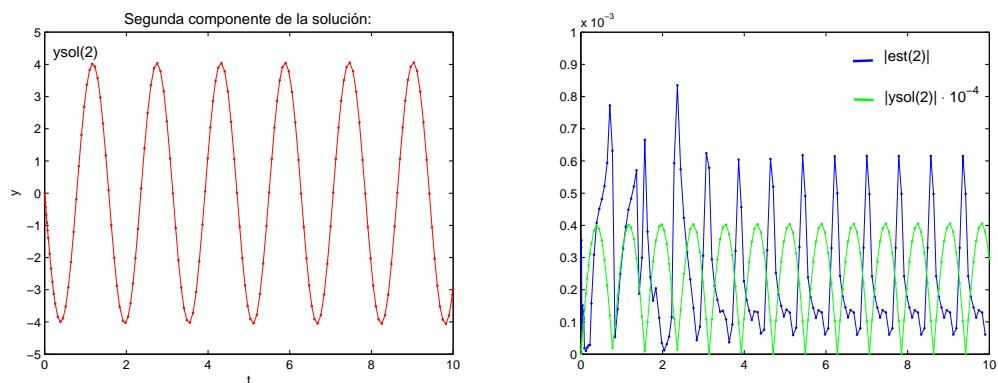


Figura 4.32: Solución y estimación del error en la 2ª componente para (4.94), $\lambda^2 = -4$.

En las Figuras 4.33 y 4.34 hemos representado el estimador del problema (4.94) calculado mediante la norma euclídea. También hemos representado el valor absoluto de las dos componentes solución multiplicado por la constante 10^{-4} . De nuevo, en los casos $\lambda^2 = -2$ y $\lambda^2 = -4$ el estimador presenta carácter periódico, y es menor cerca de donde las componentes solución tienen tangente horizontal.

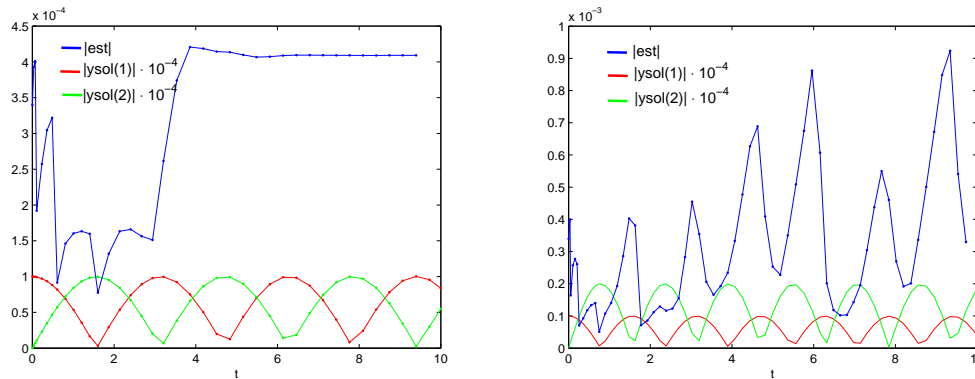


Figura 4.33: Solución y estimación del error con norma euclídea (4.94), $\lambda^2 = -1, -2$.

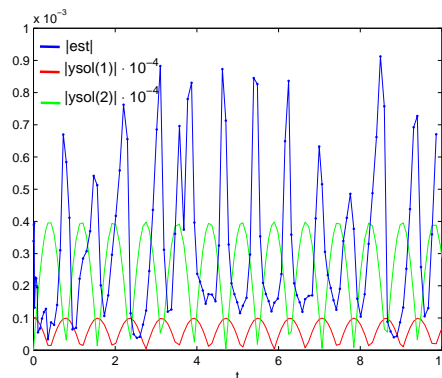


Figura 4.34: Solución y estimación del error con norma euclídea (4.94), $\lambda^2 = -4$.

4.3.4. Resolución de la ecuación de onda

Consideramos la ecuación de onda 1D (1.23) con CI correspondientes al pulso rectangular y CC dadas por (1.23), en una cuerda de 8 unidades de longitud $L = 8$ y $\alpha^2 = 1$, que es el ejemplo del Capítulo 1. Hemos visto que la función *ode15s* da muchos más pasos que la función *ode45*. Así, para una discretización de 100 elementos, la *ode45* precisa 1101 pasos y la *ode15s* 2857. Representamos en la Figura 4.35 los tamaños de paso dados por ambas funciones y en la Figura 4.36 las estimaciones del error (calculadas utilizando la norma basada en la componente máxima).

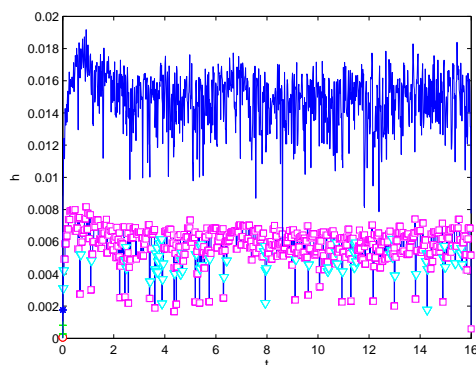


Figura 4.35: Tamaños de paso en (1.23) con CI pulso.

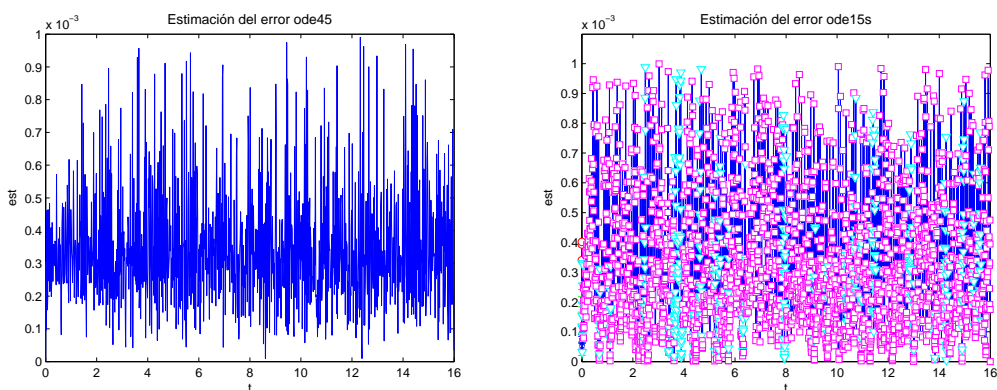


Figura 4.36: Estimación del error en (1.23) con CI pulso.

En el problema (1.23), las medias aritméticas de las estimaciones del error de ambas funciones son $est_{45} = 3,3839 \cdot 10^{-4}$ y $est_{15s} = 3,2891 \cdot 10^{-4}$, ambas muy parecidas. Sin embargo, si comparamos las medias de los tamaños de paso, la media de la función *ode45* es del orden de 2,59 veces más grande que la de la *ode15s*: $h_{45} = 1,4532 \cdot 10^{-2}$, $h_{15s} = 5,6003 \cdot 10^{-3}$. De nuevo nos encontramos con que el factor entre los tamaños de paso supera el factor entre las estimaciones del error.

La explicación a este hecho se encuentra en las expresiones (4.44) y (4.74) del tamaño de paso nuevo. Es suficiente con que nos fijemos en la forma en que cada función calcula el tamaño de paso tras un paso bueno. En la *ode45* éste se calcula utilizando:

$$h_{nuevo} = 5 \cdot h, \quad \|est\| \leq Rtol \cdot 1,0486 \cdot 10^{-4} \quad (4.95)$$

Y en la *ode15s*, suponiendo que el siguiente tamaño de paso se dará con el mismo orden k , éste se calcula:

$$h_{opt} = 10 \cdot h, \quad \|est\| \leq Rtol \cdot A_k \quad (4.96)$$

siendo las constantes A_k las de la Tabla 4.1. Escribiendo la expresión (4.96) explícitamente para cada orden se tiene:

$$h_{opt} = \begin{cases} 10 \cdot h, & \|est\| \leq Rtol \cdot 6,944 \cdot 10^{-3} \quad (k = 1) \\ 10 \cdot h, & \|est\| \leq Rtol \cdot 5,787 \cdot 10^{-4} \quad (k = 2) \\ 10 \cdot h, & \|est\| \leq Rtol \cdot 4,823 \cdot 10^{-5} \quad (k = 3) \\ 10 \cdot h, & \|est\| \leq Rtol \cdot 4,019 \cdot 10^{-6} \quad (k = 4) \\ 10 \cdot h, & \|est\| \leq Rtol \cdot 3,349 \cdot 10^{-7} \quad (k = 5) \end{cases} \quad (4.97)$$

Y comparando las expresiones (4.95) y (4.97), vemos que aunque la función *ode15s* multiplique por 10 el tamaño de paso actual y la *ode45* por 5, en la *ode15s* a partir de orden $k = 3$, el intervalo de la estimación del error para que suceda esto es considerablemente menor que en la *ode45*. Si forzamos a la función *ode15s* a que trabaje como máximo en orden 4, los pasos que precisa son 3282; en orden 3, 4899 pasos; en orden 2, 12169 y en orden 1, 56210 pasos, frente a los 2857 pasos que precisaba cuando no se le ponía ninguna restricción al orden. Esto es lógico ya que a menor orden, mayor resulta la estimación del error. Esto significa que tampoco los órdenes más pequeños presentan ninguna ventaja aunque su constante A_k sea mayor.

Como la función *ode45* resuelve en orden de precisión $k = 5$, si lo comparamos con el orden $k = 5$ de la *ode15s*, la condición $\|est\| \leq Rtol \cdot 3,349 \cdot 10^{-7}$ de la *ode15s* resulta más estricta que la condición $\|est\| \leq Rtol \cdot 1,0486 \cdot 10^{-4}$ de la *ode45*.

Pensamos que la fuerte restricción en el tamaño de paso para órdenes $k \geq 3$, se debe a que se ha querido establecer indirectamente un control sobre la posición de los valores $\hat{h} = h\lambda$ en estos órdenes, ya que a partir de orden $k = 3$ los métodos BDF y NDF no son A-estables. De esta manera se le obliga a la función *ode15s* a elegir tamaños de paso h que hacen que los valores \hat{h} estén cerca del origen de coordenadas, ver Figura 4.37.

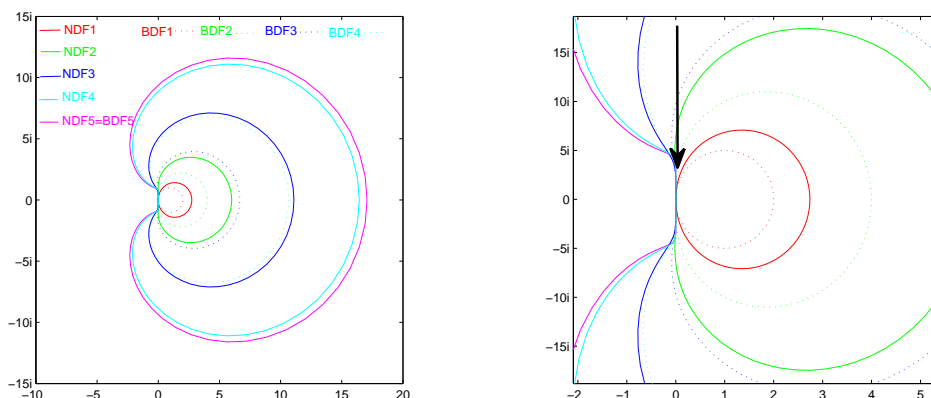


Figura 4.37: Detalle de las regiones de estabilidad BDF y NDF.

4.3.5. Conclusiones

Del estudio de estos ejemplos constatamos que:

- Las funciones *ode45* y *ode15s* sólo controlan una estimación del error local, lo que hace que el error global sólo se controle parcialmente y no totalmente.
- Las funciones *ode45* y *ode15s* no realizan el estudio de los autovalores de la matriz jacobiana. Esto significa que los valores $\hat{h} = \lambda h$, pueden caer fuera de la región de estabilidad correspondiente al método y esto puede conllevar pérdida de precisión e incluso inestabilización de la función solución.
- El estimador que utiliza la *ode15s* es la diferencia hacia atrás de orden $(k + 1)$ multiplicada por la constante de error del método, siendo k el orden al que se está resolviendo. Este estimador es una aproximación del error local de truncamiento y resulta muy pequeño cuando la función solución no muestra grandes variaciones. Es por esto que la *ode15s* es muy eficaz en el caso de EDOs de orden 1, principalmente a medida que aumenta $|\lambda|$. Sin embargo, no podemos decir lo mismo en el caso de EDOs de orden 2 con autovalores imaginarios puros. En este caso, las continuas variaciones de la función solución hacen que el estimador también vaya cambiando de forma parecida a la solución y por tanto, desaparece la ventaja que presentaba la *ode15s* en EDOs de orden 1.
- En la función *ode15s* las fórmulas de cálculo del tamaño de paso son más estrictas que en la *ode45*. Pensamos que la *ode15s* está diseñada de esta manera, para que los valores $\hat{h} = h\lambda$ estén cerca del origen de coordenadas, queriendo evitar que caigan en zonas de inestabilidad.
- La construcción de cada una de estas funciones (estimación del error, forma de calcular los tamaños de paso, etc.), es lo que hace que cada una de ellas resulte más eficiente a la hora de resolver unos problemas que otros.

Capítulo 5

Métodos resultantes de cambiar los predictores en EBDF y MEBDF

A veces, el replanteamiento de un problema es más decisivo que el hallazgo de la solución, que puede ser un puro asunto de habilidad matemática o experimental. La capacidad de suscitar nuevas cuestiones, nuevas posibilidades de mirar viejos problemas, requiere una imaginación creativa y determina los avances científicos auténticos.

ALBERT EINSTEIN

5.1. Introducción

Consideraremos la siguiente ecuación diferencial de condiciones iniciales:

$$y'(t) = f(t, y(t)), \quad y(a) = y_0 \quad (5.1)$$

definido en un intervalo finito $T = [a, b]$ y siendo $y : [a, b] \rightarrow \mathbb{R}^m$, $f : [a, b] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ funciones continuas.

Si estamos trabajando con un problema rígido, el método numérico que vayamos a utilizar en su resolución debe tener buena precisión y una región de estabilidad extensa [22]. Por ello, es natural que en los últimos años se haya trabajado mucho en conseguir métodos numéricos precisos capaces de resolver eficazmente ecuaciones diferenciales rígidas. Muchas de estas mejoras se han realizado tomando como base las BDF [30], debido a sus buenas características de estabilidad. Las BDF son A-estables hasta orden $k = 2$ y $A(\alpha)$ -estables hasta orden 6 [36, 38].

Como vimos en el Capítulo 2 una de las modificaciones hechas a las BDF son las NDF [62] que consisten en anticipar una diferencia de orden $(k + 1)$ multiplicada por

una constante $\kappa\gamma_k$, consiguiéndose así mayor precisión a costa de sacrificar un poco la estabilidad. En [62] sólo proponen añadir este sumando en órdenes $k = 1, 2, 3, 4$, ya que a partir de orden 5 la pérdida de estabilidad es significativa. En las referencias [16, 18] Cash introduce extensiones de los métodos BDF añadiendo puntos super-futuros para resolver problemas de valor inicial rígidos. A estos métodos se les conoce como BDF extendidas (EBDF) y BDF extendidas modificadas (MEBDF). Ambas utilizan dos predictores BDF y un corrector multipaso implícito. Los dos métodos son A-estables hasta orden 4 y $A(\alpha)$ -estables hasta orden 9.

Otras variaciones hechas a las BDF y que hemos visto en el Capítulo 2 consisten en: combinaciones lineales de las BDF (es el caso de los métodos A-BDF [28]), cambiar los predictores del esquema EBDF (caso de los métodos A-EBDF [44]), utilizar derivadas de orden superior (caso de los métodos de Enright [24] o los métodos SDBDF [37]), métodos multipaso extendidos que utilizan la segunda derivada (caso de los métodos E2BD [17]), etc. Todas estas modificaciones consiguen mejorar la precisión o/y el orden de los métodos BDF.

La motivación de la mejora que vamos a plantear en este capítulo es doble:

- Por una parte, sabemos que los métodos NDF suponen poco coste computacional añadido respecto de los métodos BDF y que aunque pierden en estabilidad, ganan en precisión.
- Y por otra parte, conocemos las buenas características de estabilidad que proporciona el esquema predictor-corrector propio de los métodos EBDF y MEBDF.

Teniendo en cuenta estas dos ideas, en este capítulo hemos modificado las EBDF y MEBDF utilizando como predictores las NDF y sin cambiar los correctores. Hemos experimentado con los dos predictores de la siguiente manera: utilizando NDFs en ambos predictores, utilizando NDFs como primer predictor y BDFs como segundo y viceversa. A los métodos resultantes de utilizar dos predictores NDF les hemos llamado ENDF y MENDF; a los que hemos obtenido utilizando NDFs como primer predictor y BDFs como segundo les hemos llamado métodos ENBDF y MENBDF y a los obtenidos con BDFs como primer predictor y NDFs como segundo, EBNDF y MEBNDF. Así, hemos creado las familias EBDF y MEBDF que utilizan como predictores BDFs y/o NDFs. Se han calculado las características de estabilidad y el orden de precisión de ambas familias. El resultado ha sido que se ha conseguido mejorar la estabilidad de los métodos de partida (EBDF y MEBDF) manteniendo la precisión (el orden del método).

5.2. Características de los métodos BDF, NDF, EBDF, MEBDF

En esta sección analizaremos algunas características de los métodos BDF, NDF, EBDF y MEBDF, que nos permitirán entender la construcción de estos métodos y proponer modificaciones en ellos.

5.2.1. Esquema de los métodos BDF y NDF

En el Capítulo 2 ha quedado dicho que los métodos BDF se escriben en función de diferencias hacia atrás mediante la expresión (2.40) y que desarrollando estas diferencias hacia atrás se llega a una forma equivalente de las mismas dada por (2.41):

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h f_{n+k} \quad (5.2)$$

Los métodos BDF de k -pasos tienen orden k y su error local de truncamiento es:

$$LTE = C_1 h^{k+1} y^{(k+1)}(t_n) + O(h^{k+2}) \quad (5.3)$$

De forma similar, también los métodos NDF se escriben en función de diferencias hacia atrás (2.59), y se pueden escribir utilizando una expresión parecida a la de los métodos BDF (5.2):

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h f_{n+k} + \kappa \gamma_k \nabla^{k+1} y_{n+k} \quad (5.4)$$

El coeficiente κ fue introducido por Klopfenstein y Shampine para maximizar el ángulo de $A(\alpha)$ -estabilidad al mismo tiempo que se reducía el error local de truncamiento. Las NDF son más precisas que las BDF aunque menos estables, siendo su error local de truncamiento éste:

$$LTE = C_2 h^{k+1} y^{(k+1)}(t_n) + O(h^{k+2}) \quad (5.5)$$

En este capítulo llamaremos C_1 y C_2 a las constantes de error de los métodos BDF y NDF respectivamente.

El hecho de que las NDF tengan mejor precisión que las BDF implica que pueden conseguir la misma precisión que las BDF con un tamaño de paso mayor (ver Tabla 2.5).

5.2.2. Esquema de los métodos EBDF y MEBDF

En la Sección 2.2.6.2 hemos visto que Cash [16] propuso los métodos EBDF, cuyo corrector viene dado por la siguiente expresión:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \beta_k f_{n+k} + h \beta_{k+1} \bar{f}_{n+k+1} \quad (5.6)$$

Cash también introdujo los métodos MEBDF [18], que siguiendo el esquema predictor-corrector de los métodos EBDF utilizan un corrector nuevo:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \hat{\beta}_k f_{n+k} + h \beta_{k+1} \bar{f}_{n+k+1} + h (\beta_k - \hat{\beta}_k) \bar{f}_{n+k} \quad (5.7)$$

los coeficientes β_k, β_{k+1} son iguales en ambos correctores y están disponibles en [16], y los coeficientes $\hat{\beta}_k$ son los coeficientes del método BDF al considerar éste de esta forma:

$$y_{n+k} + \sum_{j=0}^{k-1} \tilde{\alpha}_j y_{n+j} = h \hat{\beta}_k f_{n+k} \quad (5.8)$$

Suponiendo que $y_n, y_{n+1}, \dots, y_{n+k-1}$ están disponibles, el esquema predictor-corrector de los métodos EBDF y MEBDF consiste en:

1. Calcular \bar{y}_{n+k} como la solución del método BDF:

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h f_{n+k}, \quad (y_{n+k} := \bar{y}_{n+k}) \quad (5.9)$$

2. Calcular \bar{y}_{n+k+1} avanzando un nuevo paso BDF:

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j+1} = h f_{n+k+1}, \quad (y_{n+k+1} := \bar{y}_{n+k+1}) \quad (5.10)$$

3. Evaluar $\bar{f}_{n+k+1} = f(t_{n+k+1}, \bar{y}_{n+k+1})$ y también $\bar{f}_{n+k} = f(t_{n+k}, \bar{y}_{n+k})$ en el caso de las MEBDF.
4. Sustituir \bar{f}_{n+k+1} y también \bar{f}_{n+k} en el caso de las MEBDF en los correctores EBDF y MEBDF, (5.6) y (5.7) respectivamente, para calcular y_{n+k} , que será el valor numérico obtenido con el método EBDF o MEBDF.

El error local de truncamiento de estos métodos viene dado por las siguientes expresiones:

- Error local de truncamiento del método EBDF:

$$LTE_k = h^{k+2} \left[\boxed{\beta_{k+1} C_1 \left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} \right) \frac{\partial f}{\partial y} y^{(k+1)}} + C_3 y^{(k+2)} \right] (t_n) + O(h^{k+3}) \quad (5.11)$$

siendo C_1 la constante de error de los métodos BDF y C_3 la constante de error del corrector (5.6).

- Error local de truncamiento del método MEBDF:

$$LTE_k = h^{k+2} \left[\boxed{C_1 \left(\beta_{k+1} \left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} \right) + (\beta_k - \hat{\beta}_k) \right) \frac{\partial f}{\partial y} y^{(k+1)}} + C_4 y^{(k+2)} \right] (t_n) + O(h^{k+3}) \quad (5.12)$$

siendo C_1 la constante de error de los métodos BDF y C_4 la constante de error del corrector (5.7).

Lemma. Si las fórmulas correctoras (5.6) y (5.7) son de orden $(k + 1)$ y los métodos utilizados en (5.9) y (5.10) son de orden k , el algoritmo predictor-corrector (1)-(4) que corresponde al método EBDF o MEBDF es de orden $(k + 1)$. Se puede encontrar la demostración de este lema en [37]. \square

El polinomio característico de los métodos EBDF y MEBDF viene dado por:

$$A\hat{h}^3 + B\hat{h}^2 + C\hat{h} + D = 0 \quad (5.13)$$

siendo:

$$\begin{cases} A = -\hat{\beta}_k r^k \\ B = 2\hat{\alpha}_k \hat{\beta}_k r^k + T - \beta_{k+1} S - (\beta_k - \hat{\beta}_k) R \\ C = -\hat{\beta}_k \hat{\alpha}_k^2 r^k - 2\hat{\alpha}_k T + \hat{\alpha}_k \beta_{k+1} S - \beta_{k+1} \hat{\alpha}_{k-1} R + (\beta_k - \hat{\beta}_k) R \hat{\alpha}_k \\ D = \hat{\alpha}_k^2 T \\ R = \sum_{j=0}^{k-1} \hat{\alpha}_j r^j, \quad S = \sum_{j=0}^{k-2} \hat{\alpha}_j r^{j+1}, \quad T = \sum_{j=0}^k \alpha_j r^j, \end{cases} \quad (5.14)$$

Los coeficientes de la expresión (5.14) corresponden al método MEBDF y las diferencias con respecto a los coeficientes del método EBDF se han representado utilizando distintos colores: los sumandos en color azul son específicos del método MEBDF y los coeficientes en color rojo hay que sustituirlos por β_k en el caso del método EBDF. Obsérvese que las constantes α_j, β_j son las constantes de los métodos EBDF y MEBDF, y que sólo aparecen en la fórmula de T. Las constantes $\hat{\alpha}_j$ son las constantes del método BDF cuando consideramos la BDF de esta manera: $\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h f_{n+k}$.

5.3. Esquema de los nuevos métodos

5.3.1. ENDF y MENDF

Estos métodos consisten en utilizar NDFs como primer y segundo predictor en los métodos EBDF y MEBDF. A continuación, de igual manera que en los métodos EBDF y MEBDF, se evalúan $\bar{f}_{n+k+1} = f(t_{n+k+1}, \bar{y}_{n+k+1})$ y $\bar{f}_{n+k} = f(t_{n+k}, \bar{y}_{n+k})$ (este último sólo en el caso del MEBDF), y se sustituyen en las expresiones (5.6) y (5.7).

Primer predictor: En ambos métodos (ENDF y MENDF), la primera vez que se aplica el predictor NDF se obtiene el valor \bar{y}_{n+k} , siendo la diferencia entre el valor exacto y el calculado la siguiente:

$$y(t_{n+k}) - \bar{y}_{n+k}^* = C_2 h^{k+1} y^{(k+1)}(t_n) + O(h^{k+2}) \quad (5.15)$$

donde C_2 es la constante de error del método NDF. Hemos denotado por \bar{y}_{n+k}^* el valor que se calcula haciendo la asunción de localización (2.5).

Segundo predictor: De nuevo en ambos métodos (ENDF y MENDF), la segunda vez que se aplica el método NDF se obtiene \bar{y}_{n+k+1} , siendo la diferencia entre el exacto y el calculado la siguiente:

$$y(t_{n+k+1}) - \bar{y}_{n+k+1}^* = C_2 \left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} - \frac{\kappa\gamma_k(k+1)}{\hat{\alpha}_k} \right) h^{k+1} y^{(k+1)}(t_n) + O(h^{k+2}) \quad (5.16)$$

donde \bar{y}_{n+k+1}^* se ha calculado haciendo la asunción de localización y cogiendo el valor \bar{y}_{n+k}^* como valor del instante t_{n+k} .

Corrector: Finalmente, al aplicar los correctores (5.6) (caso EBDF) o (5.7) (caso MEBDF), el error local de truncamiento de los métodos ENDF y MENDF queda:

- Caso ENDF:

$$LTE_k = h^{k+2} \left(\beta_{k+1} A_k \frac{\partial f}{\partial y} y^{(k+1)} + C_3 y^{(k+2)} \right) (t_n) + O(h^{k+3}) \quad (5.17)$$

- Caso MENDF:

$$LTE_k = h^{k+2} \left[\left(\beta_{k+1} A_k + C_2(\beta_k - \hat{\beta}_k) \right) \frac{\partial f}{\partial y} y^{(k+1)} + C_4 y^{(k+2)} \right] (t_n) + O(h^{k+3}) \quad (5.18)$$

$$\text{siendo en ambos casos: } A_k = C_2 \left(1 - \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} - \frac{\kappa\gamma_k(k+1)}{\hat{\alpha}_k} \right)$$

El hecho de utilizar el método NDF como predictor de los métodos EBDF y MEBDF influye en el primer sumando del error local de truncamiento de éstos (parte encuadrada en (5.11) y (5.12)). El segundo sumando de ambos únicamente depende del corrector y nosotros no hemos actuado sobre él.

5.3.2. ENBDF y MENBDF, EBNDF y MEBNDF

En la sección anterior hemos utilizado el método NDF en las dos predicciones de los métodos EBDF y MEBDF. Pero esta opción no es la única, ya que también podemos utilizar el método NDF como primer predictor y el BDF como segundo o viceversa (primero BDF y después NDF). Por tanto, todas las opciones posibles para los predictores son: BDF-BDF, NDF-NDF, NDF-BDF, BDF-NDF. En todos los casos el error local de truncamiento se puede expresar de la siguiente manera:

$$LTE_k = h^{k+2} \left[\left(\beta_{k+1} A_k + C_i(\beta_k - \hat{\beta}_k) \right) \frac{\partial f}{\partial y} y^{(k+1)} + D_i y^{(k+2)} \right] (t_n) + O(h^{k+3}) \quad (5.19)$$

siendo: $D_i = C_3$ en el caso EBDF y $D_i = C_4$ para MEBDF. Hemos escrito en color azul el sumando extra que aparece en el caso MEBDF y que no aparece en los métodos EBDF.

Los valores de A_k dependen de los predictores utilizados, es decir:

- Caso EBDF y MEBDF (predictores BDF-BDF):

$$A_k = C_1 \left(-\frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} + 1 \right), \quad C_i = C_1 \quad (5.20)$$

- Caso ENDF y MENDF (predictores NDF-NDF):

$$A_k = C_2 \left(-\frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} - \kappa(k+1)\gamma_k \frac{1}{\hat{\alpha}_k} + 1 \right), \quad C_i = C_2 \quad (5.21)$$

- Caso ENBDF y MENBDF (predictores NDF-BDF):

$$A_k = \left(-C_2 \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} + C_1 \right), \quad C_i = C_2 \quad (5.22)$$

- Caso EBNDF y MEBNDF (predictores BDF-NDF):

$$A_k = \left(-C_1 \frac{\hat{\alpha}_{k-1}}{\hat{\alpha}_k} - C_1 \kappa(k+1)\gamma_k \frac{1}{\hat{\alpha}_k} + C_2 \right), \quad C_i = C_1 \quad (5.23)$$

5.4. Análisis de estabilidad

5.4.1. Estabilidad de los métodos ENDF y MENDF

Para calcular la región de estabilidad de estos métodos hay que aplicar el esquema completo a la ecuación de test $y' = \lambda y$. Aplicando dos veces el predictor NDF (5.4) y el corrector (5.6) en el caso ENDF y (5.7) en el caso MENDF a la ecuación de test, se llega a una ecuación en diferencias. Estableciendo $y_m = r^{m+1}$ para $m = (n-1), n, \dots, (n+k)$ y dividiendo entre r^n , se llega al polinomio característico del método:

$$A\hat{h}^3 + B\hat{h}^2 + C\hat{h} + D = 0 \quad (5.24)$$

siendo $\hat{h} = h\lambda$ y las constantes A, B, C, D dadas por:

$$\begin{cases} A = -\hat{\beta}_k r^{k+1} \\ B = 2(\hat{\alpha}_k - \kappa\gamma_k)\hat{\beta}_k r^{k+1} + T - \beta_{k+1}S + (\beta_k - \hat{\beta}_k)R \\ C = -\hat{\beta}_k(\hat{\alpha}_k - \kappa\gamma_k)^2 r^{k+1} - 2(\hat{\alpha}_k - \kappa\gamma_k)T + (\hat{\alpha}_k - \kappa\gamma_k)\beta_{k+1}S \\ \quad - \beta_{k+1}(-\hat{\alpha}_{k-1} - (k+1)\kappa\gamma_k)R - (\beta_k - \hat{\beta}_k)R(\hat{\alpha}_k - \kappa\gamma_k) \\ D = (\hat{\alpha}_k - \kappa\gamma_k)^2 T \end{cases} \quad (5.25)$$

y R, S, T dadas por:

$$\begin{cases} R = (-1)^{k+1} \binom{k+1}{0} \kappa \gamma_k + \sum_{j=1}^k r^j \left(-\hat{\alpha}_{j-1} + (-1)^{k+1-j} \binom{k+1}{j} \kappa \gamma_k \right) \\ S = (-1)^k r \kappa \gamma_k - \sum_{j=1}^{k-1} r^{j+1} \left(-\hat{\alpha}_{j-1} + (-1)^{k+1-j} \binom{k+1}{j} \kappa \gamma_k \right) \\ T = \sum_{j=0}^k \alpha_j r^{j+1} \end{cases} \quad (5.26)$$

De nuevo, los coeficientes de la expresión (5.25) corresponden al método MENDF y las diferencias con respecto a los coeficientes del método ENDF se han representado utilizando distintos colores: los sumandos en color azul son específicos del método MENDF y los coeficientes en color rojo hay que sustituirlos por β_k en el caso del método ENDF.

En la Figura 5.1 se muestran las regiones de estabilidad de los métodos EBDF y ENDF (imagen izquierda) y las de los métodos MEBDF y MENDF (imagen derecha). Las características de estabilidad se han recogido en las Tablas 5.1 y 5.2.

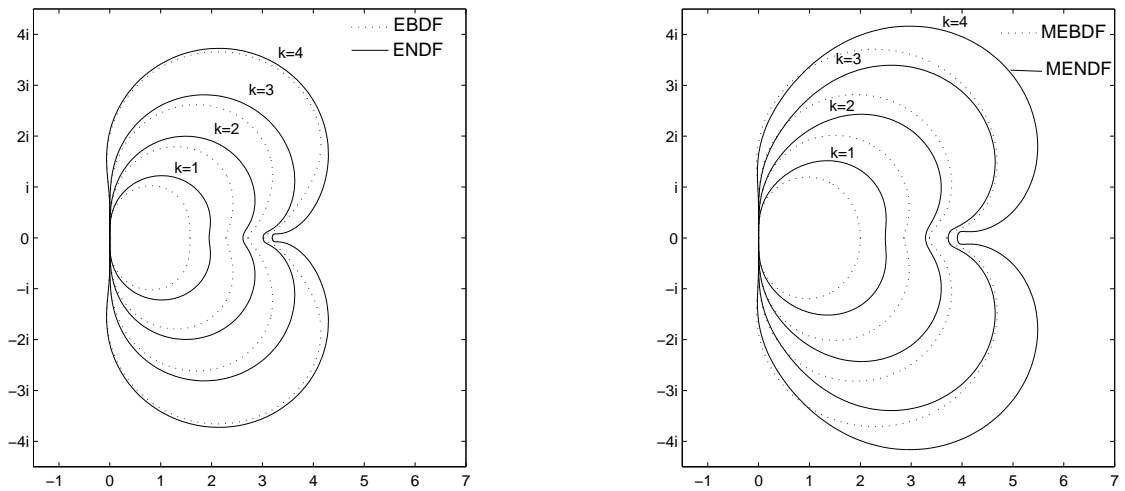


Figura 5.1: Regiones de estab. ENDF y EBDF, MENDF y MEBDF (exterior a curva).

Ejemplo de cálculo de la región de estabilidad ENDF2: Incluimos el cálculo de la región de estabilidad del método ENDF para $k = 2$. El método NDF (5.4) toma la siguiente forma cuando $k = 2$:

$$\hat{\alpha}_0 y_n + \hat{\alpha}_1 y_{n+1} + \hat{\alpha}_2 \bar{y}_{n+2} = h \bar{f}_{n+2} + \kappa \gamma_2 \nabla^3 \bar{y}_{n+2} \quad (5.27)$$

siendo: $\nabla^3 \bar{y}_{n+2} = \bar{y}_{n+2} - 3y_{n+1} + 3y_n - y_{n-1}$.

Aplicamos (5.27) a la ecuación de test ($y' = \lambda y$) y despejamos \bar{y}_{n+2} :

$$\bar{y}_{n+2} = \frac{y_{n+1}(-\hat{\alpha}_1 - 3\kappa\gamma_2) + y_n(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 y_{n-1}}{(\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})} \quad (5.28)$$

Aplicamos el segundo predictor NDF a la ecuación de test y despejamos \bar{y}_{n+3} :

$$\bar{y}_{n+3} = \frac{\bar{y}_{n+2}(-\hat{\alpha}_1 - 3\kappa\gamma_2) + y_{n+1}(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 y_n}{(\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})} \quad (5.29)$$

Sustituyendo el valor de \bar{y}_{n+2} (5.28) en (5.29) obtenemos:

$$\bar{y}_{n+3} = \left[\frac{\left(\overbrace{y_{n+1}(-\hat{\alpha}_1 - 3\kappa\gamma_2) + y_n(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 y_{n-1}}^{\bar{y}_{n+2}} \right) (-\hat{\alpha}_1 - 3\kappa\gamma_2)}{(\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})^2} + \frac{(y_{n+1}(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 y_n) (\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})}{(\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})^2} \right] \quad (5.30)$$

Evaluamos la derivada de \bar{y}_{n+3} : $f(\bar{y}_{n+3}) = \lambda \bar{y}_{n+3}$. Y por último, aplicamos el corrector (5.6) a la ecuación de test:

$$\alpha_0 y_n + \alpha_1 y_{n+1} + \alpha_2 y_{n+2} - \hat{h} \beta_2 y_{n+2} - \hat{h} \beta_3 \bar{y}_{n+3} = 0 \quad (5.31)$$

Sustituyendo $y_m = r^{m+1}$ en (5.31) y dividiendo entre r^n se llega al polinomio característico del método:

$$\begin{aligned} & (\alpha_0 r + \alpha_1 r^2 + \alpha_2 r^3 - \hat{h} \beta_2 r^3) (\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})^2 \\ & - \hat{h} \beta_3 \left[(r^2 (-\hat{\alpha}_1 - 3\kappa\gamma_2) + r (-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2) (-\hat{\alpha}_1 - 3\kappa\gamma_2) \right. \\ & \left. + (r^2 (-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 r) (\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h}) \right] = 0 \quad (5.32) \end{aligned}$$

Y haciendo las operaciones en (5.32) resulta un polinomio de grado 3 de la forma (5.24), siendo sus coeficientes:

$$\begin{cases} \hat{h}^3 : & A = -\beta_2 r^3 \\ \hat{h}^2 : & B = 2(\hat{\alpha}_2 - \kappa\gamma_2) \beta_2 r^3 + T - \beta_3 S \\ \hat{h} : & C = -\beta_2 (\hat{\alpha}_2 - \kappa\gamma_2)^2 r^3 - 2(\hat{\alpha}_2 - \kappa\gamma_2) T + (\hat{\alpha}_2 - \kappa\gamma_2) \beta_3 S - \beta_3 (-\hat{\alpha}_1 - 3\kappa\gamma_2) R \\ \hat{h}^0 : & D = (\hat{\alpha}_2 - \kappa\gamma_2)^2 T \end{cases}$$

$$\text{siendo: } \begin{cases} R = r^2(-\hat{\alpha}_1 - 3\kappa\gamma_2) + r(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 \\ S = r\kappa\gamma_2 - r^2(-\hat{\alpha}_0 + 3\kappa\gamma_2) \\ T = \sum_{j=0}^2 \alpha_j r^{j+1} \end{cases} \quad (5.33)$$

Obtenido el polinomio de estabilidad, se puede dibujar la frontera de la región de estabilidad hallando las raíces del polinomio e imponiendo la condición de que el módulo de éstos sea la unidad, es decir $r = e^{i\theta}$.

Ejemplo de cálculo de la región de estabilidad MENDF2: Para calcular la expresión de la región de estabilidad del método MENDF para $k = 2$, aprovechamos las expresiones de \bar{y}_{n+2} (5.28) y \bar{y}_{n+3} (5.30) que hemos obtenido en el caso anterior tras aplicar el predictor NDF2. Calculamos las derivadas de \bar{y}_{n+2} y de \bar{y}_{n+3} : $f(\bar{y}_{n+2}) = \lambda\bar{y}_{n+2}$, $f(\bar{y}_{n+3}) = \lambda\bar{y}_{n+3}$. Y por último, aplicamos el corrector (5.7) a la ecuación de test obteniendo:

$$\alpha_0 y_n + \alpha_1 y_{n+1} + \alpha_2 y_{n+2} - \hat{h}\hat{\beta}_2 y_{n+2} - \hat{h}\hat{\beta}_3 \bar{y}_{n+3} - \hat{h}(\beta_2 - \hat{\beta}_2)\bar{y}_{n+2} = 0 \quad (5.34)$$

Procediendo como antes, se obtiene el polinomio característico del método:

$$\begin{aligned} & (\alpha_0 r + \alpha_1 r^2 + \alpha_2 r^3 - \hat{h}\hat{\beta}_2 r^3) (\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})^2 \\ & - \hat{h}\hat{\beta}_3 [(r^2(-\hat{\alpha}_1 - 3\kappa\gamma_2) + r(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2)(-\hat{\alpha}_1 - 3\kappa\gamma_2) \\ & + (r^2(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2 r)(\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h})] \\ & - \hat{h}(\beta_2 - \hat{\beta}_2) [r^2(-\hat{\alpha}_1 - 3\kappa\gamma_2) + r(-\hat{\alpha}_0 + 3\kappa\gamma_2) - \kappa\gamma_2] (\hat{\alpha}_2 - \kappa\gamma_2 - \hat{h}) = 0 \end{aligned}$$

Que resulta un polinomio de grado 3 de la forma (5.24), siendo sus coeficientes:

$$\begin{cases} \hat{h}^3 : A = -\hat{\beta}_2 r^3 \\ \hat{h}^2 : B = 2(\hat{\alpha}_2 - \kappa\gamma_2)\hat{\beta}_2 r^3 + T - \beta_3 S + (\beta_2 - \hat{\beta}_2)R \\ \hat{h} : C = -\hat{\beta}_2(\hat{\alpha}_2 - \kappa\gamma_2)^2 r^3 - 2(\hat{\alpha}_2 - \kappa\gamma_2)T + (\hat{\alpha}_2 - \kappa\gamma_2)\beta_3 S \\ \quad \quad \quad - \beta_3(-\hat{\alpha}_1 - 3\kappa\gamma_2)R - (\beta_2 - \hat{\beta}_2)R(\hat{\alpha}_2 - \kappa\gamma_2) \\ \hat{h}^0 : D = (\hat{\alpha}_2 - \kappa\gamma_2)^2 T \end{cases}$$

y siendo R, S, T los mismos que los del método ENDF2 (5.33). \square

Para dibujar la frontera de la región de estabilidad, hay que hallar las raíces del polinomio de estabilidad e imponer la condición de que su módulo sea la unidad: $r = e^{i\theta}$.

5.4.2. Estabilidad de los métodos ENBDF y MENBDF

Aplicando los métodos ENBDF o MENBDF a la ecuación de test $y' = \lambda y$, estableciendo $y_m = r^{m+1}$ para $m = (n-1), n, \dots, (n+k)$ y dividiendo entre r^n , se llega al

k	p (orden)	$A(\alpha)$ EBDF	$A(\alpha)$ EBNDF	$A(\alpha)$ ENBDF	$A(\alpha)$ ENDF
1	2	90°	90°	90°	90°
2	3	90°	90°	90°	90°
3	4	90°	90°	90°	90°
4	5	87.61°	87.68°	87.49°	87.54°

Tabla 5.1: $A(\alpha)$ -estabilidad de los métodos EBDF, EBNDF, ENBDF, ENDF.

k	p (orden)	$A(\alpha)$ MEBDF	$A(\alpha)$ MEBNDF	$A(\alpha)$ MENBDF	$A(\alpha)$ MENDF
1	2	90°	90°	90°	90°
2	3	90°	90°	90°	90°
3	4	90°	90°	90°	90°
4	5	88.36°	88.41°	88.88°	88.93°

Tabla 5.2: $A(\alpha)$ -estabilidad de los métodos MEBDF, MEBNDF, MENBDF, MENDF.

5.4.4. Conclusiones sobre las regiones de estabilidad

- Al igual que los dos métodos de partida (EBDF y MEBDF), los nuevos métodos son A-estables hasta orden 4 ($k = 3$).
- Cuando utilizamos NDFs en el esquema predictor, no se consiguen peores características de estabilidad que cuando utilizamos BDFs, llegando en algunos casos a mejorar las características de estas últimas.
- Para $k = 4$ y en el caso de la familia EBDF, los ángulos de $A(\alpha)$ -estabilidad de los métodos EBDF y EBNDF son mayores que los correspondientes a los métodos ENBDF y ENDF. Por lo que las características de estabilidad son mejores cuando aplicamos BDFs como primer predictor. Además, los ángulos de $A(\alpha)$ -estabilidad del método EBNDF son mayores que el del EBDF y los del método ENDF son mayores que los de ENBDF. De estas dos últimas afirmaciones concluimos que se obtienen mejores características de estabilidad utilizando BDFs como primer predictor y NDFs como segundo. Así, el método EBNDF4 mejora la estabilidad del método EBDF4.
- Para $k = 4$ y en el caso de la familia MEBDF, al utilizar NDFs como predictores del método MEBDF, se obtienen métodos con mayor región de estabilidad que los métodos MEBDF. Esta vez las características de estabilidad son mejores cuando aplicamos NDFs como primer predictor. Y el método que mejores características de estabilidad presenta es el que utiliza dos predictores NDF.
- Las regiones de estabilidad de las MEBDF son mayores que las de los métodos EBDF. De igual manera, los métodos MEBNDF, MENBDF y MENDF tienen mejores propiedades de estabilidad que los métodos EBNDF, ENBDF y ENDF.

5.5. Resultados numéricos

En esta sección recogemos los resultados numéricos obtenidos al resolver algunas EDOs de orden 1 con las familias EBDF y MEBDF.

Ejemplo 1. Hemos considerado el siguiente sistema rígido que también lo resuelve Cash en [16]:

$$\begin{cases} y_1' = -y_1 - 15y_2 + 15e^{-t} \\ y_2' = 15y_1 - y_2 - 15e^{-t} \end{cases}, \text{ con valores iniciales: } (y_1(0), y_2(0)) = (1, 1)^T \quad (5.41)$$

La solución exacta del sistema (5.41) es: $y_1(t) = y_2(t) = e^{-t}$. Los autovalores de la matriz Jacobiana caen cerca del eje imaginario y son $-1 \pm 15i$.

Hemos hallado la solución de este problema en el intervalo temporal $T = [0, 20]$. Por una parte, hemos comparado los métodos ENDF y MENDF de orden 4 ($k = 3$) con el método NDF de orden 4 (esta vez, $k = 4$). Se han dado 100 pasos constantes en los tres casos. Los resultados se han recodigo en la Tabla 5.3 y se puede ver que los métodos ENDF3 y MENDF3 proporcionan buenos resultados, mientras que los obtenidos con el método NDF4 se inestabilizan. También se ve que el método NDF4 requiere alrededor de 600 pasos para obtener errores de similar magnitud que los otros dos métodos.

t	y_i	Solución exacta	Error NDF4 100 pasos	Error NDF4 600 pasos	Error ENDF3 100 pasos	Error MENDF3 100 pasos
5	y_1	$0,673794699908547 \cdot 10^{-2}$	$3,7044 \cdot 10^{-2}$	$3,6748 \cdot 10^{-6}$	$1,8866 \cdot 10^{-7}$	$9,8028 \cdot 10^{-8}$
	y_2	$0,673794699908547 \cdot 10^{-2}$	$1,6667 \cdot 10^{-1}$	$3,1445 \cdot 10^{-6}$	$2,1497 \cdot 10^{-7}$	$2,5754 \cdot 10^{-8}$
10	y_1	$0,453999297624848 \cdot 10^{-4}$	$1,1501 \cdot 10^1$	$2,2845 \cdot 10^{-8}$	$7,2092 \cdot 10^{-10}$	$6,6204 \cdot 10^{-10}$
	y_2	$0,453999297624848 \cdot 10^{-4}$	$7,0102 \cdot 10^{-1}$	$2,993 \cdot 10^{-8}$	$7,3227 \cdot 10^{-10}$	$1,7505 \cdot 10^{-10}$
20	y_1	$0,206115362243856 \cdot 10^{-8}$	$5,0777 \cdot 10^4$	$5,1991 \cdot 10^{-13}$	$3,2552 \cdot 10^{-14}$	$3,0057 \cdot 10^{-14}$
	y_2	$0,206115362243856 \cdot 10^{-8}$	$1,3224 \cdot 10^4$	$2,2219 \cdot 10^{-12}$	$3,3536 \cdot 10^{-14}$	$7,9474 \cdot 10^{-15}$

Tabla 5.3: Resultados de integración del ejemplo 1 con NDF4, ENDF3 y MENDF3.

El mismo sistema lo hemos resuelto con los métodos EBDF3, MEBDF3 y BDF4, todos ellos de orden 4 (Tabla 5.4). Esta vez el método BDF4 tiene que dar 2600 pasos para obtener unos resultados parecidos a los otros. Además, si comparamos los resultados obtenidos con los métodos EBDF3 y MEBDF3 con los obtenidos con ENDF3 y MENDF3, los errores que proporcionan los nuevos métodos son menores.

De todo esto se concluye que, aunque las familias EBDF y MEBDF evalúen tres fórmulas por paso (dos predictores y un corrector) -frente a la única evaluación por paso de los métodos BDF y NDF-, las familias que usan puntos super-futuros resultan más eficientes, puesto que el número de pasos que tienen que dar los métodos BDF y NDF para obtener errores de similar magnitud que las primeras es más que tres veces superior que el número de pasos que dan las primeras.

t	y_i	Error BDF4 100 pasos	Error BDF4 600 pasos	Error BDF4 1600 pasos	Error BDF4 2600 pasos	Error EBDF3 100 pasos	Error MEBDF3 100 pasos
5	y_1	$1,9484 \cdot 10^{-2}$	$5,5994 \cdot 10^{-6}$	$9,263 \cdot 10^{-7}$	$3,5726 \cdot 10^{-7}$	$7,9715 \cdot 10^{-8}$	$1,7398 \cdot 10^{-7}$
	y_2	$1,7076 \cdot 10^{-2}$	$8,1309 \cdot 10^{-6}$	$3,0564 \cdot 10^{-7}$	$1,2509 \cdot 10^{-7}$	$2,7319 \cdot 10^{-7}$	$1,5639 \cdot 10^{-7}$
10	y_1	$9,4972 \cdot 10^{-2}$	$1,5947 \cdot 10^{-8}$	$6,5579 \cdot 10^{-9}$	$2,5451 \cdot 10^{-9}$	$1,4329 \cdot 10^{-9}$	$1,1252 \cdot 10^{-9}$
	y_2	$1,9497 \cdot 10^{-1}$	$1,1192 \cdot 10^{-7}$	$3,9696 \cdot 10^{-10}$	$1,4963 \cdot 10^{-10}$	$1,5083 \cdot 10^{-9}$	$9,2237 \cdot 10^{-10}$
20	y_1	$5,1037 \cdot 10^0$	$1,0442 \cdot 10^{-11}$	$2,036 \cdot 10^{-13}$	$7,6352 \cdot 10^{-14}$	$6,5299 \cdot 10^{-14}$	$5,1083 \cdot 10^{-14}$
	y_2	$1,4313 \cdot 10^1$	$1,0523 \cdot 10^{-11}$	$2,1767 \cdot 10^{-13}$	$8,6876 \cdot 10^{-14}$	$6,833 \cdot 10^{-14}$	$4,1887 \cdot 10^{-14}$

Tabla 5.4: Resultados de integración del ejemplo 1 con BDF4, EBDF3 y MEBDF3.

Ejemplo 2. Consideramos el siguiente sistema de EDOs también resuelto en [44]:

$$\begin{cases} y_1' = -20y_1 - 0,25y_2 - 19,75y_3 \\ y_2' = 20y_1 - 20,25y_2 + 0,25y_3 \\ y_3' = 20y_1 - 19,75y_2 - 0,25y_3 \end{cases}, \text{ siendo: } (y_1(0), y_2(0), y_3(0)) = (1, 0, -1)^T \quad (5.42)$$

cuya solución exacta es:
$$\begin{cases} y_1(t) = \frac{1}{2} (e^{-0,5t} + e^{-20t} (\cos 20t + \sin 20t)) \\ y_2(t) = \frac{1}{2} (e^{-0,5t} - e^{-20t} (\cos 20t - \sin 20t)) \\ y_3(t) = -\frac{1}{2} (e^{-0,5t} + e^{-20t} (\cos 20t - \sin 20t)) \end{cases}$$

Hemos calculado la solución de este sistema en el intervalo $T = [0, 10]$ y dando 50 pasos constantes con los nuevos métodos desarrollados para $k = 3$. Los resultados se han recogido en las Tablas 5.5 y 5.6. Se observa que los resultados obtenidos con los nuevos métodos son superiores.

t	y_i	Solución exacta	Error EBDF	Error EBNDF	Error ENBDF	Error ENDF
1	y_1	$0,303265331217737 \cdot 10^0$	$3,7917 \cdot 10^{-4}$	$2,5018 \cdot 10^{-4}$	$1,1726 \cdot 10^{-4}$	$4,6532 \cdot 10^{-5}$
	y_2	$0,303265330376617 \cdot 10^0$	$1,4307 \cdot 10^{-3}$	$1,2405 \cdot 10^{-3}$	$1,1036 \cdot 10^{-5}$	$8,5423 \cdot 10^{-4}$
	y_3	$-0,303265329336016 \cdot 10^0$	$9,0775 \cdot 10^{-4}$	$7,5899 \cdot 10^{-4}$	$6,2476 \cdot 10^{-4}$	$4,2449 \cdot 10^{-4}$
5	y_1	$0,410424993119494 \cdot 10^{-1}$	$3,6443 \cdot 10^{-5}$	$3,3574 \cdot 10^{-5}$	$3,3307 \cdot 10^{-5}$	$2,9899 \cdot 10^{-5}$
	y_2	$0,410424993119494 \cdot 10^{-1}$	$3,6443 \cdot 10^{-5}$	$3,3574 \cdot 10^{-5}$	$3,3307 \cdot 10^{-5}$	$2,9899 \cdot 10^{-5}$
	y_3	$-0,410424993119494 \cdot 10^{-1}$	$3,6443 \cdot 10^{-5}$	$3,3574 \cdot 10^{-5}$	$3,3307 \cdot 10^{-5}$	$2,9899 \cdot 10^{-5}$
10	y_1	$0,336897349954273 \cdot 10^{-2}$	$3,1059 \cdot 10^{-6}$	$2,8632 \cdot 10^{-6}$	$2,8314 \cdot 10^{-6}$	$2,5425 \cdot 10^{-6}$
	y_2	$0,336897349954273 \cdot 10^{-2}$	$3,1059 \cdot 10^{-6}$	$2,8632 \cdot 10^{-6}$	$2,8314 \cdot 10^{-6}$	$2,5425 \cdot 10^{-6}$
	y_3	$-0,336897349954273 \cdot 10^{-2}$	$3,1059 \cdot 10^{-6}$	$2,8632 \cdot 10^{-6}$	$2,8314 \cdot 10^{-6}$	$2,5425 \cdot 10^{-6}$

Tabla 5.5: Resultados de integración del ejemplo 2 utilizando la familia EBDF.

Ejemplo 3. Consideramos el sistema no lineal de ecuaciones diferenciales [48]:

$$\begin{cases} y_1' = \lambda y_1 + y_2^2 \\ y_2' = -y_2 \end{cases}, \text{ con valores iniciales: } (y_1(0), y_2(0)) = \left(-\frac{1}{\lambda + 2}, 1\right)^T \quad (5.43)$$

t	y_i	Solución exacta	Error MEBDF	Error MEBNDF	Error MENBDF	Error MENDF
1	y_1	$0,303265331217737 \cdot 10^0$	$7,0144 \cdot 10^{-5}$	$1,8023 \cdot 10^{-5}$	$1,5822 \cdot 10^{-3}$	$1,3052 \cdot 10^{-3}$
	y_2	$0,303265330376617 \cdot 10^0$	$6,4172 \cdot 10^{-4}$	$4,5880 \cdot 10^{-4}$	$3,6462 \cdot 10^{-4}$	$2,1813 \cdot 10^{-4}$
	y_3	$-0,303265329336016 \cdot 10^0$	$2,4834 \cdot 10^{-4}$	$1,0859 \cdot 10^{-4}$	$3,2916 \cdot 10^{-5}$	$1,2826 \cdot 10^{-4}$
5	y_1	$0,410424993119494 \cdot 10^{-1}$	$2,7327 \cdot 10^{-5}$	$2,4341 \cdot 10^{-5}$	$2,7694 \cdot 10^{-5}$	$2,4149 \cdot 10^{-5}$
	y_2	$0,410424993119494 \cdot 10^{-1}$	$2,7327 \cdot 10^{-5}$	$2,4341 \cdot 10^{-5}$	$2,7694 \cdot 10^{-5}$	$2,4149 \cdot 10^{-5}$
	y_3	$-0,410424993119494 \cdot 10^{-1}$	$2,7327 \cdot 10^{-5}$	$2,4341 \cdot 10^{-5}$	$2,7694 \cdot 10^{-5}$	$2,4149 \cdot 10^{-5}$
10	y_1	$0,336897349954273 \cdot 10^{-2}$	$2,3204 \cdot 10^{-6}$	$2,0679 \cdot 10^{-6}$	$2,3595 \cdot 10^{-6}$	$2,0593 \cdot 10^{-6}$
	y_2	$0,336897349954273 \cdot 10^{-2}$	$2,3204 \cdot 10^{-6}$	$2,0679 \cdot 10^{-6}$	$2,3595 \cdot 10^{-6}$	$2,0593 \cdot 10^{-6}$
	y_3	$-0,336897349954273 \cdot 10^{-2}$	$2,3204 \cdot 10^{-6}$	$2,0679 \cdot 10^{-6}$	$2,3595 \cdot 10^{-6}$	$2,0593 \cdot 10^{-6}$

Tabla 5.6: Resultados de integración del ejemplo 2 utilizando la familia MEBDF.

siendo $\lambda = 10^4$. La solución exacta del problema es: $y_1(t) = \frac{-e^{-2t}}{(\lambda + 2)}$, $y_2(t) = e^{-t}$

Hemos calculado la solución del sistema (5.43) en el intervalo $T = [0, 5]$, dando 60 pasos y para $k = 4$. Los resultados los hemos recogido en las Tablas 5.7 y 5.8. De nuevo, los errores que se obtienen con los nuevos métodos son menores que los obtenidos con los métodos EBDF y MEBDF.

t	y_i	Solución exacta	Error EBDF	Error EBNDF	Error ENBDF	Error ENDF
3	y_1	$-0,247825652536129 \cdot 10^{-6}$	$2,6602 \cdot 10^{-10}$	$2,4466 \cdot 10^{-10}$	$2,4277 \cdot 10^{-10}$	$2,1752 \cdot 10^{-10}$
	y_2	$0,497870683678639 \cdot 10^{-1}$	$2,6713 \cdot 10^{-5}$	$2,4569 \cdot 10^{-5}$	$2,4379 \cdot 10^{-5}$	$2,1844 \cdot 10^{-5}$
5	y_1	$-0,453908515921664 \cdot 10^{-8}$	$4,866 \cdot 10^{-12}$	$4,4751 \cdot 10^{-12}$	$4,4409 \cdot 10^{-12}$	$3,9787 \cdot 10^{-12}$
	y_2	$0,673794699908547 \cdot 10^{-2}$	$3,6107 \cdot 10^{-6}$	$3,3206 \cdot 10^{-6}$	$3,2953 \cdot 10^{-6}$	$2,9524 \cdot 10^{-6}$

Tabla 5.7: Resultados de integración del ejemplo 3 utilizando la familia EBDF.

t	y_i	Solución exacta	Error MEBDF	Error MEBNDF	Error MENBDF	Error MENDF
3	y_1	$-0,247825652536129 \cdot 10^{-6}$	$1,9890 \cdot 10^{-10}$	$1,7682 \cdot 10^{-10}$	$2,0120 \cdot 10^{-10}$	$1,7511 \cdot 10^{-10}$
	y_2	$0,497870683678639 \cdot 10^{-1}$	$1,9975 \cdot 10^{-5}$	$1,7758 \cdot 10^{-5}$	$2,0206 \cdot 10^{-5}$	$1,7587 \cdot 10^{-5}$
5	y_1	$-0,453908515921664 \cdot 10^{-8}$	$3,6390 \cdot 10^{-12}$	$3,2348 \cdot 10^{-12}$	$3,6807 \cdot 10^{-12}$	$3,2032 \cdot 10^{-12}$
	y_2	$0,673794699908547 \cdot 10^{-2}$	$2,7004 \cdot 10^{-6}$	$2,4006 \cdot 10^{-6}$	$2,7313 \cdot 10^{-6}$	$2,3770 \cdot 10^{-6}$

Tabla 5.8: Resultados de integración del ejemplo 3 utilizando la familia MEBDF.

Capítulo 6

BDF- α : método multipaso con control del amortiguamiento algorítmico

Comprender las cosas que nos rodean es la mejor preparación para comprender las cosas que hay mas allá.

HIPATIA

Al resolver numéricamente el sistema de EDOs de segundo orden que se obtiene tras la semidiscretización de la EDP del tipo onda mediante el Método de Elementos Finitos (MEF), es conveniente eliminar los modos altos que suelen estar asociados a la semidiscretización y que no son representativos. El método HHT- α permite un control paramétrico de la disipación numérica. De manera similar, en este capítulo hemos construido una modificación del método BDF de orden 2 (BDF2), al que hemos llamado BDF- α . El nuevo método es de orden de precisión 2 y con una constante de error menor que el método BDF2, es A-estable para algunos valores del parámetro α y permite controlar paramétricamente la disipación numérica. De esta manera, hemos conseguido un método para sistemas de EDOs de orden 1 con propiedades similares que el método HHT- α [43] en problemas vibratorios.

6.1. Radios espectrales de los métodos BDF

Si estamos interesados en el estudio del factor de amplificación de los métodos BDF, aplicamos el método a la ecuación de test $y' = \lambda y$, y escribimos el resultado en la siguiente forma:

$$Y_{n+k} = A(\hat{h}) Y_{n+k-1} \quad (6.1)$$

siendo $\hat{h} = h\lambda$, $Y_{n+k} = (y_{n+1}, y_{n+2}, \dots, y_{n+k})^T$, $Y_{n+k-1} = (y_n, y_{n+1}, \dots, y_{n+k-1})^T$ y A es la matriz de amplificación (matriz cuadrada de dimensión $k \times k$). Se calculan los autovalores de la matriz A y el mayor en módulo es el radio espectral:

$$\rho(A) = \max \{ |\lambda_i| : \lambda_i \text{ autovalor de } A \} \quad (6.2)$$

El radio espectral está estrechamente unido a la estabilidad del método y es necesario que se verifique $\rho(A) \leq 1$, para prevenir que A^n se amplifique según aumenta n .

Cuando estamos trabajando con EDOs de segundo orden, tenemos que considerar su ecuación de test $u'' + \omega^2 u = 0$, el cual representa un sistema físico vibratorio no amortiguado con frecuencia natural $f = \omega/(2\pi)$. Transformamos esta ecuación en su sistema equivalente de dos EDOs de orden 1:

$$\begin{pmatrix} u \\ u' \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} \begin{pmatrix} u \\ u' \end{pmatrix} \quad (6.3)$$

El sistema (6.3) se puede desacoplar en dos ecuaciones independientes de la forma:

$$y' = \pm i\omega y \quad (6.4)$$

y la aplicación de la teoría general, (6.1) y (6.2), a la ecuación de segundo orden $u'' + \omega^2 u = 0$, es equivalente a aplicar esta teoría a la ecuación de primer orden $y' = \lambda y$, pero esta vez teniendo en cuenta que $\lambda = i\omega$.

En este tipo de problemas, el amortiguamiento algorítmico está relacionado con el radio espectral (6.2). Por ejemplo, es fácil ver que el método trapezoidal es un método lineal multipaso cuyo radio espectral no depende de la frecuencia, es decir, $\rho = 1$. Esto significa que este método no muestra amortiguamiento algorítmico (Figura 6.1). No obstante, los métodos BDF tienen un radio espectral que disminuye al aumentar la frecuencia (ver Figura 6.2), lo que significa que los métodos BDF tienen amortiguamiento algorítmico (Figura 6.1).

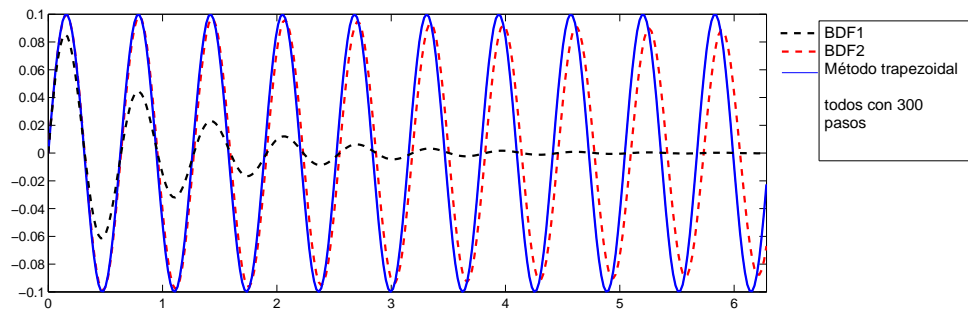


Figura 6.1: Amort. algorít. varios métodos en $y'' = -10^2 y$, $(y(0), y'(0)) = (0, 1)$.

Ejemplo de cálculo del factor de amplificación: Calcularemos el factor de amplificación del método BDF4. Este método viene dado por:

$$y_{n+4} - \frac{48}{25}y_{n+3} + \frac{36}{25}y_{n+2} - \frac{16}{25}y_{n+1} + \frac{3}{25}y_n = h\frac{12}{25}f_{n+4} \quad (6.5)$$

Aplicamos el método (6.5) a la ecuación de test $y' = \lambda y$:

$$y_{n+4} - \frac{48}{25}y_{n+3} + \frac{36}{25}y_{n+2} - \frac{16}{25}y_{n+1} + \frac{3}{25}y_n = \hat{h}\frac{12}{25}y_{n+4} \quad (6.6)$$

siendo: $\hat{h} = \lambda h$. La expresión (6.6) matricialmente se escribe de esta manera:

$$\bar{A}_1 \cdot Y_{n+4} = \bar{A}_2 \cdot Y_{n+3} \Rightarrow Y_{n+4} = AY_{n+3} \quad (6.7)$$

siendo:

$$\left\{ \begin{array}{l} \bar{A}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 - \frac{12}{25}\hat{h} \end{pmatrix}, \quad \bar{A}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{3}{25} & \frac{16}{25} & -\frac{36}{25} & \frac{48}{25} \end{pmatrix}, \\ Y_{n+3} = (y_n, y_{n+1}, y_{n+2}, y_{n+3})^T, \quad Y_{n+4} = (y_{n+1}, y_{n+2}, y_{n+3}, y_{n+4})^T \end{array} \right. \quad (6.8)$$

La matriz $A = \bar{A}_1^{-1}\bar{A}_2$ es el factor de amplificación del método (6.5). \square

La notación que hemos utilizado en toda la tesis ha sido $\hat{h} = h\lambda$ en el caso de EDOs de orden 1 y $\Omega = h\omega$ para EDOs de orden 2, aunque ambos valores representen lo mismo ($\Omega \equiv \hat{h}$). En las Figuras de radios espectrales que se incluyen en este Capítulo, en el eje de las abscisas se ha representado $\Omega/2\pi$ siendo $\Omega = h\omega$ en el caso de EDOs de orden 2 y en EDOs de orden 1, $\Omega = h\lambda = \hat{h}$ con $\lambda = i\omega$.

En el caso particular de métodos para EDOs de orden 2, si queremos estudiar cómo el radio espectral depende de las frecuencias, hay que evaluar $\rho(A(\hat{h}))$ para diferentes valores de $\hat{h} = \lambda h$, teniendo en cuenta que en este caso específico $\lambda = i\omega$. De esta manera, podemos calcular el valor del radio espectral de los métodos BDF y de otros en función de $\omega h/(2\pi) = h/T$, como se muestra en la Figura 6.2. En esta figura se puede ver que los radios espectrales de los métodos BDF tienden a cero cuando $h/T \rightarrow +\infty$. También se observa que en el caso de BDFs de órdenes 3, 4 y 5, hay una zona en la que el radio espectral se hace mayor que la unidad. Ello es debido a que estos métodos no son estables en cierta zona del eje imaginario.

Al resolver una EDO de segundo orden, medidas de precisión como el amortiguamiento algorítmico $\bar{\xi}$ y el error relativo en el periodo $(\bar{T} - T)/T$ son importantes, ver Figura 6.1. El primero de los dos $\bar{\xi}$, dice cómo decae la amplitud y la segunda mide cuánto

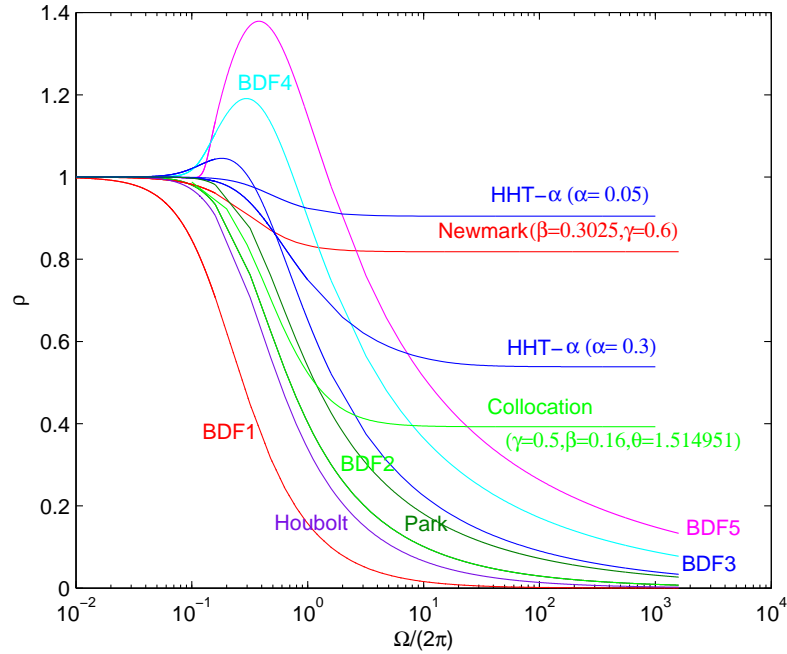


Figura 6.2: Radios espectrales de los métodos BDF.

aumenta o decrece el periodo. Se pueden calcular ambos parámetros utilizando las expresiones (3.87), (3.88) y (3.89) y mediante evaluaciones computacionales.

En [47] además de estudiar la algorítmica de los métodos clásicos de la mecánica computacional para EDOs de orden 2, se estudia un método numérico aplicable a EDOs de orden 1: el método de Park. El método de Park es un método multipaso A-estable de orden de precisión 2 y que viene dado por:

$$-y_{n+3} + \frac{15}{10}y_{n+2} - \frac{6}{10}y_{n+1} + \frac{1}{10}y_n + h\frac{6}{10}f_{n+3} = 0 \quad (6.9)$$

Al aplicar este método a la ecuación de test, la expresión (6.1) resulta:

$$\begin{pmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \end{pmatrix} = A \begin{pmatrix} y_n \\ y_{n+1} \\ y_{n+2} \end{pmatrix} \quad (6.10)$$

siendo: $A = \bar{A}_1^{-1}\bar{A}_2$, $\bar{A}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 - \frac{6}{10}h\lambda \end{pmatrix}$, $\bar{A}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{10} & -\frac{6}{10} & \frac{15}{10} \end{pmatrix}$

En la Figura 6.2 también se ha incluido el radio espectral de este método y se puede ver que retiene buena precisión en las frecuencias bajas y que tiene buenas características de disipación en las frecuencias altas.

A la hora de resolver EDOs de segundo orden que provienen de la semidiscretización MEF de la EDP del tipo onda, los modos más altos son fruto del proceso de discretización y no son representativos de la EDP que gobierna. En estos casos, suele ser conveniente introducir amortiguamiento algorítmico que aniquile la participación de las componentes modales de alta frecuencia. A pesar de que el valor mínimo que muestran los métodos BDF es el más efectivo para eliminar la participación de los modos altos ($\rho_\infty = \lim_{h/T \rightarrow \infty} \rho(A)$), es más conveniente tener un método numérico que permita un control paramétrico de la disipación numérica, permitiendo la participación de las frecuencias de rango medio. La Figura 6.2 muestra cómo el control paramétrico $\alpha = 0,3$ del método HHT- α disminuye el amortiguamiento algorítmico de las frecuencias de rango medio respecto del método BDF, a la vez que mantiene un radio espectral bajo en el rango de las frecuencias altas, la cual es suficiente para eliminar rápidamente estas componentes.

Utilizando el mismo procedimiento que con los métodos BDF, es posible calcular el factor de amplificación de otros métodos para EDOs de orden 1. Es lo que hemos hecho con los métodos EBDF, cuyos radios espectrales los hemos representado en la Figura 6.3. En esta figura observamos que también los radios espectrales de los métodos EBDF tienden a cero cuando $h/T \rightarrow +\infty$.

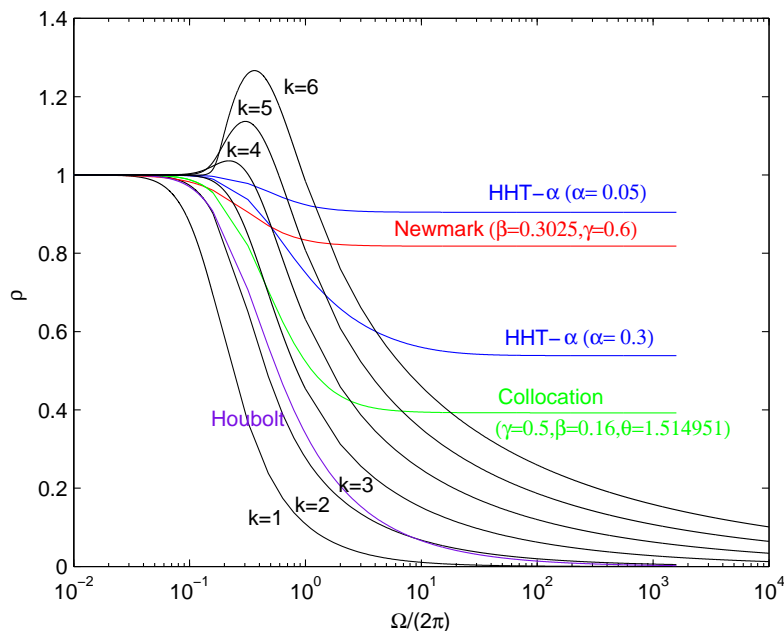


Figura 6.3: Radios espectrales de los métodos EBDF.

6.2. Método BDF- α

El método BDF2 presenta una disipación fuerte de las frecuencias medias y altas, Figura 6.2. Nuestro objetivo es construir un método numérico que permita un control paramétrico del amortiguamiento algorítmico, de igual manera que el método HHT- α .

Son éstas algunas consideraciones sobre la modificación planteada:

1. El nuevo método tiene que estar basado en el método BDF2:

$$\frac{3}{2}y_{n+2} - 2y_{n+1} + \frac{1}{2}y_n = hf_{n+2} \quad (6.11)$$

y si es posible tiene que mantener el orden de precisión y las características de estabilidad de éste: orden 2 y A-estable (incondicionalmente estable). Además, el nuevo método debe permitir obtener un rango mayor de radios espectrales ρ_∞ que el que se obtiene con el método BDF2.

2. La primera modificación que hemos considerado consiste en ponderar cada uno de los sumandos del BDF2. De esta manera se obtiene una expresión general en la que se evalúan y , y' en los instantes t_n, t_{n+1}, t_{n+2} :

$$\alpha_1 y_{n+2} + \alpha_2 y_{n+1} + \alpha_3 y_n = \beta_1 f_{n+2} + \beta_2 f_{n+1} + \beta_3 y_n \quad (6.12)$$

Hemos descartado esta opción por dos razones: por una parte, por no ser una propuesta equivalente al binomio Newmark-HHT- α y por otra parte, porque la expresión (6.12) consta de todos los sumandos que aparecen en los métodos de Adams Moulton de orden 3 y BDF2, por lo que estaríamos planteando una combinación lineal de ambos métodos que en principio no es de nuestro interés.

$$\begin{cases} \text{Adams Moulton } (k=3): y_{n+2} = y_{n+1} + h \left(\frac{5}{12} f_{n+2} + \frac{2}{3} f_{n+1} - \frac{1}{12} f_n \right) \\ \text{BDF } (k=2): \frac{3}{2} y_{n+2} - 2y_{n+1} + \frac{1}{2} y_n = hf_{n+2} \end{cases} \quad (6.13)$$

3. La forma en la que se construyó la modificación HHT- α nos puede ayudar a elaborar nuestra propuesta. El método de Newmark viene dado por las fórmulas de avance (3.60) y en el método HHT- α se mantienen las dos últimas expresiones de (3.60) y se hace una ponderación de la primera:

$$\text{Newmark: } ma_{n+1} + cv_{n+1} + kd_{n+1} = f_{n+1} \quad (6.14)$$

$$\text{HHT-}\alpha: ma_{n+1} + (1-\alpha)cv_{n+1} + \alpha cv_n + (1-\alpha)kd_{n+1} + \alpha kd_n = f(t_{n+1-\alpha}) \quad (6.15)$$

siendo: $t_{n+1-\alpha} = (1-\alpha)t_{n+1} + \alpha t_n$

4. Partiendo del método BDF2, una posibilidad es ponderar únicamente la parte derecha de la expresión (6.11):

$$\frac{3}{2}y_{n+2} - 2y_{n+1} + \frac{1}{2}y_n = h[(1+\alpha)f_{n+2} - \alpha f_{n+1}] \quad (6.16)$$

El inconveniente del método (6.16) es que cuando $\alpha \neq 0$ el método es de orden 1 y cuando $\alpha = 0$ se tiene el método BDF2.

5. Finalmente, hemos considerado una modificación con tres parámetros libres α , β y γ :

$$\frac{3}{2}((1 + \beta)y_{n+2} - \beta y_{n+1}) - 2((1 + \gamma)y_{n+1} - \gamma y_n) + \frac{1}{2}y_n = h((1 + \alpha)f_{n+2} - \alpha f_{n+1}) \quad (6.17)$$

Reagrupando términos en (6.17) tenemos la expresión de nuestra modificación:

$$y_{n+2} \frac{3}{2}(1 + \beta) + y_{n+1} \left(-\frac{3}{2}\beta - 2(1 + \gamma) \right) + y_n \left(2\gamma + \frac{1}{2} \right) = h(1 + \alpha)f_{n+2} - h\alpha f_{n+1} \quad (6.18)$$

La expresión (6.18) se puede escribir de la siguiente forma abreviada que corresponde a los métodos lineales multipaso o de k -pasos (2.3):

$$\sum_{j=0}^2 \alpha_j y_{n+j} = h \sum_{j=0}^2 \beta_j f_{n+j} \quad (6.19)$$

$$\text{siendo: } \begin{cases} \alpha_2 = \frac{3}{2}(1 + \beta), & \alpha_1 = -\frac{3}{2}\beta - 2(1 + \gamma), & \alpha_0 = 2\gamma + \frac{1}{2} \\ \beta_2 = 1 + \alpha, & \beta_1 = -\alpha, & \beta_0 = 0 \end{cases} \quad (6.20)$$

6.2.1. Orden

Teorema. Las condiciones que tienen que cumplir las constantes α , β , γ para que el método dado por (6.18) sea de orden 2 son las siguientes: $\alpha = \frac{3}{2}\beta = 2\gamma$. Así, el método BDF- α de orden 2 viene dado por:

$$\left(\frac{3}{2} + \alpha \right) y_{n+2} + (-2 - 2\alpha) y_{n+1} + \left(\frac{1}{2} + \alpha \right) y_n = h(1 + \alpha)f_{n+2} - h\alpha f_{n+1} \quad (6.21)$$

Demostración. El método BDF- α (6.18) es un método lineal multipaso (2.3), siendo las constantes del método las dadas por (6.20).

El método BDF- α será de orden 2 si $C_0 = C_1 = C_2 = 0$ y $C_3 \neq 0$ [13, 52], siendo C_i las constantes dadas por (2.19). Sustituyendo los valores (6.20) en (2.19) se tiene:

$$\begin{cases} C_0 = \sum_{i=0}^2 \alpha_i = 0 \\ C_1 = \sum_{i=0}^2 i\alpha_i - \sum_{i=0}^2 \beta_i = -2\gamma + \frac{3}{2}\beta \\ C_2 = \frac{1}{2!} \left(\sum_{i=0}^2 i^2 \alpha_i \right) - \left(\sum_{i=0}^2 i\beta_i \right) = -\gamma + \frac{9}{4}\beta - \alpha \\ C_3 = \frac{1}{3!} \left(\sum_{i=0}^2 i^3 \alpha_i \right) - \frac{1}{2!} \left(\sum_{i=0}^2 i^2 \beta_i \right) = \frac{7}{4}\beta - \frac{1}{3} - \frac{\gamma}{3} - \frac{3}{2}\alpha \end{cases} \quad (6.22)$$

Para cualquier valor de los parámetros α, β, γ se cumple $C_0 = 0$. El método BDF- α será de orden 2 si además $C_1 = C_2 = 0$ y esto sucede cuando α, β, γ cumplen: $\alpha = \frac{3}{2}\beta = 2\gamma$. Por lo que queda demostrado que el método BDF- α es de orden 2. \square

Obsérvese que tomando $\alpha = 0$, (6.21) se convierte en la expresión del método BDF2 y cuando $\alpha = -0,5$ se obtiene el método trapezoidal. Cuando $\alpha = \frac{a}{1-a}$ siendo $a \in \mathbb{R}$ el método (6.21) coincide con el método A-BDF [28] para $k = 2$.

El error local de truncamiento del método BDF- α viene dado por:

$$LTE = Ch^3 y'''(t_n) + O(h^4) \quad (6.23)$$

siendo C la constante de error del método:

$$C = \frac{C_3}{\sigma(1)} \quad (6.24)$$

Tras sustituir en C_3 la condición $\alpha = \frac{3}{2}\beta = 2\gamma$ para que el método sea de orden 2 tenemos:

$$C_3 = \frac{7}{4}\beta - \frac{1}{3} - \frac{\gamma}{3} - \frac{3}{2}\alpha = \frac{-2 - 3\alpha}{6} \quad (6.25)$$

El polinomio característico $\sigma(r)$ (2.3) en este caso queda:

$$\sigma(r) = \sum_{j=0}^2 \beta_j r^j = -\alpha r + (1 + \alpha)r^2 \Rightarrow \sigma(1) = -\alpha + 1 + \alpha = 1 \quad (6.26)$$

Sustituyendo (6.25) y (6.26) en (6.24) tenemos la constante de error del método BDF- α :

$$C = \frac{-2 - 3\alpha}{6} \quad (6.27)$$

En el caso $\alpha = 0$, tenemos el método BDF2 y se consigue su constante de error $C = -\frac{1}{3}$. Para $\alpha = -0,5$, tenemos el método trapezoidal y se obtiene la constante de error conocida: $C = -\frac{1}{12}$. En la Figura 6.4 se muestra el valor absoluto de la constante de error del método BDF- α en función de α . El método BDF- α es un método de orden 2, con una constante de error menor que la del método BDF2 si $\alpha \in [-0,5, 0)$.

En la siguiente sección veremos que el método BDF- α es A-estable para $\alpha \geq -0,5$, por lo que el método BDF- α correspondiente a $\alpha = -0,5$ es el método A-estable que menor constante de error tiene. Es decir, que se cumple el teorema de Dahlquist, que dice que un método multipaso A-estable es de orden $p \leq 2$, siendo el método trapezoidal el método A-estable de orden 2 con menor constante de error [37].

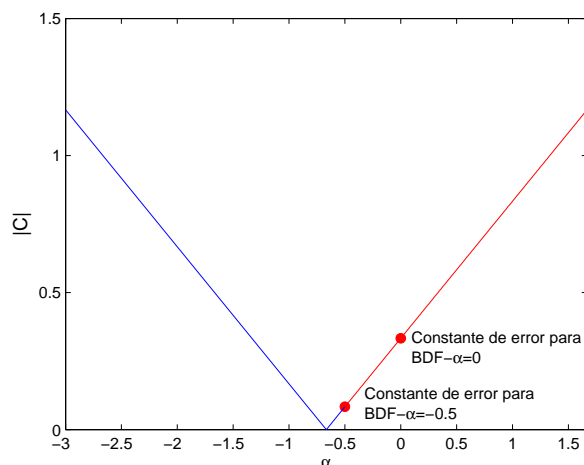


Figura 6.4: Constantes de error del método BDF- α en valor absoluto ($|C|$).

6.2.2. Regiones de estabilidad

Para hacer el estudio de estabilidad del método BDF- α , aplicamos el método (6.21) a la ecuación de test $y' = \lambda y$, obteniendo:

$$\left(\frac{3}{2} + \alpha\right) y_{n+2} + (-2 - 2\alpha) y_{n+1} + \left(\frac{1}{2} + \alpha\right) y_n = \hat{h}(1 + \alpha) y_{n+2} - \hat{h}\alpha y_{n+1} \quad (6.28)$$

siendo $\hat{h} = \lambda h$. Sustituimos $y_m = r^m$ en (6.28) para obtener el polinomio característico del método, y despejando \hat{h} tenemos:

$$\hat{h} = \frac{\left(\frac{3}{2} + \alpha\right) r^2 + (-2 - 2\alpha) r + \left(\frac{1}{2} + \alpha\right)}{(1 + \alpha)r^2 - \alpha r} \quad (6.29)$$

Sustituyendo $r = e^{i\theta}$ en (6.29) tenemos la frontera de la región de estabilidad:

$$\hat{h}(\theta) = \frac{(1 + 2\alpha)(\cos\theta - 1)^2 + i\sin\theta \left[(1 + 2\alpha)(1 - \cos\theta) + \frac{1}{1+\alpha} \right]}{(1 + \alpha) \left[\left(\cos\theta - \frac{\alpha}{1+\alpha} \right)^2 + \sin^2\theta \right]} \quad (6.30)$$

Para $\alpha \geq -0,5$ el denominador de $\hat{h}(\theta)$ está acotado por debajo por una constante positiva. Por lo que, fijando $\alpha \geq -0,5$, para un número real suficientemente grande que depende de α e independiente de θ , $R(\alpha) \in \mathbb{R}$, la parte real de $\hat{h}(\theta)$ verifica:

$$0 \leq \text{Re}(\hat{h}(\theta)) \leq R(\alpha) \quad (6.31)$$

La desigualdad (6.31) implica que la frontera de la región de estabilidad $\hat{h}(\theta)$ cae en un subconjunto compacto del semiplano derecho \mathbb{C}^+ . Para cualquier $\hat{h} \in \mathbb{C}^-$, se consigue estabilidad absoluta y por razones de continuidad se tiene que \mathbb{C}^- pertenece a la región de estabilidad [28]. Esto significa que el método es A-estable cuando $\alpha \in [-0,5, +\infty)$. Se han representado en la Figura 6.5 las regiones de estabilidad BDF- α para algunos valores de α .

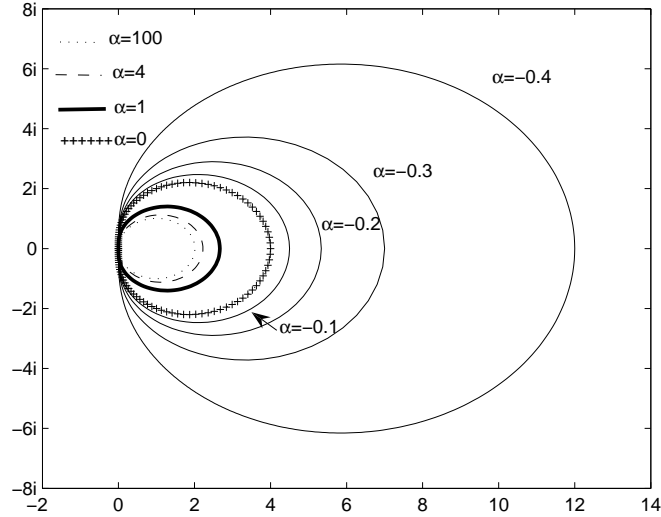


Figura 6.5: Regiones de estabilidad del método BDF- α (exterior a curva).

6.2.3. Factor de amplificación

La expresión (6.28) que hemos obtenido tras aplicar el método BDF- α a la ecuación de test se puede expresar matricialmente:

$$Y_{n+2} = AY_{n+1} \quad (6.32)$$

$$\text{siendo: } \begin{cases} Y_{n+2} = (y_{n+1}, y_{n+2})^T, Y_{n+1} = (y_n, y_{n+1})^T, A = \bar{A}_1^{-1} \bar{A}_2 \\ \bar{A}_1 = \begin{pmatrix} 1 & 0 \\ 0 & \frac{3}{2} + \alpha - \hat{h}(1 + \alpha) \end{pmatrix}, \bar{A}_2 = \begin{pmatrix} 0 & 1 \\ -\frac{1}{2} - \alpha & 2 + 2\alpha - \hat{h}\alpha \end{pmatrix} \end{cases} \quad (6.33)$$

La matriz A es la matriz de amplificación y sus autovalores son:

$$\lambda_{1,2} = \frac{-2 - 2\alpha + \hat{h}\alpha \pm \sqrt{\hat{h}^2\alpha^2 + 2\hat{h}(\alpha + 1) + 1}}{-3 - 2\alpha + 2\hat{h}(1 + \alpha)} \quad (6.34)$$

Para caracterizar la disipación numérica del método en función del parámetro α , calculamos el valor del radio espectral en $\hat{h} \rightarrow \infty$:

$$\rho_\infty = \lim_{\hat{h} \rightarrow \infty} \max \{|\lambda_1, \lambda_2|\} = \max \left\{ \left| \frac{\alpha \pm |\alpha|}{2 + 2\alpha} \right| \right\} = \begin{cases} \frac{-2\alpha}{-2-2\alpha} > 1, & \alpha \in (-\infty, -1) \\ \frac{-2\alpha}{2+2\alpha}, & \alpha \in (-1, 0] \\ \frac{2\alpha}{2+2\alpha} < 1, & \alpha \in [0, +\infty) \end{cases} \quad (6.35)$$

En el intervalo $\alpha \in (-\infty, -1)$, ρ_∞ es mayor que 1 (anteriormente ya hemos visto que en este caso el método no es A-estable). Para $\alpha \in (-1, 0]$ la expresión (6.35) toma los valores:

$$\rho_\infty = \frac{-2\alpha}{2+2\alpha} = \begin{cases} \frac{-2\alpha}{2+2\alpha} > 1, & \alpha \in (-1, -0,5) \\ 1, & \alpha = -0,5 \\ \frac{-2\alpha}{2+2\alpha} < 1, & \alpha \in [-0,5, 0) \end{cases} \quad (6.36)$$

De nuevo, observamos que $\rho_\infty > 1$ cuando $\alpha \in (-1, -0,5)$.

Para el método BDF- α A-estable, y esto sucede cuando $\alpha \geq -0,5$, el valor ρ_∞ viene dado por:

$$\rho_\infty = \begin{cases} 1, & \alpha = -0,5 \\ \frac{-2\alpha}{2+2\alpha} < 1, & \alpha \in [-0,5, 0) \\ \frac{2\alpha}{2+2\alpha} < 1, & \alpha \in [0, +\infty) \end{cases} \quad (6.37)$$

y de esta última expresión es fácil concluir que en el caso del BDF- α A-estable, ρ_∞ toma todos los valores del intervalo $(0, 1]$. Esto significa que fijando un valor de $\rho_\infty \in (0, 1]$, es posible obtener un método BDF- α de orden 2 y A-estable que tiene ese radio espectral en $\hat{h} \rightarrow \infty$. Por lo que, nuestro método permite un control paramétrico de la disipación numérica. En la Figura 6.6 se muestran los radios espectrales de algunos métodos BDF- α en función de $\Omega/(2\pi)$.

Hemos calculado el valor del parámetro α del método BDF- α con igual radio espectral que el método HHT- α cuando $\Omega/(2\pi) \rightarrow +\infty$. Lo hemos hecho para los casos HHT- $\alpha = 0,05$ y HHT- $\alpha = 0,3$. El radio espectral del método HHT- $\alpha = 0,05$ en $\Omega/(2\pi) \rightarrow +\infty$ es $\rho = 0,90476$. Cogiendo $\alpha_1 = -0,475065$ o $\alpha_2 = 9,50$ en el método BDF- α , se obtiene el mismo radio espectral en $\Omega/(2\pi) \rightarrow +\infty$. Escribimos $\alpha_1 = -0,475065$ y no $\alpha_1 = -0,47$ porque los radios espectrales correspondientes a estos dos son significativamente distintos. Aunque los radios espectrales de los métodos BDF- $\alpha = -0,475065$ y HHT- $\alpha = 0,05$ son muy parecidos no son exactamente iguales, ver Figura 6.7. En el caso de HHT- $\alpha = 0,3$, el valor del radio espectral en $\Omega/(2\pi) \rightarrow +\infty$ es $\rho = 0,53846$. El método BDF- α tiene el mismo radio espectral en $\Omega/(2\pi) \rightarrow +\infty$ cuando $\alpha_1 = -0,35$ o $\alpha_2 = 1,17$.

6.2.4. Error relativo en el periodo y amortiguamiento algorítmico

En las Figuras 6.8 y 6.9 se muestran el amortiguamiento algorítmico y el error relativo en el periodo de algunos métodos BDF- α . En la imagen izquierda de la Figura 6.8

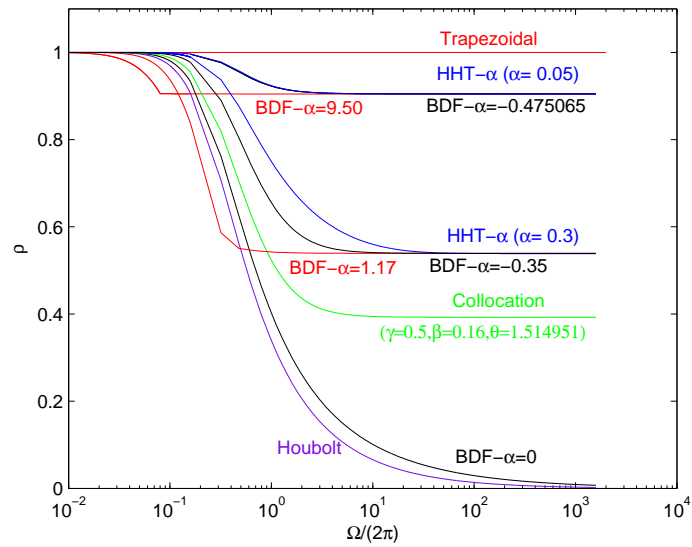


Figura 6.6: Radios espectrales del método BDF- α .

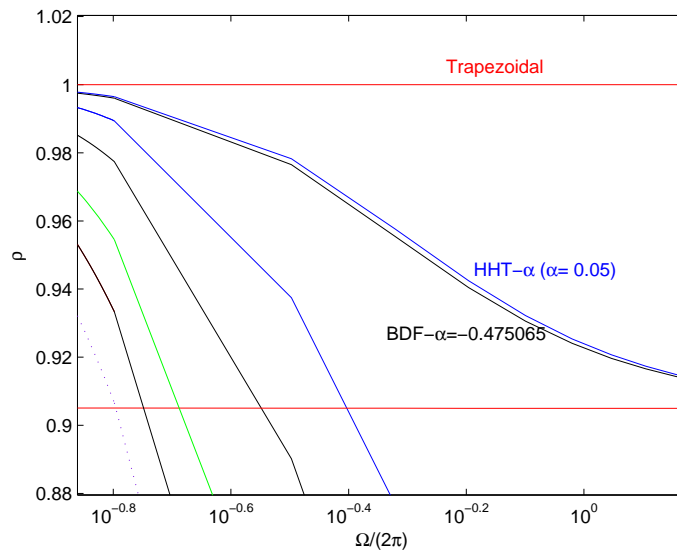


Figura 6.7: Radios espectrales del método BDF- α (detalle de la Figura 6.6).

se puede ver que cuanto mayor es el valor de α , mayor resulta el amortiguamiento algorítmico. Se obtiene un amortiguamiento algorítmico nulo con $\alpha = -0,5$ en el método BDF- α (método trapezoidal). Sucede lo mismo con el error relativo en el periodo: cuanto mayor es el valor de α , mayor resulta el error relativo en el periodo.

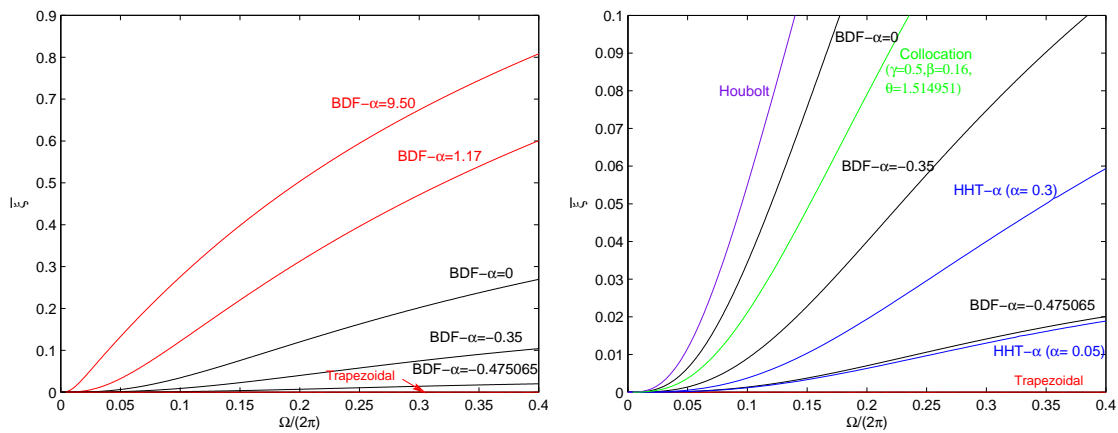


Figura 6.8: Amortiguamiento algorítmico del método BDF- α .

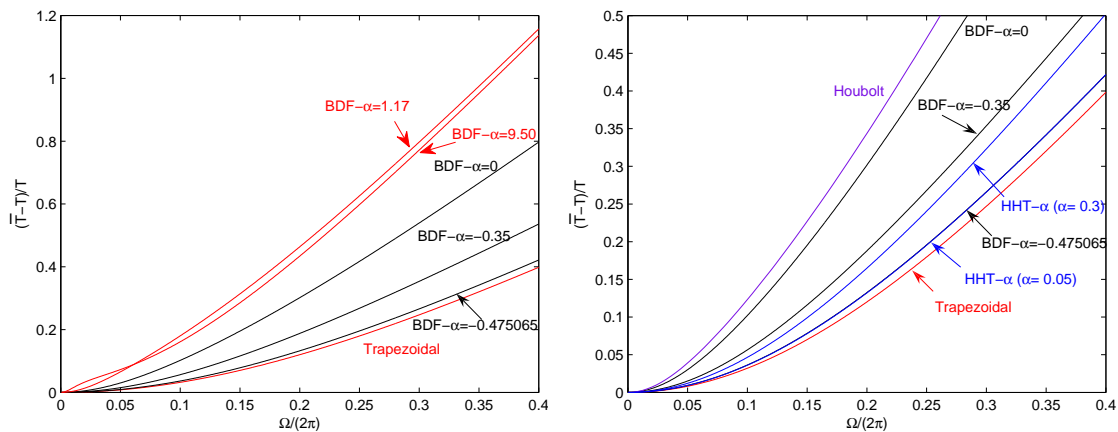


Figura 6.9: Error relativo en el periodo del método BDF- α .

Si comparamos las gráficas del amortiguamiento algorítmico correspondientes al método HHT con $\alpha = 0,05$ y $\alpha = 0,3$ con sus respectivos en el método BDF- α (BDF- $\alpha = -0,475065$ y BDF- $\alpha = -0,35$), vemos que los de los métodos BDF- α son mayores que los del método HHT- α . Sucede lo mismo con los errores relativos en el periodo (Figuras 6.9 y 6.10).

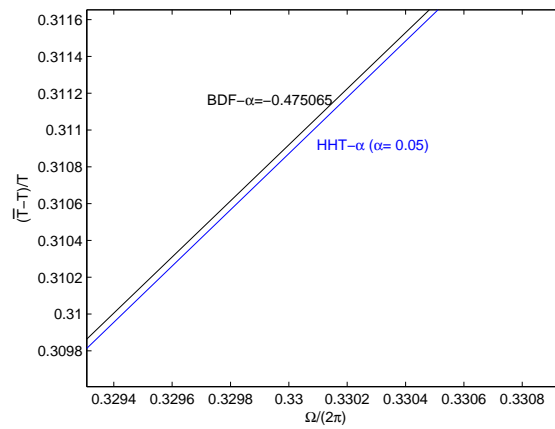


Figura 6.10: Error relativo en el periodo del método BDF- α (detalle de la Figura 6.9).

Capítulo 7

Metodología y resultados

Invertir en conocimientos produce siempre los mejores beneficios.

BENJAMIN FRANKLIN

El objetivo de este capítulo es describir la metodología orientada a objeto que se ha empleado para la experimentación de los algoritmos numéricos de integración temporal con distintos problemas lineales y no lineales. A partir de una base ya desarrollada para resolver problemas cuasi-estáticos no lineales, Laplace-Poisson y Elasticidad, se han implementado distintos métodos de integración temporal de paso constante (tales como Runge-Kutta, trapezoidal clásico, el método HHT- α , BDF, NDF, BDF- α , familia EBDF y familia MEBDF), y también de paso variable incorporando las funciones de la *odesuite* de Matlab.

A continuación se expone la arquitectura orientada a objeto resultante, en la que se distinguen tres partes:

- la parte relacionada con el cálculo de las integrales planteadas sobre la malla y que se calculan en el *objeto malla*,
- la relacionada con los aspectos específicos del problema tratado, que se resuelven mediante el *objeto problema*,
- y la relacionada con el método de resolución (lineal, no lineal o de integración paso a paso en el tiempo) que se desarrolla en el *objeto método*.

Una imagen de la organización de los objetos resultantes se muestra en la Figura 7.1. La jerarquía del organigrama es por agregación. El objeto método elementos finitos (*objmef*), que es el objeto principal, contiene los objetos malla (*objmalla*), problema (*objprob*) y método (*objmetodo*); *objmalla* contiene los objetos elemento finito (*objelef*), material (*objmat*) y campo U (*objU*),... y así sucesivamente.

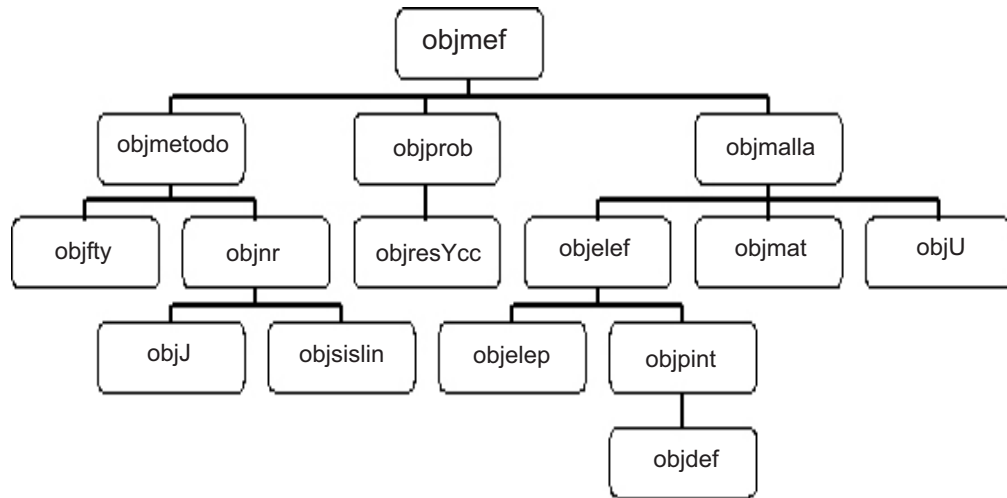


Figura 7.1: Arquitectura del objeto principal por agregación de objetos subordinados.

Abordaremos su descripción en tres ejemplos, de menor a mayor complejidad:

1. Problema lineal, donde se describe principalmente el objeto malla.
2. Problema no lineal, donde se completa la explicación del objeto problema.
3. Problema dinámico lineal y no lineal, donde se describe con más detalle el objeto método y sus variantes, que ha sido desarrollado en este trabajo.

7.1. Problema lineal. El objeto malla

El problema lineal es el más sencillo. Basta con construir la matriz de coeficientes y el término independiente del sistema lineal de ecuaciones y para ello calcular las integrales definidas sobre la malla (*objmalla*). El objeto problema (*objprob*) organiza los cálculos, ordena la resolución del sistema lineal y postprocesa los resultados.

7.1.1. Resumen de la formulación

Exponemos en primer lugar un resumen de la formulación matemática de la aproximación mediante el MEF de la EDP de Laplace hasta plantear el sistema de ecuaciones (no lineal en general) cuya resolución proporciona los valores nodales de la aproximación. Este resumen será válido para la descripción del problema no lineal de la siguiente sección (Sección 7.2). En esta sección consideraremos una ecuación constitutiva lineal, para obtener el sistema lineal de ecuaciones del problema de Laplace, que extenderemos a continuación al problema elástico lineal.

También queda resumido el alisado de flujos y tensiones para el postproceso de los resultados mediante minimización cuadrática, que será aplicable en esta y en las siguientes secciones: resolución del problema estacionario no lineal Sección 7.2, y del problema dinámico lineal y no lineal 7.3.

7.1.1.1. Laplace-Poisson

- Formulación fuerte: Hallar $u(\mathbf{x}) \in C^2(\Omega)$ tal que verifique:

$$\begin{cases} \nabla \cdot \phi(u(\mathbf{x})) = f(\mathbf{x}), & \forall \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}), & \forall \mathbf{x} \in \Gamma_g \\ \phi(u(\mathbf{x})) \cdot \mathbf{n} = -h(\mathbf{x}), & \forall \mathbf{x} \in \Gamma_h \end{cases} \quad (7.1)$$

- Formulación variacional (débil):

Hallar $u(\mathbf{x}) \in \delta = \{u(\mathbf{x}) \in H^1(\Omega) : u(\mathbf{x}) = g(\mathbf{x}), \forall \mathbf{x} \in \Gamma_g\}$ tal que:

$$\int_{\Gamma_h} w(\mathbf{x}) (\phi(u(\mathbf{x})) \cdot \mathbf{n}) d\Gamma_h - \int_{\Omega} \nabla w(\mathbf{x}) \cdot \phi(u(\mathbf{x})) d\Omega = \int_{\Omega} w(\mathbf{x}) f(\mathbf{x}) d\Omega, \quad \forall w(\mathbf{x}) \in V \quad (7.2)$$

donde: $V = \{w(\mathbf{x}) \in H^1(\Omega) : w(\mathbf{x}) = 0, \forall \mathbf{x} \in \Gamma_g\}$

- Aproximación Galerkin-MEF:

Malla: $\{\eta : \text{conjunto de } n \text{ nodos}; \eta_g : \text{conjunto de } n_g \text{ nodos Dirichlet}; \eta_d = \eta - \eta_g\}$

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{j \in \eta_d} d_j N_j(\mathbf{x}) + \sum_{j \in \eta_g} g(\mathbf{x}_j) N_j(\mathbf{x}) = \sum_{j \in \eta} u_j N_j(\mathbf{x})$$

donde $N_i(\mathbf{x})$ son las funciones de base MEF.

- Sistema no lineal de ecuaciones:

$$- \int_{\Omega^h} \nabla N_i(\mathbf{x}) \cdot \phi^h(u(\mathbf{x})) d\Omega^h = \int_{\Omega^h} N_i(\mathbf{x}) f(\mathbf{x}) d\Omega^h + \int_{\Gamma_h^h} N_i(\mathbf{x}) h(\mathbf{x}) d\Gamma_h^h, \quad \forall i \in \eta_d$$

Lenguaje MEF: $\boxed{F_{int}^i(u^h) = F_{ext}^i, \quad \forall i \in \eta_d}$

$$\begin{cases} F_{int}^i(u^h) = - \int_{\Omega^h} \nabla N_i(\mathbf{x}) \cdot \phi^h(u(\mathbf{x})) d\Omega^h \\ F_{ext}^i = \int_{\Omega^h} N_i(\mathbf{x}) f(\mathbf{x}) d\Omega^h + \int_{\Gamma_h^h} N_i(\mathbf{x}) h(\mathbf{x}) d\Gamma_h^h \end{cases}$$

7.1.1.2. Formato matricial (2D)

Si el número total de nodos es n , se tiene:

$$u^h(\mathbf{x}) = \sum_{j=1}^n u_j N_j(\mathbf{x}) \Rightarrow u^h(\mathbf{x}) = N \cdot U = \underbrace{(N_1(\mathbf{x}), \dots, N_n(\mathbf{x}))}_N \cdot \underbrace{\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}}_U = N \cdot U \quad (7.3)$$

$$\nabla u^h(\mathbf{x}) = \sum_{j=1}^n u_j \nabla N_j(\mathbf{x}) \Rightarrow \nabla u^h(\mathbf{x}) = \underbrace{\begin{pmatrix} \frac{\partial N_1}{\partial x} & \dots & \frac{\partial N_n}{\partial x} \\ \frac{\partial N_1}{\partial y} & \dots & \frac{\partial N_n}{\partial y} \end{pmatrix}}_B \cdot \underbrace{\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}}_U = B \cdot U \quad (7.4)$$

Reordenando: $U^T = (U_a^T \ U_p^T)$, $N = (N_a \ N_p)$, $B = (B_a \ B_p)$, donde U_a es el vector de incógnitas nodales y U_p contiene los valores Dirichlet conocidos.

- Sistema no lineal de ecuaciones:

$$F_{int}(U) = F_{ext} \quad (7.5)$$

siendo, ver [10]:

$$\begin{cases} F_{int}(U) = - \int_{\Omega^h} B_a^T \phi^h(U) d\Omega^h, \text{ donde } \phi^h = -k(u) \nabla u^h(\mathbf{x}) = -k(u) B U \\ F_{ext} = \int_{\Omega^h} N_a^T f(\mathbf{x}) d\Omega^h + \int_{\Gamma_h^h} N_a^T h(\mathbf{x}) d\Gamma_h^h \end{cases} \quad (7.6)$$

- Problema lineal: Si $\phi^h = -k(\mathbf{x}) \nabla u^h(\mathbf{x}) = -k(\mathbf{x}) B U$:

$$F_{int}(U) = \int_{\Omega^h} B_a^T k B U d\Omega^h \quad (7.7)$$

En este caso, el sistema de ecuaciones (7.5) es lineal:

$$K U_a = F \quad (7.8)$$

siendo:

$$\begin{cases} K = \int_{\Omega^h} B_a^T k B_a d\Omega^h \\ F = \int_{\Omega^h} N_a^T f(\mathbf{x}) d\Omega^h + \int_{\Gamma_h^h} N_a^T h(\mathbf{x}) d\Gamma_h^h - \left(\int_{\Omega^h} B_a^T k B_p d\Omega^h \right) U_p \end{cases} \quad (7.9)$$

- Cálculo elemental (ne : nodos elemento):

$$\left\{ \begin{array}{l} K^e = \int_{\Omega^e} B^{eT} k B^e d\Omega^e = \int_{\Omega^e} \begin{pmatrix} \nabla N_1^{eT} \\ \vdots \\ \nabla N_{ne}^{eT} \end{pmatrix} k(\mathbf{x}) (\nabla N_1^{eT}, \dots, \nabla N_{ne}^{eT}) d\Omega^e \\ F^e = \int_{\Omega^e} N^{eT} f(\mathbf{x}) d\Omega^e = \int_{\Omega^e} \begin{pmatrix} N_1^e \\ \vdots \\ N_{ne}^e \end{pmatrix} f(\mathbf{x}) d\Omega^e \\ F_{int}^e = \int_{\Omega^e} B^{eT} \phi^h d\Omega^e = \int_{\Omega^e} \begin{pmatrix} \nabla N_1^{eT} \\ \vdots \\ \nabla N_{ne}^{eT} \end{pmatrix} \begin{pmatrix} \phi_x^h \\ \phi_y^h \end{pmatrix} d\Omega^e \end{array} \right. \quad (7.10)$$

El ensamblaje de las matrices elementales K^e y F^e proporciona la parte no encuadrada de la expresión (7.9) en dimensión completa. La aportación Neumann y Dirichlet, parte enmarcada de (7.9), y la eliminación de filas y columnas pasivas (Dirichlet) finalizan la construcción del sistema de ecuaciones (7.8). Para el caso no lineal se muestra el cálculo elemental de las fuerzas internas.

7.1.1.3. Elasticidad lineal

De manera análoga a (7.10), el cálculo a nivel elemental en elasticidad viene dado por:

$$\left\{ \begin{array}{l} K^e = \int_{\Omega^e} B^{eT} D B^e d\Omega^e \\ F^e = \int_{\Omega^e} N^{eT} f(\mathbf{x}) d\Omega^e \\ F_{int}^e = \int_{\Omega^e} B^{eT} \sigma^h d\Omega^e \end{array} \right. \quad (7.11)$$

donde N , B , σ y D quedan posteriormente definidos en (7.16), (7.17) y (7.27) respectivamente.

7.1.1.4. Postproceso. Alisado de flujos y tensiones

Sea $c^*(\mathbf{x})$ una de las funciones componente del flujo o del tensor de tensiones, cuyos valores conocemos en los puntos de integración. Si aproximamos $c^*(\mathbf{x})$ mediante las funciones de base MEF (alisado):

$$c^*(\mathbf{x}) \approx c(\mathbf{x}) = \sum_{j=1}^n c_j N_j(\mathbf{x})$$

Los coeficientes c_j representan los valores nodales de la aproximación $c(\mathbf{x})$. Por la teoría de aproximación lineal mínimo cuadrática de funciones, estos coeficientes c_j verifican el sistema lineal de ecuaciones:

$$\sum_{j=1}^n (N_i, N_j) c_j = (N_i, c^*), \quad i = 1, \dots, n \quad (7.12)$$

$$\text{siendo: } \begin{cases} (N_i, N_j) = \int_{\Omega} N_i(\mathbf{x}) N_j(\mathbf{x}) d\Omega = m_{ij} \\ (N_i, c^*) = \int_{\Omega} N_i(\mathbf{x}) c^*(\mathbf{x}) d\Omega = C_i^* \end{cases} \quad (7.13)$$

Y en formato matricial: $Mc = C^*$.

7.1.2. Objetos y resolución

En la Figura 7.2 se muestra la organización del proceso de cálculo orientado a objeto, para cualquiera de los dos casos (Laplace y Elasticidad) lineales.

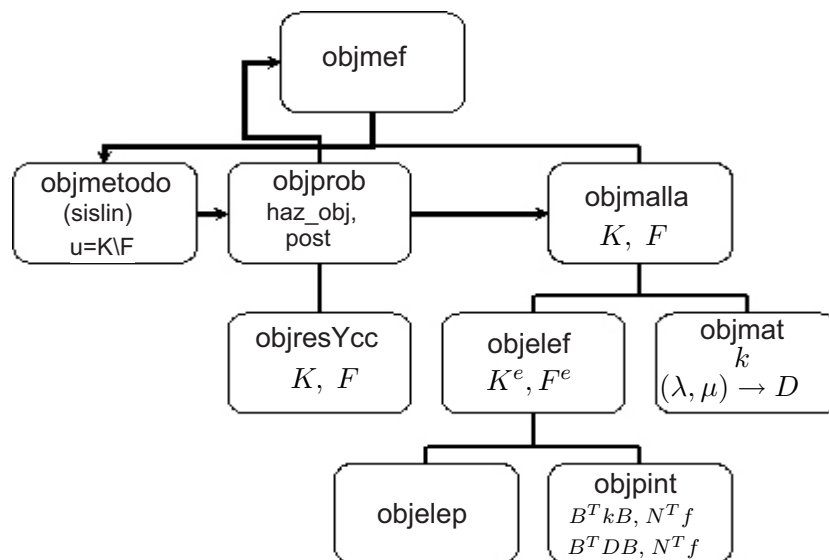


Figura 7.2: Organización del proceso de cálculo, problema lineal.

7.1.2.1. Datos del problema y resolución

En el siguiente guión *script* se definen los datos del problema, se generan los objetos, se ordena su resolución y se postprocesan los resultados:

```

%Forma estándar para introducir la malla, las CC, los puntos de integración
%(nodos=, conec=, ...), (ccd, valccd, ...), (nint1=, nint2=)

%Especificación de los objetos a utilizar y del postproceso:
tipomet='sislin'; tipoprob='laplace'; tipoelef='elef'; tipoelep='cualin';
tipopint='plap'; tipomat='potlin'; flujos={'xx','yy','modulo'};
save datos, dt=load(datos) %dt es una estructura con todos los datos introducidos

%Construcción de los objetos y resolución:
objprob=feval(tipoprob);
[objmef,objprob,objmetodo,obju,objresYcc,objmalla,objelef,...
                                objpint,objmat]=haz_obj(objprob,dt)
resolver(objmef); post(objmef)

```

Se distinguen tres bloques:

1. definición de malla,
2. especificaciones de los objetos a utilizar,
3. generación de objetos y resolución.

Ejecutados los dos primeros bloques se almacenan todos los datos en disco y también en una estructura *dt*. De esta forma se podrá en cualquier punto del programa acceder a cualquiera de los datos originales guardados en disco, o en la estructura *dt* en memoria, si existe acceso a ella.

En el tercer bloque, se crea la función *haz_obj(objprob,dt)* de acuerdo con el tipo de problema especificado, que construye el *objmef* de la Figura 7.2 a partir de los datos de entrada y los derivados, que se encuentran en la estructura *dt*.

A continuación se describen de forma más detallada los objetos y el itinerario desde el fichero de datos hasta el postproceso numérico y gráfico.

7.1.2.2. Objeto método elementos finitos

Es el objeto principal (*objmef*) donde se agregan los objetos método, problema y malla. En ellos reside toda la información necesaria (tanto datos como funciones) para realizar los cálculos. Cuando el problema es lineal, el objeto método consistirá en un objeto sistema lineal (*sislin*) que construirá el sistema de ecuaciones mediante las funciones del objeto problema especificado y del objeto malla, para resolverlo posteriormente.

7.1.2.3. Objeto sistema lineal de ecuaciones

Como veremos en secciones posteriores existen distintas clases del tipo método que implementan procedimientos de resolución estáticos y dinámicos, lineales y no lineales.

Para el caso lineal que estamos considerando, se ha creado una única clase denominada *sislin*, cuyo fin es generar y resolver el sistema lineal de ecuaciones (7.8).

Resuelto el sistema de ecuaciones, se completa la solución (*solc*) con los valores Dirichlet y se guarda la solución en el *objU* de *objmalla* para postproceso. Observamos que el objeto *sislin* recurre al objeto problema para que la construcción del sistema de ecuaciones y el tratamiento de las condiciones de contorno se ajuste al tipo de problema especificado.

7.1.2.4. Objeto problema

El objeto problema gestiona las características específicas del sistema de EDPs y el tratamiento de las Condiciones de Contorno. Al inicio construye los objetos adecuados para la resolución de las EDPs mediante la función *haz_obj*, correspondiente al problema especificado. También, es el que organiza el cálculo del sistema de ecuaciones completo y el que considera las Condiciones de Contorno para obtener K y F de (7.9). Los datos y procedimientos (funciones) comunes residen en la clase *base_prob* de la que heredan las clases *laplace* y *elas* completando sus datos y procedimientos específicos.

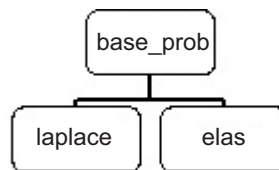


Figura 7.3: Arquitectura de objetos problema lineales por herencia.

Obtenida la solución u del sistema de ecuaciones, *objprob* se encarga del postproceso, calculando los flujos en el problema de Laplace, las tensiones en el problema elástico y realizando su representación gráfica. Cada objeto problema contiene un objeto de restricciones y condiciones de contorno (agregación), Figura 7.2.

7.1.2.5. Objeto Restricciones y Condiciones de Contorno

Este objeto añade al término independiente del sistema de ecuaciones (7.8) las aportaciones Neumann y Dirichlet encuadradas en (7.9) y elimina las filas y columnas Dirichlet, quedando el sistema de ecuaciones dimensionado al número de grados de libertad activos.

7.1.2.6. Objeto malla

Este objeto calcula la matriz de rigidez K y el vector de fuerzas externas F de la expresión (7.9). La función *intnumK* se encarga del bucle en elementos calculando los valores elementales K^e y F^e (7.10), que se ensamblan en las globales K y F .

La función *intnumF* obtiene el vector de fuerzas elementales internas y lo ensambla en F_{int} (7.7). Este último será útil para introducir efectos térmicos y dinámicos en el problema lineal y sobre todo en el caso no lineal, donde es necesario para calcular los residuos del método de Newton.

La función *intnumNv* se utiliza en postproceso (descrito en la sección siguiente), calculando y ensamblando los valores elementales según (7.13).

Existe una única clase del tipo malla que se denomina *mallamef*. Cada objeto malla contiene dos objetos adicionales: un objeto elemento finito (*objelef*) y un objeto material (*objmat*), ver Figura 7.2.

7.1.2.7. Objeto elemento finito

En este objeto se realizan los cálculos a nivel elemental. Las funciones *intnumK*, *intnumF* e *intnumNv*, se encargan de realizar el bucle en puntos de integración, muestrear los integrandos, pesarlos y adicionarlos de acuerdo con las fórmulas de integración numérica de Gauss, respectivamente para el cálculo elemental (7.10) de matrices de rigidez K y vector de fuerzas externas F , vectores de fuerzas internas F_{int} (7.7) y de la expresión (7.13) en el alisamiento de tensiones (flujos) previo al postproceso.

De la clase *elef* derivan por herencia las clases *elef_axi* y *elef_bb*. Estas clases heredan los datos y métodos de la clase *elef* y añaden los específicos para el cálculo elemental axisimétrico (*elef_axi*) y b-barra (*elef_bb*) para el tratamiento de la incompresibilidad de los problemas elásticos. La clase *elef_flu* implementa el cálculo elemental del sistema de EDPs de Navier-Stokes, pero no se describe en este trabajo.

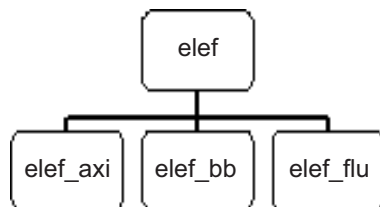


Figura 7.4: Arquitectura de objetos elemento finito por herencia.

Cada objeto de este grupo contiene dos objetos adicionales: un objeto elemento patrón (*objelep*) y un objeto punto de integración (*objpint*), ver Figura 7.2.

7.1.2.8. Objeto elemento patrón

Este objeto proporciona las posiciones, los pesos y los valores de las funciones de forma (matriz N de (7.3)) y sus derivadas (matriz B de (7.4)) en los puntos de integración

del elemento patrón. Por ejemplo para el cuadrado bilineal:

$$\begin{cases} \hat{N}_1(\xi, \eta) = \frac{(1-\xi)(1-\eta)}{4}, & \hat{N}_2(\xi, \eta) = \frac{(1+\xi)(1-\eta)}{4} \\ \hat{N}_3(\xi, \eta) = \frac{(1+\xi)(1+\eta)}{4}, & \hat{N}_4(\xi, \eta) = \frac{(1-\xi)(1+\eta)}{4} \end{cases}$$

La función *jac* proporciona la matriz jacobiana de la transformación del dominio patrón al dominio real del elemento y el valor de su determinante:

$$\begin{cases} x = \sum_{j=1}^4 x_j^e \hat{N}_j(\xi, \eta) \\ y = \sum_{j=1}^4 y_j^e \hat{N}_j(\xi, \eta) \end{cases} \Rightarrow \begin{cases} J(\xi, \eta) = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} \\ dxdy = |J(\xi, \eta)| d\xi d\eta \end{cases}$$

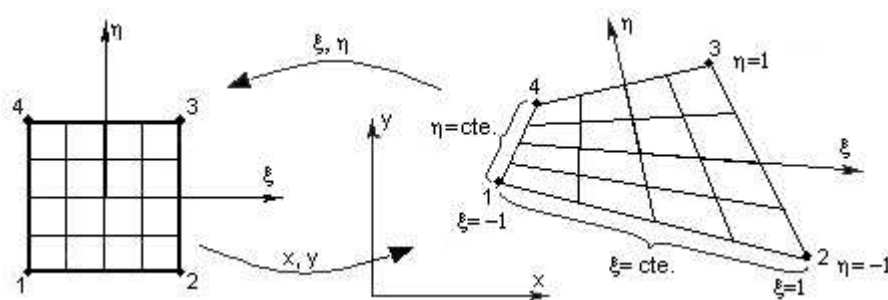


Figura 7.5: Transformación del dominio patrón al dominio real del elemento.

La función *ndreal* proporciona a *objefef* los valores en los puntos de integración del cuadrilátero real de las derivadas de las funciones de forma $N_i^e(\mathbf{x})$, de acuerdo con la relación entre los gradientes de las funciones de forma reales y patrón:

$$\nabla_{(\mathbf{x})} N_i^e(\mathbf{x}) = J^{-1}(\xi, \eta) \nabla_{(\xi, \eta)} \hat{N}_i(\xi, \eta), \quad i = 1, \dots, 4$$

Se han implementado 6 tipos de elementos: segmento, triángulo y cuadrilátero con interpolación lineal y cuadrática. Los datos y procedimientos comunes residen en la clase *base_elep* de la que heredan las 6 clases mencionadas, completando sus datos y procedimientos específicos, ver Figura 7.6.

7.1.2.9. Objeto punto de integración

Este objeto proporciona los valores de los integrandos en los puntos de integración. Es uno de los más utilizados durante el cálculo, por lo que es conveniente optimizar su diseño y programación para conseguir respuestas eficientes.

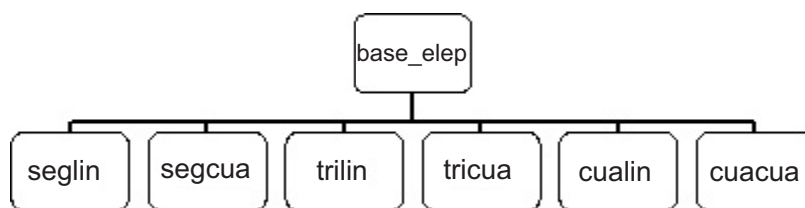


Figura 7.6: Arquitectura de objetos elemento patrón por herencia.

En el caso lineal se han implementado tres tipos de puntos de integración: uno para Laplace (*plap*), otro para elasticidad (*pelas*) y otro para el cálculo de la matriz de masa (*pmasa*). Los datos y procedimientos comunes residen en la clase *base_pint* de la que heredan las tres clases citadas, donde se completan los datos y los métodos específicos.

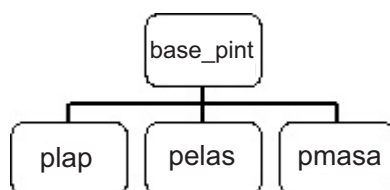


Figura 7.7: Arquitectura de objetos punto de integración por herencia.

En el caso del problema de Laplace (*plap*), como la función constitutiva del material proporciona la constante física k , es inmediato calcular la matriz integrando $B^T k B$ y el vector $N^T f$ de (7.9). En el caso no lineal calcula la matriz integrando de F_{int} (7.6).

De manera similar en elasticidad (*pelas*), se obtienen las deformaciones virtuales correspondientes a los desplazamientos asociados a N en el punto de integración (matriz B de la expresión (7.17)). Como la función constitutiva del material proporciona la matriz producto $D \cdot B$ será inmediato evaluar la matriz integrando $B^T D B$ y el vector $N^T f$ de (7.11).

La clase *pmasa* define el integrando del término independiente del sistema de ecuaciones (7.12), expresión (7.13).

7.1.2.10. Objeto material

En cada punto de integración este objeto proporciona el flujo a partir del gradiente en Laplace y la tensión a partir de la deformación en Elasticidad. Se han implementado 2 clases lineales. Los datos y procedimientos comunes residen en la clase *base_mat* de la que heredan el objeto elástico lineal basado en las ecuaciones de Lamé y el objeto potencial lineal para el problema de Laplace.

Ambos materiales son la base de la que heredan los materiales no lineales que se describen en la Sección 7.2.

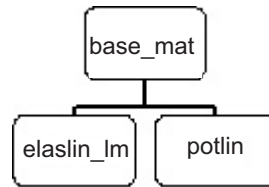


Figura 7.8: Arquitectura de objetos material por herencia.

7.1.3. Postproceso

Calculadas las incógnitas nodales U , el postproceso dirigido por la función del objeto problema *flujos* (o *tensiones*) se encargará de calcular los valores nodales de flujos (o de tensiones) y de representarlos gráficamente según (7.13).

7.1.4. Conclusiones

En esta sección se ha descrito la propuesta del objeto malla, cuyos datos y funciones permiten realizar numéricamente integrales sobre dominios de forma cualquiera. La particularización del integrando debe hacerse en un nuevo objeto (punto de integración), habiéndose presentado dos en particular: los correspondientes al problema de Laplace-Poisson y al problema elástico. Queda abierta la posibilidad de introducir a conveniencia nuevos integrandos con facilidad y sin necesidad de modificar el proceso general de cálculo: bucle en elementos y bucle en puntos de integración.

Para el cálculo de las integrales (7.10) o (7.11) el objeto malla agrega un objeto elemento finito que admite actualmente tres clases de formulación (objetos): la normal (*elef*), la axisimétrica y la b-barra. Igualmente el objeto elemento finito agrega un objeto elemento patrón, que incorpora 6 clases (objetos distintos) o tipos diferentes de interpolación 1D o 2D. Contabilizando a grosso modo los posibles experimentos distintos que se pueden realizar con esta arquitectura salen: 2 (Laplace y Elasticidad) \times 2 (*elef* y *axi*) \times 6 (interpolación) = 24 experimentos distintos lineales y una plataforma y metodología que facilita nuevas adiciones y desarrollos. La Figura 7.9 muestra dos posibles casos de los 24 posibles: el de la izquierda corresponde a un problema de Laplace 1D lineal con interpolación elemental lineal y el de la derecha a un problema elástico lineal axisimétrico con interpolación bilineal en cuadriláteros.

El objeto *sislin* implementa en la actualidad un único método de resolución por eliminación gaussiana (en Matlab) pero la implementación de procedimientos iterativos alternativos podría hacerse enriqueciendo este objeto o generando nuevos objetos y sin alterar los desarrollos restantes. Dos nuevos objetos *sislin2* y *sislin3* por ejemplo, deberían heredar los mismos datos básicos y añadir quizá algunos datos adicionales y la función *resolver* que implementa el método específico. Las opciones resultantes se multiplicarían por 3, pasando a tener: $24 \times 3 = 72$ opciones.

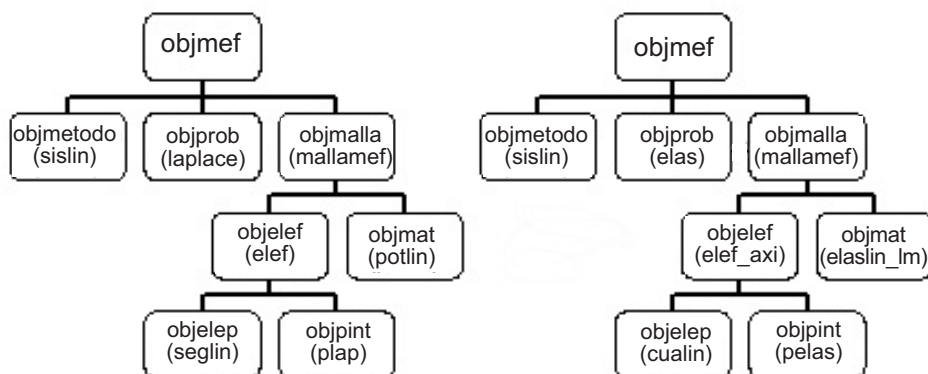


Figura 7.9: Dos posibles casos de la arquitectura elemento finito (caso lineal).

7.2. Problema no lineal

Añadiendo complejidad a la resolución del sistema de ecuaciones con un método iterativo Newton Raphson, podremos resolver problemas cuasi-estáticos no lineales. Para ello completaremos el objeto material con ecuaciones constitutivas no lineales diversas, y en paralelo, desarrollaremos objetos capaces de representar las medidas de deformación requeridas por las ecuaciones constitutivas y que admitan grandes deformaciones.

7.2.1. Resumen de la formulación

Exponemos un resumen de la formulación matemática de la aproximación MEF del sistema de EDPs que gobierna el problema de la elasticidad no lineal, hasta plantear el sistema de ecuaciones cuya resolución proporciona los valores nodales de la aproximación del campo de los desplazamientos. Por analogía extenderemos este planteamiento, completando el caso más sencillo del problema no lineal de Laplace ya introducido en la sección anterior (Sección 7.1).

7.2.1.1. Elasticidad

- Formulación fuerte: Hallar $\mathbf{u}(\mathbf{x}) \in C^2(\Omega)$ tal que verifique:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x})) + \mathbf{f} = 0, & \forall \mathbf{x} \in \Omega \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), & \forall \mathbf{x} \in \Gamma_g \\ \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x})) \cdot \mathbf{n} = \mathbf{t}(\mathbf{x}), & \forall \mathbf{x} \in \Gamma_t \end{cases} \quad (7.14)$$

- Formulación variacional (débil):

Hallar $\mathbf{u}(\mathbf{x}) \in \boldsymbol{\delta} = \{\mathbf{u}(\mathbf{x}) \in H^1(\Omega) : \mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), \forall \mathbf{x} \in \Gamma_g\}$ tal que:

$$\int_{\Omega} \delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x})) d\Omega = \int_{\Gamma_t} \delta \mathbf{u} \cdot \mathbf{t} d\Gamma_2 + \int_{\Omega} \delta \mathbf{u} \cdot \mathbf{f} d\Omega \quad \delta \boldsymbol{\varepsilon}, \quad \forall \delta \mathbf{u} \in \mathbf{V}$$

donde $\delta \boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \delta \mathbf{u} + (\nabla \delta \mathbf{u})^T)$ y $V = \{\mathbf{w}(\mathbf{x}) \in H^1(\Omega) : w(\mathbf{x}) = 0, \forall (\mathbf{x}) \in \Gamma_g\}$.

- Aproximación Galerkin MEF: Malla 2D: n nodos, $2n$ grados de libertad.

- Conjunto $\eta_g^{1,2}$ de nodos Dirichlet para la 1ª o 2ª componente de \mathbf{u} .
- Conjunto $\eta_d^{1,2}$ de nodos no Dirichlet ($\eta_d^{1,2} = \eta - \eta_g^{1,2}$) para la 1ª o 2ª componente.
- Conjunto $\eta_d = \eta_d^1 + \eta_d^2$ de grados de libertad no Dirichlet.

$$\begin{aligned} \mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) &= \begin{pmatrix} u(\mathbf{x}) \\ v(\mathbf{x}) \end{pmatrix} = \\ &= \sum_{j \in \eta_d^1} u_j \begin{pmatrix} N_j(\mathbf{x}) \\ 0 \end{pmatrix} + \sum_{j \in \eta_d^2} v_j \begin{pmatrix} 0 \\ N_j(\mathbf{x}) \end{pmatrix} + \sum_{j \in \eta_g^1} u_j \begin{pmatrix} N_j(\mathbf{x}) \\ 0 \end{pmatrix} + \sum_{j \in \eta_g^2} v_j \begin{pmatrix} 0 \\ N_j(\mathbf{x}) \end{pmatrix} \end{aligned}$$

7.2.1.2. Formato matricial 2D

Definimos $\boldsymbol{\varepsilon}$ y $\boldsymbol{\sigma}$ de esta manera:

$$\begin{cases} \boldsymbol{\varepsilon} = (\varepsilon_x, \varepsilon_y, 2\varepsilon_{xy})^T \\ \boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_{xy})^T \end{cases} \quad (7.14)$$

Utilizando (7.14) la operación $\boldsymbol{\varepsilon} : \boldsymbol{\sigma}$ queda:

$$\boldsymbol{\varepsilon} : \boldsymbol{\sigma} = \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \quad (7.15)$$

$$\mathbf{u}^h(\mathbf{x}) = \underbrace{\begin{pmatrix} N_1 & \cdots & N_n & 0 & \cdots & 0 \\ 0 & \cdots & 0 & N_1 & \cdots & N_n \end{pmatrix}}_N \cdot \underbrace{\begin{pmatrix} u_1 \\ \vdots \\ u_n \\ v_1 \\ \vdots \\ v_n \end{pmatrix}}_U = N \cdot U \quad (7.16)$$

$$\boldsymbol{\varepsilon}^h = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ 2\varepsilon_{xy} \end{pmatrix}^h = \underbrace{\begin{pmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix}}_L \cdot \underbrace{\begin{pmatrix} u^h(\mathbf{x}) \\ v^h(\mathbf{x}) \end{pmatrix}}_{N \cdot U} = L \cdot N \cdot U = B \cdot U \quad (7.17)$$

siendo: $B = \begin{pmatrix} N_{1x} & \cdots & N_{nx} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & N_{1y} & \cdots & N_{ny} \\ N_{1y} & \cdots & N_{ny} & N_{1x} & \cdots & N_{nx} \end{pmatrix}$, siendo $N_{ix_j} = \frac{\partial N_i}{\partial x_j}$ con $x_j = x, y$.

Reordenando: $U^T = (U_a^T \ U_p^T)$, $N = (N_a \ N_p)$, $B = (B_a \ B_p)$, donde U_a es el vector de incógnitas nodales y U_p contiene los valores Dirichlet conocidos.

- Sistema no lineal de ecuaciones: $F_{int}(U) = F_{ext}$

$$\begin{cases} F_{int}(U) = \int_{\Omega^h} B_a^T \sigma d\Omega^h \\ F_{ext} = \int_{\Omega^h} N_a^T \mathbf{f} d\Omega^h + \int_{\Gamma_t^h} N_a^T \mathbf{t} d\Gamma_2^h \end{cases} \quad (7.18)$$

- Resolución mediante Newton Raphson:

$$\begin{cases} R(U_k) = F_{ext} - F_{int}(U_k) \\ U_{k+1} = U_k - J^{-1}R(U_k) \end{cases} \quad k = 0, 1, 2, \dots \text{ hasta que } \|R(U_{k+1})\| < tol \approx 0 \quad (7.19)$$

- Matriz de rigidez tangente:

$$K^t(U_k) = -J(U_k) = -\frac{\partial R(U_k)}{\partial U_k} = \frac{\partial F_{int}^k(U_k)}{\partial U_k} \quad (7.20)$$

- Cálculo del vector de fuerzas internas: Cálculo elemental (ne : nodos elemento), ensamblaje y eliminación de filas pasivas.

$$F_{int}^e = \int_{\Omega^e} B^T \sigma d\Omega^e = \int_{\Omega^e} \begin{pmatrix} \frac{\partial N_1^e}{\partial x} & 0 & \frac{\partial N_1^e}{\partial y} \\ \vdots & \vdots & \vdots \\ 0 & \frac{\partial N_{ne}^e}{\partial y} & \frac{\partial N_{ne}^e}{\partial x} \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix} d\Omega^e \quad (7.21)$$

- Cálculo de la matriz de rigidez tangente (material K_m^t y geométrica K_g^t): Cálculo elemental, ensamblaje y eliminación de filas y columnas pasivas (Dirichlet).

$$K^t(U) = \frac{\partial F_{int}(U)}{\partial U_a} = \frac{\partial}{\partial U_a} \int_{\Omega} B_a^T \sigma d\Omega = \int_{\Omega} \left(B_a^T \frac{\partial \sigma}{\partial U_a} + \frac{\partial B_a^T}{\partial U_a} \sigma \right) d\Omega = K_m^t + K_g^t \quad (7.22)$$

La matriz de rigidez tangente material viene dada por:

$$K_m^t = \int_{\Omega^e} B^T \cdot \frac{\partial \sigma}{\partial U} d\Omega^e \quad (7.23)$$

y teniendo en cuenta:

$$\frac{\partial \sigma}{\partial U} = \frac{\partial \sigma}{\partial \varepsilon} \cdot \frac{\partial \varepsilon}{\partial U} = \frac{\partial \sigma}{\partial \varepsilon} \cdot B = M \cdot B \quad (7.24)$$

siendo M el módulo material (en cada punto de integración):

$$M(\mathbf{x}) = \frac{\partial \sigma}{\partial \varepsilon} = \begin{pmatrix} \frac{\partial \sigma_x}{\partial \varepsilon_x} & \frac{\partial \sigma_x}{\partial \varepsilon_y} & \frac{\partial \sigma_x}{\partial \varepsilon_{xy}} \\ \frac{\partial \sigma_y}{\partial \varepsilon_x} & \frac{\partial \sigma_y}{\partial \varepsilon_y} & \frac{\partial \sigma_y}{\partial \varepsilon_{xy}} \\ \frac{\partial \sigma_{xy}}{\partial \varepsilon_x} & \frac{\partial \sigma_{xy}}{\partial \varepsilon_y} & \frac{\partial \sigma_{xy}}{\partial \varepsilon_{xy}} \end{pmatrix} \quad (7.25)$$

Con lo que la expresión (7.23) queda:

$$K_m^t = \int_{\Omega^e} B^T M B d\Omega^e \quad (7.26)$$

- Problema lineal: Si $\sigma = D\varepsilon = DBU$, siendo D (una matriz 3×3) función de los coeficientes de Lamé (λ, μ):

$$F_{int}(U) = \int_{\Omega^h} B_a^T \sigma d\Omega^h = \left(\int_{\Omega^h} B_a^T D B d\Omega^h \right) U \quad (7.27)$$

En este caso, el sistema de ecuaciones (7.27) es lineal:

$$KU_a = F \quad (7.28)$$

siendo:

$$\begin{cases} F = F_{ext} - \left(\int_{\Omega^h} B_a^T D B_p d\Omega^h \right) U_p \\ K = \left(\int_{\Omega^h} B_a^T D B_a d\Omega^h \right) \end{cases} \quad (7.29)$$

7.2.1.3. Laplace-Poisson

De manera análoga si $\phi^h = -k(u)\nabla u^h(\mathbf{x})$ se tiene:

- Cálculo del vector de fuerzas internas: De (7.3), (7.4) y (7.6), cálculo elemental (7.10), ensamblaje y eliminación de filas pasivas se tiene:

$$F_{int}^e = \int_{\Omega^e} B^{eT} \phi^h d\Omega^e = \int_{\Omega^e} \begin{pmatrix} \frac{\partial N_1^e}{\partial x} & \frac{\partial N_1^e}{\partial y} \\ \vdots & \vdots \\ \frac{\partial N_{ne}^e}{\partial x} & \frac{\partial N_{ne}^e}{\partial y} \end{pmatrix} \begin{pmatrix} \phi_x^h \\ \phi_y^h \end{pmatrix} d\Omega^e \quad (7.30)$$

- Cálculo de la matriz de rigidez tangente: Cálculo elemental, ensamblaje y eliminación de filas y columnas pasivas.

$$K^t(U) = \frac{\partial F_{int}(U)}{\partial U_a} = - \int_{\Omega^h} B_a^T \frac{\partial \phi^h(U)}{\partial U_a} d\Omega^h = K(U) + K^*(U) \quad (7.31)$$

$$\text{siendo: } \begin{cases} K(U) = \int_{\Omega^h} B_a^T k(u) B_a d\Omega^h \\ K^*(U) = - \int_{\Omega^h} B_a^T \phi^{h*} N_a d\Omega^h \\ \phi^{h*} = -k'(u)BU \end{cases}$$

7.2.2. Objetos y resolución

En la Figura 7.10, particularizada para el caso de elasticidad, se puede observar la organización del proceso de cálculo orientado a objeto, que también vale para Laplace con ligeras modificaciones.

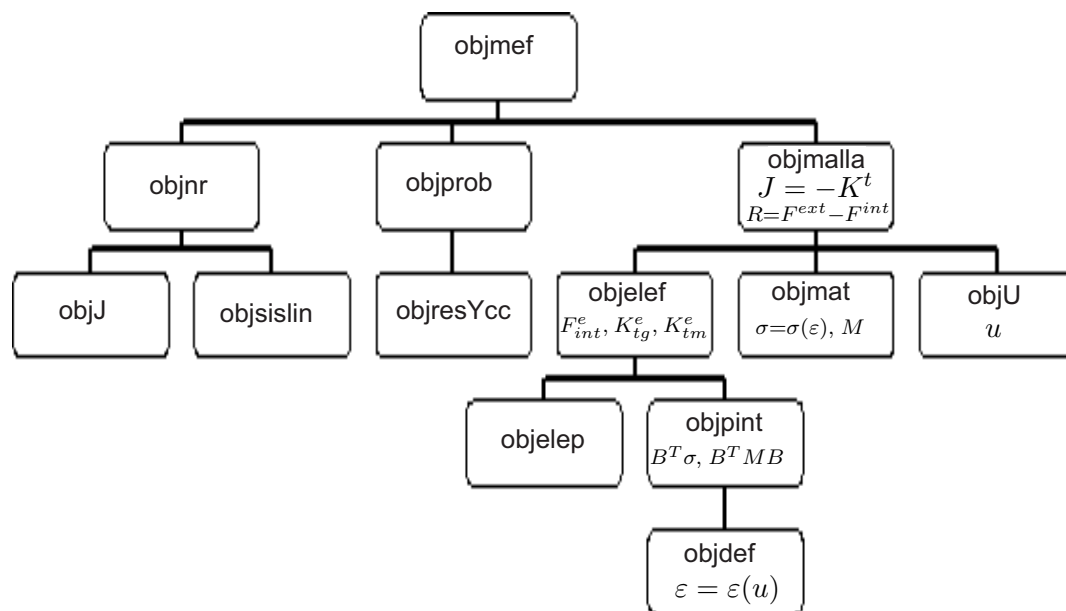


Figura 7.10: Organización del proceso de cálculo, problema no lineal.

La diferencia principal frente a lo explicado en la Sección 7.1 es el *objeto método*, que ahora implementa el método iterativo de Newton Raphson. El *objeto malla* no cambia pero incorpora el cálculo de las fuerzas internas y de la rigidez tangente material y geométrica, mediante el diseño de nuevos objetos puntos de integración con algo más de complejidad. También es válida la descripción del *objeto mef* de la Sección 7.1 y el itinerario que va desde el fichero de datos hasta el postproceso numérico y gráfico del *objeto problema*.

7.2.2.1. Objeto malla

Este objeto calcula la matriz de rigidez K y el vector de fuerzas externas F de la expresión (7.9) en el caso lineal, y en el caso no lineal calcula R y J de las expresiones (7.19) y (7.20). Se encarga del bucle en elementos calculando los valores elementales que se ensamblan en las globales. Para ello, el objeto malla contiene dos objetos adicionales (agregación): un objeto elemento finito (*objelef*), y un objeto material (*objmat*). Ver Figura 7.1.

7.2.2.2. Objeto elemento finito

Este es el mismo objeto que el descrito en el problema lineal, Sección 7.1.2.7, donde se realizan los cálculos a nivel elemental.

7.2.2.3. Objeto elemento patrón

Este es el mismo objeto que el del problema lineal Sección 7.1.2.8. Es el objeto que proporciona las posiciones, los pesos y los valores de las funciones de forma (matriz N de (7.16)) y sus derivadas (matriz B de (7.17)) en los puntos de integración del elemento patrón.

7.2.2.4. Objeto punto de integración

Agregado al objeto elemento finito (*objeief* Figura 7.10) esta clase tiene como fin definir los integrandos de las expresiones (7.18) y (7.26).

Se han implementado 7 tipos de puntos de integración. Los datos y procedimientos comunes residen en la clase *base_pint* de la que heredan los 6 objetos que se indican a continuación donde se completan los datos y los métodos específicos de la clase.

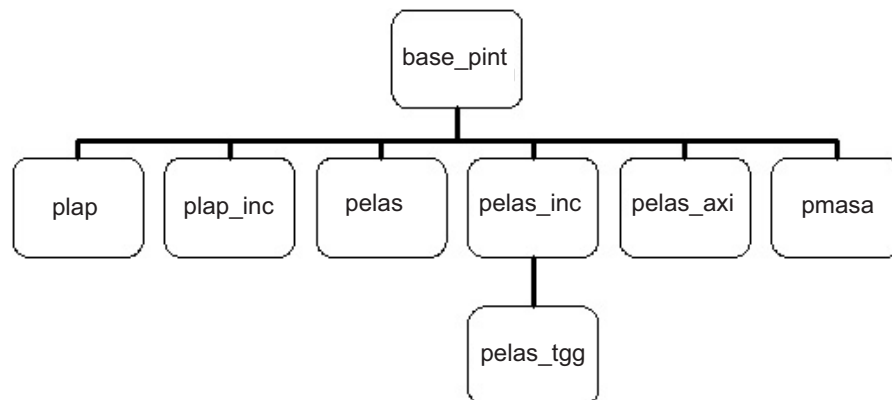


Figura 7.11: Arquitectura de objetos punto de integración por herencia.

El objeto más significativo es *pelas_inc* (*plap_inc* en Laplace) que está pensado para proporcionar los integrandos del proceso no lineal incremental: K_m^t y F_{int} . La clase *pelas_axi* es similar a *pelas_inc* con algún aspecto adicional para calcular los integrandos del caso axisimétrico. La clase *pelas_tgg* hereda de *pelas_inc* y calcula el integrando de la matriz tangente geométrica.

La función *integK* calcula el integrando $B^T M B$ de la matriz tangente material (7.26) y es similar a la del caso lineal pero utilizando el módulo material (7.25) que es calculado por el objeto material en función de las tensiones existentes en cada punto de integración

y en cada iteración. La función $integF$ calcula en el punto de integración el integrando (7.21) de las fuerzas internas, para lo cual como paso previo precisa calcular la deformación y la tensión en el punto de integración a partir de los desplazamientos elementales de prueba.

Pelas_inc incorpora a sus datos la identificación del punto de integración y del elemento tratado. Esto es necesario para acceder a las variables internas de la ecuación constitutiva no lineal en cada punto de integración real. Asimismo incorpora el objeto deformación especificado en los datos (ver Figura 7.10), lo cual permite definir la medida de deformación de acuerdo con lo precisado por la ecuación constitutiva del material.

7.2.2.5. Objeto deformación

Este objeto proporciona en cada punto de integración una medida de deformación asociada al campo de desplazamientos u . Los datos y procedimientos comunes residen en la clase *base_def* de la que heredan las 6 clases que se indican a continuación. De nuevo, cada uno de los objetos añade los datos y los métodos específicos de la clase.

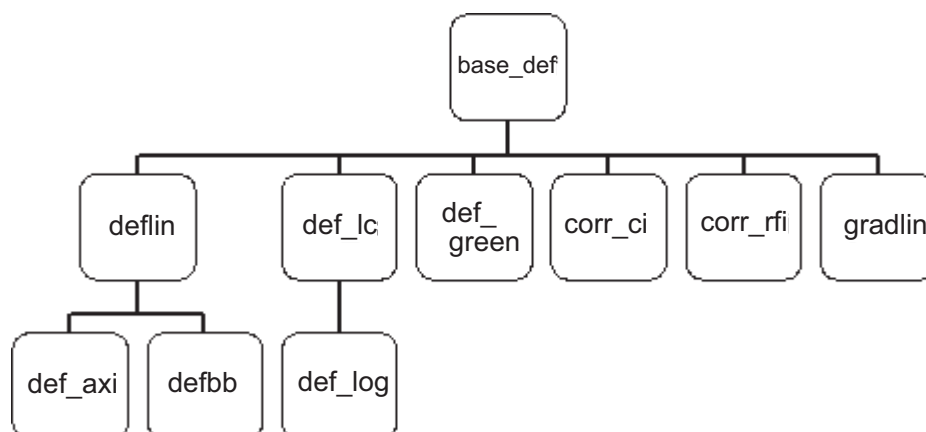


Figura 7.12: Arquitectura de objetos deformación por herencia.

La clase *deflin* implementa la relación lineal entre la deformación y el desplazamiento. De ella heredan *def_axi* y *defbb* que sirven para las formulaciones axisimétricas y b-barra respectivamente. Las tres son las que se utilizan para calcular las fuerzas internas en los problemas de ecuación constitutiva no lineal (elastoplasticidad por ejemplo) pero con pequeñas deformaciones y pequeños giros y desplazamientos. La clase *gradlin* es la análoga de *deflin* para el problema de Laplace. Las clases restantes implementan deformaciones no lineales para problemas de no linealidad geométrica: problemas con grandes giros y desplazamientos, y pequeñas o grandes deformaciones.

Las clases *def_lc*, *def_log* y *def_green* son las que implementan las medidas de deformación left-Cauchy, logarítmica y Green respectivamente para las ecuaciones constitutivas

hiperelásticas. Todas ellas agregan como dato un objeto *deflin* para el cálculo de la matriz B adaptada en su caso al problema axisimétrico en tiempo de ejecución.

Las clases *corr_ci* y *corr_rfi* implementan deformaciones corrotacionales para leyes constitutivas hipoelásticas.

7.2.2.6. Objeto material

Este objeto implementa la ley constitutiva $\sigma(\epsilon)$ del material que proporciona la tensión a partir de la deformación (o el flujo a partir del gradiente). Se han implementado varios tipos de material no lineal que heredan de la clase lineal y son padres de las clases tangentes que proporcionan el módulo del material (7.25) para el cálculo de las matrices de rigidez tangentes materiales (7.26). La clase *plasxy* es para material elastoplástico con pequeñas deformaciones (deformación lineal), *hipo* para grandes deformaciones en formato incremental (deformación corrotacional) y las tres restantes para grandes deformaciones en formato hiperelástico: *neoh* implementa un material neohookeano (deformación *def_lc*), *neoh_nc* para materiales neohookeanos cuasi-incompresibles y *hencky* que trabaja en direcciones principales y con la deformación logarítmica (*def_log*), ver [63].

Todos ellos son ampliaciones más elaboradas del material lineal, a cuyos datos y ecuación constitutiva recurren con frecuencia. Es por esto que todos heredan del material lineal (*potlin* y *elaslin_lm*), aumentando sus datos y funciones con los propios específicos. Por ejemplo, la ecuación constitutiva de la clase *plasxy* implementa el algoritmo de retorno radial, recibe la deformación total en un punto de integración, calcula la deformación plástica e introduce la deformación elástica como input de la función constitutiva de la clase *elaslin_lm* para obtener la tensión, ver [63].

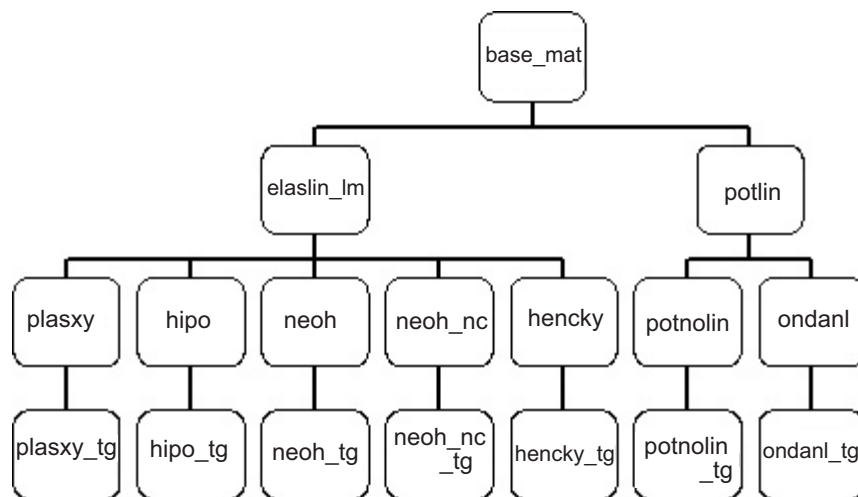


Figura 7.13: Arquitectura de objetos material por herencia.

Las clases material tangente heredan de las no lineales para computar el módulo material M (7.25). Por ejemplo, la clase *plasty_tg* calcula el módulo elastoplástico del material en tres modalidades según lo indicado en los datos de entrada: módulo algorítmico o consistente, módulo continuo, y módulo lineal o elástico.

La segunda rama de la Figura 7.13, *potlin*, implementa la ecuación constitutiva para el problema de potencial (Laplace). Por ejemplo, *potnolin* introduce una ecuación constitutiva no lineal para el problema de Laplace $k = k(u)$, y *ondanl* modifica el tiro constante T de una cuerda vibrante para que dependa del desplazamiento transversal u de la cuerda.

7.2.2.7. Objeto problema

Al objeto problema del caso lineal se le incorporan nuevas características (datos y funciones), necesarias para implementar el proceso iterativo no lineal, que en la Sección 7.3 quedará inserto en los métodos paso a paso implícitos no lineales.

En el caso no lineal, el objeto problema organiza los cálculos del residuo R y la matriz jacobiana J (7.19) que demanda el método de Newton. También debe guardar y gestionar el estado tensional (o de flujo) en cada uno de los puntos de integración de cada uno de los elementos de la discretización. Esto es necesario para gestionar las ecuaciones constitutivas de tipo incremental y para calcular la matriz de rigidez tangente geométrica en grandes deformaciones.

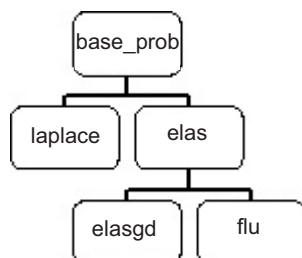


Figura 7.14: Arquitectura de objetos problema lineales y no lineales por herencia.

Además, de la clase *elas* heredan las clases *elasgd* para grandes deformaciones y *flu* para el sistema de EDPs de Navier-Stokes, aunque este último no se describe en estas líneas.

7.2.2.8. Objeto método Newton Raphson

Este objeto implementa el algoritmo iterativo (7.19) donde se resuelve el paso de un problema no lineal, pidiendo los valores de R y J al problema considerado. El itinerario que sigue es: *objmetodo*, *objprob*, *objmalla*. De esta forma es posible adaptar el cálculo de

R y J al método especificado (cuasi-estático o dinámico) y a continuación al problema considerado (elástico, Laplace, general).

Cada objeto de este grupo contiene dos objetos adicionales (agregación), ver Figura 7.10: un objeto *sislin* para resolver los sistemas lineales y otro objeto jacobiano (*objJ*) para el cálculo de la matriz de rigidez tangente. El primero de ellos ha quedado descrito en la sección referente a problemas lineales. Describimos el segundo a continuación.

7.2.2.9. Objeto Jacobiano

El fin del objeto Jacobiano es almacenar la matriz jacobiana y proporcionar los métodos para su cálculo de maneras distintas, siendo tres las implementadas. Por un lado, por derivación analítica (*jder*) según (7.22) y por otro, por diferencias finitas del vector residuo en dos versiones: la versión completa (*jdf*) y la simplificada (*jdfmef*) aprovechando las características dispersas de la matriz J del MEF. Para ello existen tres tipos de objeto Jacobiano J . Los datos y procedimientos comunes residen en la clase *base_J* de la que heredan la clase *jder*, la clase *jdf* y la clase *jdfmef*.

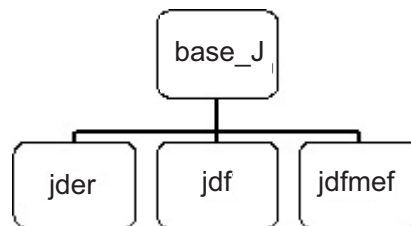


Figura 7.15: Arquitectura de objetos Jacobiano por herencia.

7.2.2.10. Objeto método elemento finito

Es el objeto principal donde se agregan los objetos método, problema y malla, que contienen toda la información necesaria para realizar los cálculos, ver Figura 7.1.

Si el problema es lineal, el objeto método consistirá en un objeto sistema lineal (*sislin*) que primero construirá el sistema de ecuaciones mediante las funciones del objeto problema especificado y del objeto malla, para resolverlo posteriormente.

En el caso de un problema estacionario no lineal, el objeto método coincidirá con el objeto Newton Raphson (*objnr*) o con el objeto *cuasi*. En el segundo caso, el proceso de resolución se realizará paso a paso en la forma indicada en la Sección 7.2.1.2, resolviendo un proceso iterativo de Newton en cada paso. Si el problema es dinámico el objeto método especificado coincidirá con uno de los indicados en la Sección 7.3.

7.2.3. Conclusiones

En esta sección observamos que respetando la arquitectura de la sección anterior e introduciendo un nuevo objeto método que implementa el proceso de resolución iterativo del método de Newton Raphson, es posible resolver problemas no lineales reutilizando los recursos desarrollados en la sección anterior y añadiendo nuevos objetos materiales no lineales y de deformación para grandes deformaciones.

El diseño de los objetos, basado en la herencia cuando conviene reutilizar los datos y funciones del padre, y en la agregación cuando conviene discriminar automáticamente entre distintas opciones dejando la puerta abierta para nuevas incorporaciones futuras, permite proponer una metodología flexible, basada en la reutilización, la sobrecarga (mismo lenguaje) y la encapsulación (compartimentos estancos). Ello permite añadir nuevos desarrollos aprovechando los anteriores pero sin modificarlos: integrales tipo K (para rigideces tangentes con el objeto punto de integración adecuado) y tipo F (para fuerzas internas). Los nuevos desarrollos MEF implican principalmente nuevos objetos punto de integración (EDP) o del tipo deformación y material.

Contabilizando a grosso modo los posibles experimentos distintos que se pueden realizar, salen:

- 1 (Laplace) x 2 (*elef* y *axi*) x 6 (interpolación) x 1 (material no lineal) x 2 (J exacto o diferencias finitas)=24.
- 1 (elasticidad) x 2 (*elef* o *axi*) x 6 (interpolación) x 6 (materiales) x 2 (J por derivación exacta o diferencias finitas) = 144.
- que dan un total de $24+144 = 168$ posibles experimentos distintos y una plataforma y metodología que facilita nuevas adiciones y desarrollos de leyes constitutivas no lineales en pequeñas o grandes deformaciones.

La Figura 7.16 muestra dos posibles casos de los 168 posibles: el de la izquierda corresponde a un problema de Laplace 1D no lineal con cálculo por diferencias finitas de la matriz jacobiana e interpolación elemental lineal. El de la derecha corresponde a un problema elastoplástico 2D con pequeñas deformaciones (deformación lineal), axisimétrico, con cálculo exacto de la matriz jacobiana y con interpolación bilineal en cuadriláteros.

En general, el proceso iterativo de Newton, incluso con “linesearch” puede no ser suficiente para la convergencia de los problemas no lineales. Es preciso entonces complementar el método mediante la continuación o subincrementación, con un nuevo objeto que denominaremos *cuasi* (cuasi-estático) y que se presenta en la siguiente sección conjuntamente con los métodos paso a paso para la resolución de los problemas dinámicos.

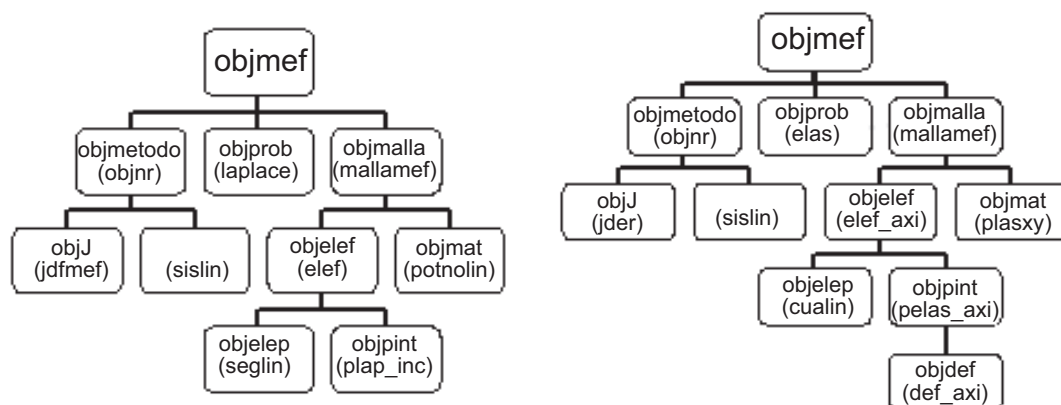


Figura 7.16: Dos posibles casos de la arquitectura elemento finito (caso no lineal).

7.3. El problema dinámico

Finalmente, en esta sección se desarrolla un proceso para la resolución paso a paso de problemas dinámicos y donde el método cuasi-estático es un caso particular. Para ello completamos el objeto método con distintos métodos capaces de resolver ecuaciones diferenciales ordinarias.

Implementaremos métodos capaces de resolver EDOs de orden 1. Empezaremos por el método trapezoidal, que es un método unipaso, desarrollándolo tanto para el caso general de EDOs de orden 1, como para el caso específico lineal de coeficientes constantes. A continuación, extenderemos estos desarrollos a los métodos multipaso (BDF, NDF y BDF- α) y a los métodos multipaso que utilizan puntos super-futuros (familia EBDF y MEBDF). De nuevo, se hará el desarrollo general de cada uno de los métodos y la particularización al problema lineal de coeficientes constantes. Terminaremos esta parte con el método *odesuite*, que permitirá utilizar las funciones que proporciona Matlab para la integración numérica de EDOs de orden 1.

Por último, realizaremos el desarrollo del método alfa (HHT- α), método de resolución de EDOs de orden 2 (sin ser éstas reducidas a orden 1), tanto en su versión genérica como adaptada al problema lineal de coeficientes constantes.

7.3.1. Resumen de la formulación

Generalizamos el desarrollo matricial (7.18), incluyendo la variable temporal t entre las variables independientes (\mathbf{x}, t) o (x, y, z, t) y añadiendo a las fuerzas de volumen las fuerzas de inercia y de amortiguamiento:

$$\mathbf{u}(\mathbf{x}, t) \approx \mathbf{u}^h(\mathbf{x}, t) = N(\mathbf{x})U(t) \quad (7.32)$$

La función $\mathbf{f}(\mathbf{x})$ de la sección anterior se convierte ahora en:

$$\mathbf{f}(\mathbf{x}, t) - \rho \mathbf{u}_{tt}(\mathbf{x}, t) - c \mathbf{u}_t(\mathbf{x}, t) \approx \mathbf{f}(\mathbf{x}, t) - \rho N(\mathbf{x}) U''(t) - c N(\mathbf{x}) U'(t) \quad (7.33)$$

donde ρ y c son la densidad y el amortiguamiento respectivamente.

La integral $\int_{\Omega} N^T \mathbf{f} d\Omega$ de (7.18) queda ahora de esta manera:

$$\int_{\Omega} N^T \mathbf{f} d\Omega - \underbrace{\left(\int_{\Omega} N^T \rho N d\Omega \right)}_M U'' - \underbrace{\left(\int_{\Omega} N^T c N d\Omega \right)}_C U' \quad (7.34)$$

Y el sistema de ecuaciones (7.18), ahora es el sistema de EDOs de orden 2:

$$M U'' + C U' + F_{int}(U) = F_{ext} \quad (7.35)$$

donde M , C , F_{int} y F_{ext} vienen dados por [9]:

$$\begin{cases} M = \int_{\Omega} N^T \rho N d\Omega, & C = \int_{\Omega} N^T c N d\Omega \\ F_{int} = \int_{\Omega} B^T \sigma^h(U) d\Omega, & F_{ext} = \int_{\Gamma_t} N^T \mathbf{t} dS_t + \int_{\Omega} N^T \mathbf{f} d\Omega \end{cases} \quad (7.36)$$

Transformamos el sistema de EDOs de orden 2 (7.35) (en general no lineal), en un sistema de EDOs equivalente de orden 1, $M y' = f(t, y)$:

$$\begin{cases} d(t) = U(t) \\ v(t) = U'(t) \end{cases} \Rightarrow \begin{cases} d'(t) = U'(t) = v(t) \\ M v'(t) = M U''(t) = -F_{int}(d) - C v + F_{ext} \end{cases} \quad (7.37)$$

Teniendo en cuenta la definición del residuo en (7.19), el sistema (7.37) en formato matricial será:

$$\underbrace{\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}}_{M^*} \underbrace{\begin{pmatrix} d'(t) \\ v'(t) \end{pmatrix}}_{y'} = \underbrace{\begin{pmatrix} v(t) \\ R(d) - C v \end{pmatrix}}_{f(t,y)} \Rightarrow M^* y' = f(t, y) \quad (7.38)$$

donde f y $j = \partial f / \partial y$ son:

$$\begin{cases} f_1(t, y) \equiv y_2 = v, & f_2(t, y) \equiv R(y_1) - C y_2 = R(d) - C v \\ j = \frac{\partial f(t, y)}{\partial y} = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial v}{\partial d} & \frac{\partial v}{\partial v} \\ \frac{\partial(R(d)-Cv)}{\partial d} & \frac{\partial(R(d)-Cv)}{\partial v} \end{pmatrix} \\ = \begin{pmatrix} 0 & I \\ \frac{\partial R(d)}{\partial d} & -C \end{pmatrix} = \begin{pmatrix} 0 & I \\ J(U) & -C \end{pmatrix} \end{cases} \quad (7.39)$$

donde $J(U) = -K^t(U)$ quedó definida en (7.20).

El resumen anterior es válido para elasticidad dinámica y también para el problema de onda (Laplace dinámico de orden 2). La formulación del problema de difusión (Laplace dinámico de orden 1) se produce de manera similar. Por ejemplo, si en el problema de difusión térmica añadimos las fuentes la inercia térmica, tendremos:

$$f(\mathbf{x}) \rightarrow f(\mathbf{x}, t) - c_e u_t(\mathbf{x}, t) \approx f(\mathbf{x}, t) - c_e N(\mathbf{x}) U'(t) \quad (7.40)$$

donde c_e es el calor específico del material.

Entonces:

$$\int_{\Omega} N^T f d\Omega \rightarrow \int_{\Omega} N^T f d\Omega - \left(\int_{\Omega} N^T c_e N d\Omega \right) U' \quad (7.41)$$

Y obtenemos el sistema de EDOs de orden 1:

$$MU' + F_{int}(U) = F_{ext} \Rightarrow MU'(t) = F_{ext} - F_{int}(U) = R(U) \quad (7.42)$$

siendo M , F_{int} y F_{ext} de la expresión (7.42):

$$\begin{cases} M = \int_{\Omega} N^T c_e N d\Omega, \\ F_{int} = \int_{\Omega} B^T \phi^h(U) d\Omega, \quad F_{ext} = \int_{\Gamma_h} N^T h d\Gamma_h + \int_{\Omega} N^T f d\Omega \end{cases} \quad (7.43)$$

El sistema de EDOs de orden 1 (7.42) tiene la forma:

$$My' = f(t, y), \quad \text{siendo: } \begin{cases} f = R(y) = R(U) \\ j = \frac{\partial f(t, y)}{\partial y} = \frac{\partial R(U)}{\partial U} = J = -K^t(U) \end{cases} \quad (7.44)$$

7.3.2. Objetos

En ambos casos (7.38) y (7.44), integraremos el sistema de EDOs mediante métodos paso a paso, que si son de carácter implícito precisarán en cada paso la resolución de un sistema no lineal de ecuaciones mediante el método de Newton Raphson. En consecuencia, la organización del proceso de cálculo orientado a objeto reutiliza la organización anterior ampliándola.

El cálculo de las integrales de la formulación (7.36) y (7.43) ya ha sido explicado en los puntos anteriores. El *objeto malla* se reutiliza al 100 % sin extensiones adicionales, por lo que en este punto nos centraremos en la explicación del *objeto método* (ver Figura 7.1) y muy levemente en alguna extensión del *objeto problema* que en un 90 % es una reutilización de lo expuesto en las Secciones 7.1 y 7.2. Al comienzo de este capítulo se explicó la organización del proceso de resolución de un problema, desde la definición y lectura de datos hasta el postproceso y la representación gráfica de los resultados, pasando por la generación y creación de los objetos y el desencadenamiento de la resolución:

$$resolver(objmef) \rightarrow resolver(objmetodo) \rightarrow objprob, objmalla$$

A continuación se muestra un “fichero.m” (guión *script*) con la definición de los datos correspondientes a un problema de CI y CC (vibraciones libres). Se trata de una placa 2D rectangular traccionada inicialmente con unas CI de alargamiento. Se modela un cuarto de la misma con CC deslizantes en ambos ejes de simetría. El material es neohookeano (deformación *def_lc*), y se muestran varias opciones con respecto al método: *ode45*, *ode15s*, trapezoidal, Newmark, HHT- α , etc. siendo esta última la que está activada con cálculo exacto de la matriz jacobiana, aunque se muestra cómo hacerlo por diferencias finitas.

```

%*****Datos geométricos:
L=1; m=1; n=1; coorv=[0 L; 0 L]; ngl=2;
nele=m*n; nnodos=(m+1)*(n+1); [nodos, conec,lados]=cuadreg(coorv,m,n);
%*****Número de puntos de integración numérica (máximo 5):
nint1=2; nint2=2; nint=[nint1 nint2];
%*****Funciones físicas:
mody=1000; modp=.25; f=[]; ro=1; d=0; dilat=[]; temp=[];
%*****CC Dirichlet y Neumann:
lado1=lados(find(lados(:,1)),1); lado4=lados(find(lados(:,4)),4); lado3=
ccd=[lado4' lado1'+nnodos]; valccd=zeros(size(ccd)); ccn=[]; valccn=[];
%*****Condiciones Iniciales e intervalo temporal:
ci='[-x/8; y/2]'; cider='[0; 0]'; tini=0; tfin=0.5; npasos=100
%*****Tipos (objetos):
tipoele='dplana'; tipoele='cualin'; tipoprob='elasgd'; tipoelef='elef';
tipopint='pelas_inc'; tipodef='def_lc'; tipomat='neoh'; tipofext='fijas';
%*****Método (objetos):
tipomet='ode45'; tipomet='ode15s'; tipomet='rk';
%-----
tipomet='alfa'; metNR='nr'; alfa=-0.3; alfa=0; gamma=(1-2*alfa)/2; beta=(1-alfa)^2/4;
%-----
% tipomet='trap2'; metNR='nrg'; npasos=50;
%-----
fty=@onda; metJ='jdf'; metJ='jdfmef'; metJ='jder';
tolNR=1E-6; maxiterNR=10; escpaso='si'; escNR='si'; esc_iterNR='si'; esc_iterNR=[];
%*****Postproceso:
tensiones_dat={'yy'};

```

7.3.2.1. Objeto método ode (1), objeto base_met

En el avance paso a paso de los distintos métodos se distingue una estructura común: inicialización, pasos (y_n , y_{n+1} , *objnr*, actualización) y actualización final. Los datos y funciones comunes se organizan en una clase padre (*base_met*) de la que heredan los métodos específicos.

- Datos: *objfty*, *interv*, *yini*, *npasos*, *h*, *tsol*, *ysol*, *dat*.
- Funciones: *resolver*(objode, objprob, objmalla),
actpaso(t, y, objode, objprob, objmalla),
actfin(t, y, objode, objprob, objmalla).

La función *resolver*, que contiene el bucle en pasos, y las funciones de actualización (en cada paso y al final), son comunes a casi todos los métodos y quedan explicados en los siguientes cuadros:

```
function resolver(objode,objprob,objmalla)
inic(objode,objprob,objmalla); t=objode.interv(1);
y=objode.yini; h=objode.h; m=objode.npasos;
for j1=2:m+1
    escpaso(objode.esc,j1-1,objode.tsol(j1))
    [t,y]=paso(objode,t,y,h,varargin{:});
    objode.ysol(j1,:)=y';
end;
actfin([],y,objode,objprob,objmalla);
```

```
function actpaso(t,y,objode,objprob,objmalla_n)
na=objprob.ngdla; y_n=y(1:na);
yc=solc(objprob,y_n,'t'); uc=yc-objode.dat.un;
actmalla(objprob,objmalla_n,uc,'n1_n');
objode.dat.un=yc; actpaso(objprob,objmalla_n)
```

```
function actfin(t,y,objode,objprob,objmalla)
na=objprob.ngdla; u=solc(objprob,y(1:na));
set_u(objmalla,u); actpaso(objprob,objmalla)
```

Cada método (*objode*) tendrá su función *paso* específica que contiene su fórmula de avance. Si el método es implícito se deberá recurrir en cada paso al método de Newton (*objnr*) y entonces la fórmula de avance precisará las funciones *calcR*(objmetodo) y *calcJ*(objmetodo) que proporcionan el residuo y la matriz jacobiana del sistema de ecuaciones no lineal resultante.

Las soluciones numéricas se van almacenando en los datos *tsol* e *ysol* del objeto ode (*objode*), que también contiene como dato el objeto *objfty* con la información del sistema de EDOs que estamos resolviendo. Se describe a continuación este objeto *objfty*.

7.3.2.2. Objeto fty

Este objeto se agrega como dato al objeto ode para definir el sistema de EDOs a resolver, proporcionando f y $j = \partial f / \partial y$ del sistema $My' = f(t, y)$ que se pretende resolver.

En particular, gestiona las diferencias existentes entre los problemas de difusión y onda, implementando una tercera vía para la resolución del problema general $My' = f(t, y)$.

La clase *difus* resuelve sistemas de EDOs de orden 1 (7.44) y la clase *onda* sistemas de EDOs de orden 2 (7.38) incluyendo los problemas de elasticidad dinámica. También se ha implementado un caso general para resolver EDOs de orden 1 que no provienen

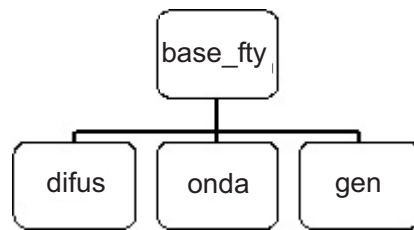


Figura 7.17: Arquitectura de objetos fty por herencia.

de los problemas de onda o difusión al que hemos llamado *gen*, y que precisará dos funciones dato adicionales: la función que define el sistema de EDOs específico a resolver ($f(t, y) \equiv fundat$) y la que define el jacobiano ($\frac{\partial f}{\partial y} \equiv fundat2$).

Ambas clases *difus* y *onda* carecen de datos específicos y comparten las tres funciones (*inicM*, *calcf*, *calcj*) muy parecidas, que se diferencian en la dimensión de f y j que es doble en onda. Ambas recurren al objeto problema para calcular R y J según lo indicado en (7.38) y (7.44), tal como se muestra a continuación para el objeto *onda*.

```
function inicM(objjode,objprob,objmalla) %ONDA
na=objprob.ngdla; M=calcmasa(objprob,objmalla,tipomas);
M=[eye(n2) zeros(n2) ; zeros(n2) M]; objjode.M=M;
```

```
function yp=calcf(t,y,objjode,objprob,objmalla) %ONDA
na=objprob.ngdla; y_n=objjode.dat.un(objprob.gdla);
u=y(1:na)-y_n; v=y(na+1:end);
R=calcR(objprob,objmalla,u); yp=[ v ; R ];
```

```
function j=calcj(objjode,objprob,objmalla) %ONDA
na=objprob.ngdla; J=calcJ(objprob,objmalla);
j=[zeros(n2) , eye(na) ; J , zeros(na)];
```

7.3.2.3. Objeto método ode (2)

Los datos y procedimientos comunes del objeto método ode residen en la clase *base_met* de la que heredan 4 clases adicionales: *odesuite* que gestiona la conexión con los procedimientos de la *odesuite* de Matlab; *rk* que representa los métodos típicos estándar explícitos Runge-Kutta de paso constante; *base_imp* que agrega a los datos del método un objeto del tipo Newton Raphson como los descritos en la Sección 7.2, de la que heredan los métodos implícitos a implementar, en particular el trapezoidal, la familia HHT- α , el método cuasi y la familia de métodos multipaso; y finalmente la clase *lincc* de la que heredan *traplinc*, *alfalinc* y *mplinc* que también corresponden a métodos implícitos pero que implementan las fórmulas explícitas que se derivan en el problema dinámico lineal de coeficientes constantes (*lincc*), fórmulas mucho menos costosas ya que no deben iterar.

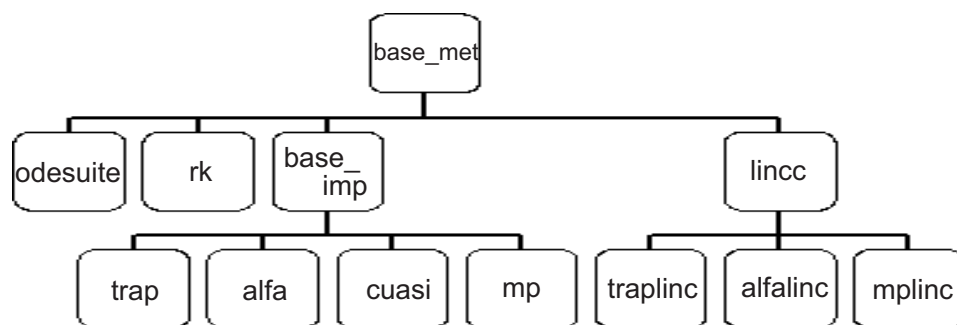


Figura 7.18: Arquitectura de objetos método ODE por herencia.

Todos los métodos implementados avanzan paso a paso según su fórmula característica de avance, demandando del sistema de EDOs siempre el valor de f , expresiones (7.39) y (7.44), y en los métodos implícitos no lineales también el valor de j , (7.39) y (7.44). Describiremos primero el método más general de todos los implementados: el método implícito trapezoidal, que servirá de base para la descripción de los demás métodos.

7.3.2.4. Objeto problema

Este objeto ya ha quedado descrito en su mayor parte en puntos anteriores, por lo que aquí nos limitaremos a describir aspectos adicionales que se reducen a funciones de actualización. En particular la función *actpaso(objprob)* se encarga de actualizar al final de cada paso las variables de almacenamiento del objeto problema llamando a *act_varele(objprob)* que se encarga de actualizar las tensiones (*tensele_n* con los valores *tensele_k* convergidos). También actualiza las variables internas del material, llamando a *act_varint(objmat)*.

7.3.3. Método cuasi-estático

Este es un problema especial donde resolvemos un caso estático no lineal $f(t, y) = 0$, introduciendo un parámetro de carga t . Entonces, el proceso es evolutivo en t , pero con velocidad infinitamente lenta $y' = 0$. No existen fuerzas de inercia ni de amortiguamiento, pero podemos resolver paso a paso $f(t, y) = 0$ utilizando los recursos de actualización entre pasos para el problema dinámico. En cada paso de t_n a t_{n+1} resolvemos mediante Newton Raphson el problema no lineal a partir del valor inicial y_n . Dado el carácter local del método de Newton, el proceso iterativo será mucho más robusto al encontrarse los valores iniciales y_n mucho más cercanos a la solución y_{n+1} .

Basta con descargar los términos asociados a M y C (7.35) para observar que resolveremos (7.18) aprovechando los pasos y actualizaciones del objeto *base_met*. Un nuevo objeto *cuasi* que hereda de *base_met* implementa los aspectos específicos: cálculo de R

y J , (7.18) y (7.19):

$$f(t, y) \equiv F_{ext}(t) - F_{int}(y) = 0 \quad (7.45)$$

iterando de la siguiente manera:

$$y_{n+1}^{m+1} = y_{n+1}^m - (J(y_{n+1}^m))^{-1} R(y_{n+1}^m), \quad m = 0, 1, 2, \dots \quad (7.46)$$

Se puede utilizar como predictor el valor obtenido en el paso anterior ($y_{n+1}^0 = y_n$).

7.3.3.1. Objeto cuasi

- Datos: *xini*, *paso_ad*.
- Funciones: *inic*(objcuasi, objprob, objmalla),
[tn1, yn1, uinc_n, div] = paso(objcuasi, tn1, yn, h, objprob, objmalla),
resolver(objcuasi, objprob, objmalla),
[R, Fext, Reac, Fint] = calcR(u, objcuasi, objprob, objmalla_n),
J = calcJ(u, objjode, objprob, objmalla, objJ).

7.3.4. Método trapezoidal

Consideramos el sistema de EDOs dado de la siguiente manera:

$$M \cdot y' = f(t, y) \quad (7.47)$$

Avanzamos del paso n al paso $(n + 1)$ utilizando el método trapezoidal:

$$M \cdot y_{n+1} = M \cdot y_n + \frac{h}{2} (f_n + f_{n+1}) \quad (7.48)$$

siendo $f_n = f(t_n, y_n)$, $f_{n+1} = f(t_{n+1}, y_{n+1})$.

En cada paso se resuelve el sistema no lineal de ecuaciones (7.48) mediante el método de Newton, pasando de la iteración m a la iteración $(m + 1)$ mediante:

$$y_{n+1}^{m+1} = y_{n+1}^m - (\bar{J}(y_{n+1}^m))^{-1} \bar{R}(y_{n+1}^m), \quad m = 0, 1, 2, \dots \quad (7.49)$$

siendo:

$$\begin{cases} \bar{R}(y_{n+1}^m) = \frac{2}{h} M (y_{n+1}^m - y_n) - (f_n + f_{n+1}^m) \\ \bar{J}(y_{n+1}^m) = \frac{\partial \bar{R}(y_{n+1}^m)}{\partial y_{n+1}} = \frac{2}{h} M - j_{n+1}^m \end{cases} \quad (7.50)$$

donde, las funciones f y j en el caso del problema de onda vienen dadas por (7.39):

$$\begin{cases} f_n = \begin{pmatrix} v_n \\ R(U_n) - Cv_n \end{pmatrix}; \quad f_{n+1}^m = \begin{pmatrix} v_{n+1}^m \\ R(U_{n+1}^m) - Cv_{n+1}^m \end{pmatrix} \\ j_{n+1}^m = \begin{pmatrix} 0 & I \\ \frac{\partial R(U_{n+1}^m)}{\partial U_{n+1}} & -C \end{pmatrix} = \begin{pmatrix} 0 & I \\ J(U_{n+1}^m) & -C \end{pmatrix} \end{cases} \quad (7.51)$$

Para el problema de difusión las expresiones de f y j son más sencillas y vienen dadas por (7.44). Las expresiones de $R(U)$ y $J(U)$ se definen mediante (7.19) y (7.20) en ambos casos. En el caso general *gen*, los valores de f y j deben definirse mediante una función dato específica.

Se puede utilizar como predictor el valor obtenido en el paso anterior ($y_{n+1}^0 = y_n$), y se itera hasta que se alcance el número máximo de iteraciones fijadas y/o hasta que $\|\bar{R}(y_{n+1}^m)\| < tol \approx 0$. También existe la posibilidad de iterar hasta que $\|(\bar{J}(y_{n+1}^m))^{-1} \bar{R}(y_{n+1}^m)\| < tol \approx 0$.

7.3.4.1. Objeto trap

Este objeto implementa el método trapezoidal para resolver EDOs de orden 1. Hereda de *base.met* los datos y funciones indicados en 7.3.2.1, e incorpora los siguientes datos y funciones específicos:

- Datos: M , fk .
- Funciones: *inic*(objjode, objjprob, objjmalla),
 $[tk1, yk1] = paso(tk, yk, h, objjode, varargin)$,
 $Rb = calcR(yk1, tk1, yk, objjode, varargin)$, $Jb = calcJ(u, varargin)$.

La función *paso* resuelve la fórmula de avance (7.48) mediante el método de Newton.

```
function [tk1,yk1] = paso(tk,yk,h,objjode,varargin)
tk1 = tk+h;
objjfty=objjode.objjfty;
objjode.fk=objjfty.calcf(tk,yk,objjode,varargin{:});
objjode.objjnr.xini=yk;
[~,~,yk1]=resolver(objjode.objjnr,tk1,yk,objjode,varargin{:}); %Orden exacto
yk1=yk1(:,end);
actpaso(tk1,yk1,objjode,varargin{:})
```

La función *resolver*(objjnr) de la Sección 7.2 ejecuta las iteraciones indicadas en (7.49), llamando a *calcR*(objjode) y *calcJ*(objjode) que evalúan (7.50), llamando a *calcf*(objjode.objjfty) y *calcj*(objjode.objjfty) que proporcionan (7.44) o (7.51). Para calcular (7.44) o (7.51) se llama a *calcR*(objjprob) y *calcJ*(objjprob) que son las que calculan R y J según lo indicado en (7.19), (7.20), (7.21) y (7.22).

Las siguientes funciones son las que calculan \bar{R} y \bar{J} de las fórmulas (7.50), en las variables Rb y Jb respectivamente.

```
function Rb=calcR(objjode,yk1,tk1,yk,objjprob,objjmalla)
objjfty=objjode.objjfty; fk=objjode.fk; M=objjode.M;
fk1=calcf(objjfty,tk1,yk1,objjode,objjprob,objjmalla);
h=objjode.h; Rb=2*M/h*(yk1-yk)-(fk+fk1);
```

```
function Jb=calcJb(objjode,yk1,tk1,objjode,objjprob,objjmall)
M=objjode.M; h=objjode.h; objjfty=objjode.objjfty;
j=calcj(objjfty,objjode,objjprob,objjmall); Jb=2*M/h-j;
```

7.3.4.2. Objeto traplinc

Este objeto implementa la particularización del método trapezoidal en el caso lineal de coeficientes constantes, en un algoritmo explícito de avance donde no hay que iterar. La fórmula de avance se simplifica de manera muy importante, dando lugar a un producto de matrices constantes que se calculan una única vez en la inicialización.

Sustituyendo la relación lineal $F_{int}(d) = Kd$ en (7.37) y (7.42), obtenemos el sistema de EDOs:

$$M \cdot y' = A \cdot y + g(t) \quad (7.52)$$

- siendo para el problema de Laplace: $A \rightarrow -K$, $y \rightarrow d(t)$, $g(t) \rightarrow F_{ext}(t)$
- para el problema de la onda:

$$M \rightarrow \begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}, \quad A \rightarrow \begin{pmatrix} 0 & I \\ -K & -C \end{pmatrix}, \quad y \rightarrow \begin{pmatrix} d(t) \\ v(t) \end{pmatrix}, \quad g(t) \rightarrow \begin{pmatrix} 0 \\ F_{ext}(t) \end{pmatrix}$$

- y en el caso general *gen* tendremos una ecuación de la forma (7.52) a la que se le proporcionan M , A y $g(t)$.

Aplicando la formula de avance trapezoidal (7.47) al sistema (7.52) se tiene:

$$My_{n+1} = My_n + \frac{h}{2} (Ay_n + g(t_n) + Ay_{n+1} + g(t_{n+1})) \quad (7.53)$$

y despejando el valor de y_{n+1} de (7.53):

$$\left(M - \frac{h}{2}A\right)y_{n+1} = \left(M + \frac{h}{2}A\right)y_n + \frac{h}{2}(g(t_n) + g(t_{n+1})) \quad (7.54)$$

$$y_{n+1} = C_1 \left[C_2 y_n + \frac{h}{2}(g(t_n) + g(t_{n+1})) \right] \quad (7.55)$$

siendo: $C_1 = (M - \frac{h}{2}A)^{-1}$, $C_2 = (M + \frac{h}{2}A)$.

Todo ello queda implementado en el objeto *traplinc* que hereda de *base_met* (ver Figura 7.18) los datos y funciones comunes y añade los datos C_1 , C_2 y gk y las dos funciones específicas, *inic* y *paso*, que se indican a continuación:

```
function inic(objode,objprob,objmalla)
M=inicM(objode.objfity,objode,objprob,objmalla);
A=Alin(objode.objfity,objprob,objmalla);
tini=objode.interv(1); h=objode.h;
g0=g(tini,objode.objfity,objprob,objmalla);
objode.gk=g0; I=speye(size(A));
objode.C1=(M-A*h/2)\I; objode.C2=M+A*h/2;
```

```
function [tk1,yk1] = paso(tk,yk,h,objode,objprob,objmalla)
objfity=objode.objfity; TI=objode.C2*yk; gk=objode.gk;
tk1=tk+h;
if ~isempty(gk)
gk1=objfity.g(tk1,objode,objprob,objmalla);
TI=TI+(gk+gk1)*(h/2); objode.gk=gk1;
end
yk1=objode.C1*TI;
```

La matriz A y el vector g definidos en (7.52) toman formas distintas para onda y difusión. Para gestionar esta diferencia, en el objeto ode disponemos del objeto agregado *objfity* (*difus*, *onda*, *gen*) a la que pertenecen las funciones *Alin* y *g* que se llaman desde la función *paso*. Se listan a continuación ambas funciones correspondientes al objeto *fty onda*.

```
function A=onda.Alin(objprob,objmalla)
K=rigidez(objprob,objmalla);
na=objprob.ngdla;
A=[sparse(na,na) speye(na);-K sparse(na,na)];
```

```
function gt=onda.g(t,objprob,objmalla)
Fext=calfext(objprob,objmalla,t); na=length(Fext);
gt=[]; if any(Fext), gt =[zeros(na,1);Fext]; end
```

7.3.5. Métodos multipaso

En esta sección programaremos algunos métodos multipaso lineales. En concreto, hemos creado los objetos correspondientes a los métodos: BDF, NDF, BDF- α , la familia EBDF (EBDF, EBPDF, ENPDF, ENDF) y la familia MEBDF (MEBDF, MEBPDF, MENPDF, MENDF). Hemos procedido de similar manera que con el método trapecoidal, aplicando primero cada método a la EDO genérica de orden 1 dada por (7.47) y particularizándolo al problema lineal de coeficientes constantes (7.52) a continuación.

En la Figura 7.19 se muestra la arquitectura de los objetos multipaso para problemas no lineales. En *mp* se tienen dos funciones (*pasos_ini* y *act_paso*). La primera sirve para inicializar el método (*pasos_ini*) y la otra, una vez dado el paso, actualiza los valores que se necesitan para el siguiente paso. Un método multipaso de k pasos necesita k valores

$\{y_n, y_{n+1}, \dots, y_{n+k-1}\}$ para calcular el valor y_{n+k} correspondiente al instante t_{n+k} . En la inicialización sólo se tiene el valor y_0 ($y_n = y_0$), los $(k - 1)$ valores restantes que se necesitan para dar el paso se calculan con la función *pasos_ini* utilizando el método trapezoidal. Una vez que se tienen los k valores necesarios, se dan los pasos con el método multipaso elegido.

Objeto mp

- Datos: No tiene datos específicos.
- Funciones: *pasos_ini*(objode,objprob,objmalla), *actpaso*(tk1,yk1,objode,varargin).

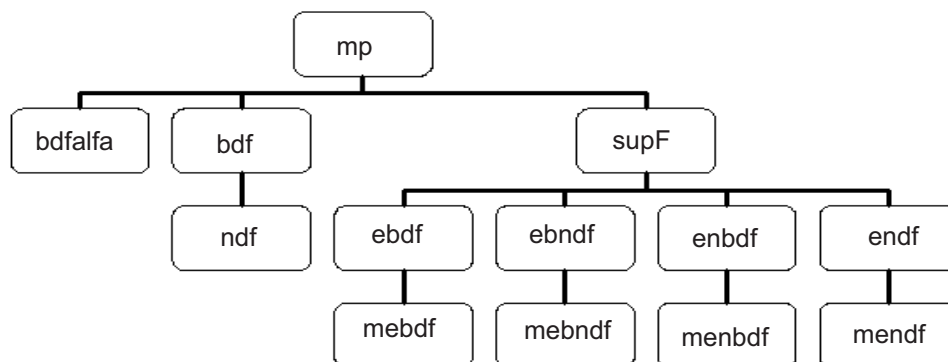


Figura 7.19: Arquitectura de objetos métodos multipaso no lineales por herencia.

La Figura 7.20 muestra el esquema seguido en los métodos multipaso particularizados al caso de un problema lineal de coeficientes constantes. De nuevo, en *mplinc* tenemos las funciones de arranque y de actualización. Esta vez, se han calculado utilizando el método trapezoidal lineal los $(k - 1)$ valores que se requieren para dar el paso.

Objeto mplinc

- Datos: No tiene datos específicos.
- Funciones: *pasos_ini*(objode, objprob, objmalla),
actpaso(tk1, yk1, objode, objprob, objmalla).

7.3.5.1. Método BDF

Consideramos la EDO genérica (7.47) y avanzamos del paso $(n+k-1)$ al paso $(n+k)$ utilizando el método BDF dado por (2.41):

$$\sum_{j=0}^k \hat{\alpha}_j M y_{n+j} = h f_{n+k} \quad (7.56)$$

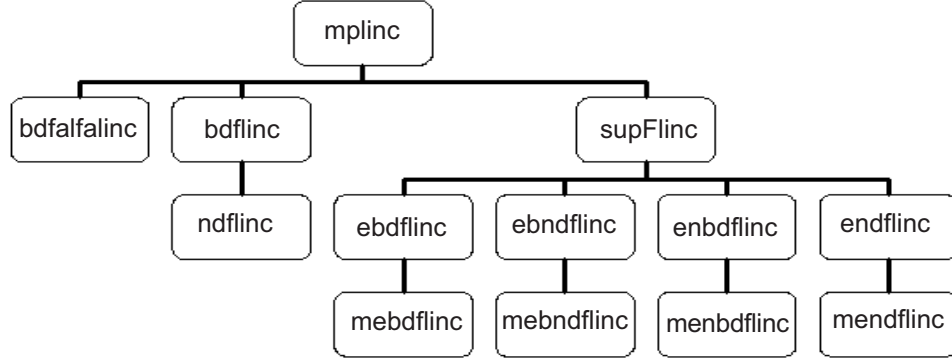


Figura 7.20: Arquitectura de objetos métodos multipaso lineales por herencia.

siendo $f_{n+k} = f(t_{n+k}, y_{n+k})$ y $\hat{\alpha}_j$ los coeficientes del método BDF recogidos en la Tabla 2.3. En (7.56) debemos calcular el valor de y_{n+k} a partir de los valores anteriores $\{y_n, y_{n+1}, \dots, y_{n+k-1}\}$. Como f es función no lineal de y , no podemos obtener una expresión explícita de y_{n+k} en función de los valores anteriores y utilizamos el método iterativo de Newton Raphson para calcular y_{n+k} (7.49), siendo \bar{R} y \bar{J} :

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} \left(\hat{\alpha}_k M y_{n+k}^m + \sum_{j=0}^{k-1} \hat{\alpha}_j M y_{n+j} \right) - f_{n+k}^m \\ \bar{J}(y_{n+k}^m) = \frac{\partial \bar{R}(y_{n+k}^m)}{\partial y_{n+k}} = \frac{1}{h} \hat{\alpha}_k M - j_{n+k}^m \end{cases} \quad (7.57)$$

donde, f_{n+k} y j_{n+k} vienen dadas por (7.39) para elasticidad dinámica y para el problema de onda, y por (7.44) en el problema de difusión, ambas evaluadas en el instante t_{n+k} . Por ejemplo, en el caso de elasticidad dinámica y onda quedan:

$$\begin{cases} f_{n+k}^m = \begin{pmatrix} v_{n+k}^m \\ R(U_{n+k}^m) - C v_{n+k}^m \end{pmatrix} \\ j_{n+k}^m = \begin{pmatrix} 0 & I \\ \frac{\partial R(U_{n+k}^m)}{\partial U_{n+k}} & -C \end{pmatrix} = \begin{pmatrix} 0 & I \\ J(U_{n+k}^m) & -C \end{pmatrix} \end{cases} \quad (7.58)$$

A la hora de programar hemos introducido la variable C_2 que nos ha permitido generalizar la expresión \bar{R} de (7.57) para cualquier orden de la siguiente manera:

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} (\hat{\alpha}_k M y_{n+k}^m + C_2 Y_{n+k-1}) - f_{n+k}^m \\ \bar{J}(y_{n+k}^m) = \frac{1}{h} \hat{\alpha}_k M - j_{n+k}^m \end{cases} \quad (7.59)$$

siendo:

$$\begin{cases} C_2 = \left(\hat{\alpha}_{k-1} M \quad \hat{\alpha}_{k-2} M \quad \hat{\alpha}_{k-3} M \quad \dots \quad \hat{\alpha}_1 M \quad \hat{\alpha}_0 M \right) \\ Y_{n+k-1} = \left(y_{n+k-1} \quad y_{n+k-2} \quad y_{n+k-3} \quad \dots \quad y_{n+1} \quad y_n \right)^T \end{cases} \quad (7.60)$$

Objeto bdf

- Datos: M , $alfas$, C_2 , $ykm1$.
- Funciones: $inic(objode,objprob,objmalla)$,
 $[tk1,yk1]=paso(tk,yk,h,objode,varargin)$,
 $Rb=calcR(yk1,tk1,yk,objode,varargin)$, $Jb=calcJ(u,varargin)$.

El objeto *bdf* hereda de *mp* las funciones comunes y añade los datos M , $alfas$, C_2 e $ykm1$ específicos del método. Las funciones de inicialización, *paso*, *calcR* y *calcJ* del objeto *bdf* se pueden ver en el Anexo D.

Tras haber dado el paso, se utiliza la función de actualización de los métodos multi-paso, con el que se actualiza el vector Y_{n+k-1} .

Objeto bdfinc

- Datos: $alfas$, C_1 , C_2 , $ykm1$.
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1,yk1] = paso(tk, yk, h, objode, objprob, objmalla)$.

Este objeto implementa la particularización del método BDF al caso lineal de coeficientes constantes dado por (7.52): $M \cdot y' = A \cdot y + g(t)$. Aplicando la fórmula de avance del BDF a este sistema resulta:

$$\sum_{j=0}^k \hat{\alpha}_j M y_{n+j} = h (A y_{n+k} + g(t_{n+k})) \quad (7.61)$$

Agrupamos en la parte izquierda de la igualdad (7.61) la incógnita y_{n+k} :

$$(\hat{\alpha}_k M - hA) y_{n+k} = - \sum_{j=0}^{k-1} \hat{\alpha}_j M y_{n+j} + hg(t_{n+k}) \quad (7.62)$$

Los términos $\{y_{n+j} : j = 0, 1, \dots, k-1\}$ que se tienen a la derecha de la expresión (7.62) son los valores de la incógnita en los pasos anteriores. Igual que hemos hecho en el caso genérico de las BDF, utilizamos C_2 e Y_{n+k-1} dados por (7.60) para escribir la expresión (7.62):

$$(\hat{\alpha}_k M - hA) y_{n+k} = -C_2 Y_{n+k-1} + hg(t_{n+k}) \quad (7.63)$$

Y despejando y_{n+k} de la expresión (7.63) se tiene:

$$y_{n+k} = C_1 (-C_2 Y_{n+k-1} + hg(t_{n+k})) \quad (7.64)$$

siendo $C_1 = (\hat{\alpha}_k M - hA)^{-1}$, y C_2 y Y_{n+k-1} dados por (7.60).

El objeto *bdfinc* hereda de *mplinc* las funciones comunes y añade los datos que son específicos del método. Las funciones de inicialización y de paso del objeto *bdfinc* se han recogido en el Apéndice D.

7.3.5.2. Método NDF

En la Sección 2.2.5 vimos los método NDF, que consisten en añadir la diferencia hacia atrás de orden $(k + 1)$ en la expresión de los métodos BDF de orden k . Consideramos la EDO genérica dada por (7.47) y avanzamos del paso $(n + k - 1)$ al paso $(n + k)$ utilizando el método NDF:

$$\sum_{j=0}^k \hat{\alpha}_j M y_{n+j} = h f_{n+k} + \boxed{\kappa \gamma_k M \nabla^{k+1} y_{n+k}} \quad (7.65)$$

donde $f_{n+k} = f(t_{n+k}, y_{n+k})$. La expresión (7.56) de los métodos BDF difiere de la expresión (7.65) correspondiente a NDFs, en la parte que en esta última hemos encuadrado. Desarrollando la diferencia hacia atrás de orden $(k + 1)$, la expresión (7.65) queda de la siguiente manera:

$$\sum_{j=0}^k \hat{\alpha}_j M y_{n+j} = h f_{n+k} + \kappa \gamma_k M \sum_{j=0}^{k+1} (-1)^j \binom{k+1}{j} y_{n+k-j} \quad (7.66)$$

De igual manera que en el método BDF, en cada paso NDF se aplica el método de Newton (7.57), hasta que $\|\bar{R}(y_{n+k}^m)\| < tol \approx 0$, siendo:

$$\left\{ \begin{array}{l} \bar{R}(y_{n+k}^m) = \frac{1}{h} \left((\hat{\alpha}_k M - \kappa \gamma_k M) y_{n+k}^m + \sum_{j=0}^{k-1} \hat{\alpha}_j M y_{n+j} \right. \\ \quad \left. - \kappa \gamma_k M \sum_{j=1}^{k+1} (-1)^j \binom{k+1}{j} y_{n+k-j} \right) - f_{n+k}^m \\ \bar{J}(y_{n+k}^m) = \frac{\partial \bar{R}(y_{n+k}^m)}{\partial y_{n+k}} = \frac{1}{h} (\hat{\alpha}_k M - \kappa \gamma_k M) - j_{n+k}^m \end{array} \right. \quad (7.67)$$

donde f_{n+k} y j_{n+k} vienen dadas por (7.39) y (7.44).

El sumando adicional que se añade en los métodos NDF hace que en la expresión (7.66) aparezca un valor adicional y_{n-1} correspondiente al instante t_{n-1} y que no lo teníamos en el método BDF. Esto significa que para calcular y_{n+k} utilizando (7.65), se requieren los valores anteriores $\{y_{n-1}, y_n, \dots, y_{n+k-1}\}$ y que el vector Y_{n+k-1} que hemos definido en el caso de las BDF mediante (7.60) tendrá este elemento adicional.

Con el objeto de aprovechar lo construido en BDFs, identificamos en las expresiones de \bar{R} y \bar{J} de los métodos NDF lo que tienen en común con las \bar{R} y \bar{J} de las BDF. Esta

parte común es la que se ha encuadrado en las siguientes expresiones:

$$\left\{ \begin{array}{l} \bar{R}(y_{n+k}^m) = \frac{1}{h} \left(\hat{\alpha}_k M y_{n+k}^m + \sum_{j=0}^{k-1} \hat{\alpha}_j M y_{n+j} \right) - f_{n+k}^m \\ \quad + \frac{1}{h} \left(-\kappa \gamma_k M y_{n+k}^m - \kappa \gamma_k M \sum_{j=1}^{k+1} (-1)^j \binom{k+1}{j} y_{n+k-j} \right) \\ \bar{J}(y_{n+k}^m) = \frac{1}{h} \hat{\alpha}_k M - j_{n+k}^m - \frac{1}{h} \kappa \gamma_k M \end{array} \right. \quad (7.68)$$

De esta manera, utilizando \bar{R} y \bar{J} de las BDF y definiendo las siguientes variables:

$$\left\{ \begin{array}{l} C_4 = \left(\kappa \gamma_k M (-1)(k+1), \kappa \gamma_k M (-1)^2 \binom{k+1}{2}, \dots \right. \\ \quad \left. \kappa \gamma_k M (-1)^k \binom{k+1}{k}, \kappa \gamma_k M (-1)^{k+1} \binom{k+1}{k+1} \right) \\ \tilde{Y}_{n+k-1} = \left(y_{n+k-1} \quad y_{n+k-2} \quad y_{n+k-3} \quad \dots \quad y_{n+1} \quad y_n \quad y_{n-1} \right)^T \end{array} \right. \quad (7.69)$$

las expresiones de (7.68) de \bar{R} y \bar{J} de los métodos NDF quedan:

$$\left\{ \begin{array}{l} \bar{R}(y_{n+k}^m) = \bar{R}_{BDF} - \frac{1}{h} \kappa \gamma_k M y_{n+k}^m - \frac{1}{h} C_4 \tilde{Y}_{n+k-1} \\ \bar{J}(y_{n+k}^m) = \bar{J}_{BDF} - \frac{1}{h} \kappa \gamma_k M \end{array} \right. \quad (7.70)$$

Objeto ndf

- Datos: $kappa$, $gamma$, C_4 .
- Funciones: $inic(objode,objprob,objmalla)$,
 $Rb=calcR(yk1,tk1,yk,objode,varargin)$, $Jb=calcJ(u,varargin)$.

El objeto *ndf* hereda del objeto *bdf* los datos y funciones comunes y añade los datos $kappa$, $gamma$ y C_4 . La inicialización la hace utilizando *pasos.ini* dando un paso adicional que las BDF para poder completar \tilde{Y}_{n+k-1} (k pasos). Se pueden ver en el Apéndice D las funciones de inicialización, *calcR* y *calcJ* de este objeto.

Objeto ndfinc

- Datos: κ , γ , C_4 .
- Funciones: $inic(objode,objprob,objmalla)$.

Este objeto implementa la particularización del método NDF al caso lineal de coeficientes constantes (7.52). Aplicando la fórmula de avance de las NDF a este sistema resulta:

$$\sum_{j=0}^k \hat{\alpha}_j M y_{n+j} = h (A y_{n+k} + g(t_{n+k})) + \boxed{\kappa \gamma_k M \nabla^{k+1} y_{n+k}} \quad (7.71)$$

siendo la parte recuadrada de (7.71) la proveniente del sumando adicional de los métodos NDF. Desarrollamos la diferencia hacia atrás y agrupamos en la parte izquierda de la igualdad (7.71) la incógnita y_{n+k} :

$$(\hat{\alpha}_k M - hA - \kappa \gamma_k M) y_{n+k} = - \sum_{j=0}^{k-1} \hat{\alpha}_j M y_{n+j} + \kappa \gamma_k M \sum_{j=1}^{k+1} (-1)^j \binom{k+1}{j} y_{n+k-j} + hg(t_{n+k}) \quad (7.72)$$

Utilizando las variables C_2, Y_{n+k-1} dadas por (7.60) y C_4, \tilde{Y}_{n+k-1} dadas por (7.69), la expresión (7.72) queda:

$$(\alpha_k M - hA - \kappa \gamma_k M) y_{n+k} = -C_2 Y_{n+k-1} + C_4 \tilde{Y}_{n+k-1} + hg(t_{n+k}) \quad (7.73)$$

Y despejando y_{n+k} de la expresión (7.73) se tiene:

$$y_{n+k} = C_1 \left(-C_2 Y_{n+k-1} + C_4 \tilde{Y}_{n+k-1} + hg(t_{n+k}) \right) \quad (7.74)$$

siendo: $C_1 = (\alpha_k M - hA - \kappa \gamma_k M)^{-1}$.

Con el fin de que el objeto *ndfinc* utilice la misma función paso que el objeto *bdfinc*, a la expresión (7.74) le hemos dado la forma (7.64) redimensionando la matriz C_2 a la misma dimensión que C_4 añadiendo ceros. A esta nueva matriz le hemos llamado $C_{2,0}$, con lo que (7.74) queda:

$$y_{n+k} = C_1 \left(-C_{2,0} \tilde{Y}_{n+k-1} + C_4 \tilde{Y}_{n+k-1} + hg(t_{n+k}) \right) \Rightarrow \\ y_{n+k} = C_1 \left((-C_{2,0} + C_4) \tilde{Y}_{n+k-1} + hg(t_{n+k}) \right) \quad (7.75)$$

Y haciendo $\tilde{C}_2 = C_{2,0} - C_4$, la expresión (7.75) queda:

$$y_{n+k} = C_1 \left(-\tilde{C}_2 \tilde{Y}_{n+k-1} + hg(t_{n+k}) \right) \quad (7.76)$$

La expresión (7.76) tiene la misma forma que la que tiene el método BDF aplicado a un problema lineal (7.64), lo que hace posible que el objeto *ndfinc* utilice la misma función paso que el objeto *bdfinc*.

El objeto *ndflinc* hereda de *bdfinc* los datos y funciones comunes y añade los datos *kappa*, *gamma* y *C4*. Adecúa los datos *C1*, *C2* e *ykm1* que hereda del objeto *bdfinc*, al ser éstos distintos para el caso *ndflinc*. El método arranca utilizando la función *pasos_ini* con el que da *k* pasos utilizando el método trapezoidal lineal y completando así \tilde{Y}_{n+k-1} .

7.3.5.3. Método BDF- α

En el Capítulo 6 vimos que el método BDF- α es un método multipaso ($k = 2$) y de orden 2, por lo que necesita los valores de los dos pasos anteriores, y_{n+1} , y_n , para poder avanzar un paso. Su fórmula de avance viene dada por la expresión (6.21):

$$\left(\frac{3}{2} + \alpha\right) y_{n+2} + (-2 - 2\alpha) y_{n+1} + \left(\frac{1}{2} + \alpha\right) y_n = h(1 + \alpha)f_{n+2} - h\alpha f_{n+1}$$

Simplificaremos la escritura de la expresión anterior de la siguiente manera:

$$\sum_{j=0}^2 \beta_j y_{n+j} = h(1 + \alpha)f_{n+2} - h\alpha f_{n+1} \quad (7.77)$$

con $\beta_0 = \frac{1}{2} + \alpha$, $\beta_1 = -2 - 2\alpha$, $\beta_2 = \frac{3}{2} + \alpha$.

Consideramos la EDO genérica (7.47) y avanzamos del paso ($n + 1$) al paso ($n + 2$) utilizando este método:

$$\sum_{j=0}^2 \beta_j M y_{n+j} = h(1 + \alpha)f_{n+2} - h\alpha f_{n+1} \quad (7.78)$$

En cada paso de este objeto también se aplica el objeto *objnr* hasta que $\|\bar{R}(y_{n+2}^m)\| < tol \approx 0$, siendo:

$$\begin{cases} \bar{R}(y_{n+2}^m) = \frac{M}{h} (\beta_2 y_{n+2}^m + \beta_1 y_{n+1} + \beta_0 y_n) - ((1 + \alpha)f_{n+2}^m - \alpha f_{n+1}) \\ \bar{J}(y_{n+2}^m) = \frac{\partial R(y_{n+2}^m)}{\partial y_{n+2}} = \frac{M}{h} \beta_2 - (1 + \alpha)j_{n+2}^m \end{cases} \quad (7.79)$$

Con el objetivo de utilizar el mismo esquema en todas las multipaso, la expresión (7.79) la hemos programado de esta forma:

$$\begin{cases} \bar{R}(y_{n+2}^m) = \frac{1}{h} (\beta_2 M y_{n+2}^m + C_2 Y_{n+1}) - ((1 + \alpha)f_{n+2}^m - \alpha f_{n+1}) \\ \bar{J}(y_{n+2}^m) = \frac{\partial R(y_{n+2}^m)}{\partial y_{n+2}} = \frac{M}{h} \beta_2 - (1 + \alpha)j_{n+2}^m \end{cases} \quad (7.80)$$

$$\text{donde: } C_2 = (\beta_1 M \quad \beta_0 M), \quad Y_{n+1} = (y_{n+1} \quad y_n)^T \quad (7.81)$$

Objeto bdfalfa

- Datos: $M, C_2, fk, beta, alfa, ykm1$.
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1, yk1] = paso(tk, yk, h, objode, varargin)$,
 $Rb = calcR(yk1, tk1, yk, objode, varargin)$,
 $Jb = calcJ(u, varargin)$.

El objeto *bdfalfa* hereda de *mp* los datos y funciones comunes y añade los datos $M, alfa, C_2, beta, fk$ e $ykm1$ específicos del método.

Objeto bdfalfalinc

- Datos: $C_1, C_2, beta, alfa, ykm1, gk$.
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1, yk1] = paso(tk, yk, h, objode, objprob, objmalla)$.

El objeto *bdfalfalinc* es la particularización del BDF- α para el caso lineal de coeficientes constantes. Aplicamos la fórmula de avance del método BDF- α al sistema lineal (7.52):

$$\sum_{j=0}^2 \beta_j M y_{n+j} = h(1 + \alpha) (A y_{n+2} + g(t_{n+2})) - h\alpha (A y_{n+1} + g(t_{n+1})) \quad (7.82)$$

Agrupamos en la parte izquierda de la igualdad (7.82) la incógnita y_{n+2} :

$$(\beta_2 M - h(1 + \alpha)A) y_{n+2} = -(\beta_1 M y_{n+1} + \beta_0 M y_n) - h\alpha A y_{n+1} + h(1 + \alpha)g(t_{n+2}) - h\alpha g(t_{n+1}) \quad (7.83)$$

Y utilizando las expresiones de C_2 e Y_{n+1} dadas por (7.81) y despejando y_{n+2} de (7.83) se tiene:

$$y_{n+2} = C_1 [-C_2 Y_{n+1} - h\alpha A y_{n+1} + h(1 + \alpha)g(t_{n+2}) - h\alpha g(t_{n+1})] \quad (7.84)$$

donde C_1, C_2, Y_{n+1} vienen dados por:

$$\begin{cases} C_1 = (\beta_2 M - h(1 + \alpha)A)^{-1}, \\ C_2 = (\beta_1 M \quad \beta_0 M) \\ Y_{n+1} = (y_{n+1} \quad y_n)^T \end{cases} \quad (7.85)$$

El objeto *bdfalfalinc* hereda de *mplinc* los datos y funciones comunes y añade los datos $C_1, C_2, alfa, beta, ykm1$ y gk . El método BDF- α lineal se inicializa mediante la función *pasos.ini* y al ser un método de 2 pasos sólo necesita dar un paso en el arranque.

7.3.5.4. Método EBDF

En el Capítulo 5 introducimos los métodos EBDF que consisten en predecir dos veces utilizando el método BDF y después corregir mediante la expresión (5.6):

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\beta_k f_{n+k} + h\beta_{k+1} \bar{f}_{n+k+1}$$

Las etapas predictoras las damos aprovechando el objeto *bdf*. Si aplicamos el corrector EBDF a la EDO genérica (7.47), avanzando del paso $(n+k-1)$ al paso $(n+k)$:

$$\sum_{j=0}^k \alpha_j M y_{n+j} = h\beta_k f_{n+k} + h\beta_{k+1} \bar{f}_{n+k+1} \quad (7.86)$$

También se aplica el método de Newton Raphson en el corrector hasta que $\|\bar{R}(y_{n+k}^m)\| < tol \approx 0$, siendo:

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} \left(\alpha_k M y_{n+k}^m + \sum_{j=0}^{k-1} \alpha_j M y_{n+j} \right) - \beta_k f_{n+k}^m - \beta_{k+1} \bar{f}_{n+k+1} \\ \bar{J}(y_{n+k}^m) = \frac{\partial R(y_{n+k}^m)}{\partial y_{n+k}} = \frac{1}{h} \alpha_k M - \beta_k j_{n+k}^m \end{cases} \quad (7.87)$$

Simplificamos la expresión (7.87) introduciendo las variables C_2 e Y_{n+k-1} dadas por:

$$\begin{cases} C_2 = \left(\alpha_{k-1} M & \alpha_{k-2} M & \alpha_{k-3} M & \dots & \alpha_1 M & \alpha_0 M \right) \\ Y_{n+k-1} = \left(y_{n+k-1} & y_{n+k-2} & y_{n+k-3} & \dots & y_{n+1} & y_n \right)^T \end{cases} \quad (7.88)$$

Y utilizando (7.88), las expresiones de \bar{R} y \bar{J} quedan:

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} \left(\alpha_k M y_{n+k}^m + C_2 Y_{n+k-1} \right) - \beta_k f_{n+k}^m - \beta_{k+1} \bar{f}_{n+k+1} \\ \bar{J}(y_{n+k}^m) = \frac{1}{h} \alpha_k M - \beta_k j_{n+k}^m \end{cases} \quad (7.89)$$

Objeto supF

- Datos: AA, BB, C_2 .
- Funciones: $Rb = calcR(yk1, tk1, yk, objode, varargin)$, $Jb = calcJ(u, varargin)$.

Hemos agrupado en el objeto *supF* los métodos multipaso que utilizan puntos super-futuros (familia EBDF y familia MEBDF). Este objeto consta de dos funciones *calcR* y *calcJ* que heredarán los métodos de estas dos familias. Son propiedades de este objeto

la matriz C_2 , la matriz de coeficientes α_j a la que hemos llamado $AA = (\alpha_j)$ y la matriz de coeficientes $BB = (\beta_j)$.

Objeto *ebdf*

- Datos: $M, ykm1, fbk1, fbk2, objbdf$.
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1, yk1] = paso(tk, yk, h, objode, varargin)$.

El objeto *ebdf* hereda de *supF* los datos y funciones comunes y añade los datos $M, ykm1, fbk1, fbk2$ y *objbdf*. En el arranque utiliza *pasos_ini* y utiliza *actpaso* en la actualización.

Objeto *supF*linc

- Datos: AA, BB, C_2 .
- Funciones: No tiene funciones específicas.

Hemos agrupado en el objeto *supF*linc la versión lineal de los métodos multipaso que utilizan puntos super-futuros (familia EBDF y familia MEBDF). Este objeto no tiene funciones específicas y tiene las mismas propiedades que el objeto *supF* ($C_2, AA = (\alpha_j)$ y $BB = (\beta_j)$).

Objeto *ebdf*linc

- Datos: $C_1, ykm1, gk1, objbdflinc.$
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1, yk1] = paso(tk, yk, h, objode, objprob, objmalla)$.

El objeto *ebdf*linc consiste en aplicar el método EBDF al caso lineal de coeficientes constantes. Utiliza el objeto *bdf*linc en las etapas predictoras. Por tanto, sólo falta construir la parte correctora del método EBDF. Para ello aplicamos la expresión (5.6) al sistema lineal (7.52):

$$\sum_{j=0}^k \alpha_j M y_{n+j} = h\beta_k (A y_{n+k} + g(t_{n+k})) + h\beta_{k+1} (A \bar{y}_{n+k+1} + g(t_{n+k+1})) \quad (7.90)$$

Agrupamos en la parte izquierda de la igualdad (7.90) los valores de y_{n+k} :

$$(\alpha_k M - h\beta_k A) y_{n+k} = - \sum_{j=0}^{k-1} \alpha_j M y_{n+j} + h\beta_k g(t_{n+k}) + h\beta_{k+1} (A \bar{y}_{n+k+1} + g(t_{n+k+1})) \quad (7.91)$$

Despejamos y_{n+k} de la expresión (7.91), resultando:

$$y_{n+k} = C_1 (-C_2 Y_{n+k-1} + h\beta_{k+1} A \bar{y}_{n+k+1} + h\beta_k g(t_{n+k}) + h\beta_{k+1} g(t_{n+k+1})) \quad (7.92)$$

siendo $C_1 = (\alpha_k M - h\beta_k A)^{-1}$, y C_2 y Y_{n+k-1} dados por (7.88).

El objeto *ebdfinc* hereda de *supFinc* los datos comunes y añade los datos C_1 , $ykml$, $gk1$ y *objbdfinc*. En el arranque utiliza *pasos_ini* y tras dar el paso utiliza *actpaso* para la actualización.

7.3.5.5. Método MEBDF

En el Capítulo 5 introdujimos los métodos MEBDF que consisten en predecir dos veces utilizando el método BDF, corrigiendo a continuación mediante la expresión (5.7):

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\hat{\beta}_k f_{n+k} + h\beta_{k+1} \bar{f}_{n+k+1} + h(\beta_k - \hat{\beta}_k) \bar{f}_{n+k}$$

Las constantes α_j , β_k y β_{k+1} de este método coinciden con las del método EBDF y los coeficientes $\hat{\beta}_k$ son los del método BDF cuando éste se escribe como (5.8). Utilizamos el objeto *bdfinc* en las etapas predictoras y al aplicar el corrector MEBDF al problema (7.47) se obtiene:

$$\sum_{j=0}^k \alpha_j M y_{n+j} = h\hat{\beta}_k f_{n+k} + h\beta_{k+1} \bar{f}_{n+k+1} + h(\beta_k - \hat{\beta}_k) \bar{f}_{n+k} \quad (7.93)$$

En cada paso de este método se aplica el método de Newton Raphson, hasta que $\|\bar{R}(y_{n+k}^m)\| < tol \approx 0$, siendo \bar{R} y \bar{J} :

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} \left(\alpha_k M y_{n+k}^m + \sum_{j=0}^{k-1} \alpha_j M y_{n+j} \right) - \hat{\beta}_k f_{n+k}^m \\ \quad - \beta_{k+1} \bar{f}_{n+k+1} - (\beta_k - \hat{\beta}_k) \bar{f}_{n+k} \\ \bar{J}(y_{n+k}^m) = \frac{\partial \bar{R}(y_{n+k}^m)}{\partial y_{n+k}} = \frac{1}{h} \alpha_k M - \hat{\beta}_k j_{n+k}^m \end{cases} \quad (7.94)$$

Como en casos anteriores escribimos la expresión (7.94) de esta forma:

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} (\alpha_k M y_{n+k}^m + C_2 Y_{n+k-1}) - \hat{\beta}_k f_{n+k}^m - \beta_{k+1} \bar{f}_{n+k+1} - (\beta_k - \hat{\beta}_k) \bar{f}_{n+k} \\ \bar{J}(y_{n+k}^m) = \frac{\partial \bar{R}(y_{n+k}^m)}{\partial y_{n+k}} = \frac{1}{h} \alpha_k M - \hat{\beta}_k j_{n+k}^m \end{cases} \quad (7.95)$$

donde C_2 e Y_{n+k-1} vienen dados por (7.88).

Comparando las expresiones de \bar{R} y \bar{J} de los métodos EBDF y MEBDF, (7.89) y (7.95) respectivamente, nos damos cuenta de que es posible utilizar un único *calcR* y

$calcJ$ para los dos métodos, ya que ambos tienen la forma:

$$\begin{cases} \bar{R}(y_{n+k}^m) = \frac{1}{h} (\alpha_k M y_{n+k}^m + C_2 Y_{n+k-1}) - \gamma_k f_{n+k}^m - \beta_{k+1} \bar{f}_{n+k+1} - \bar{\gamma}_k \bar{f}_{n+k} \\ \bar{J}(y_{n+k}^m) = \frac{\partial \bar{R}(y_{n+k}^m)}{\partial y_{n+k}^m} = \frac{1}{h} \alpha_k M - \gamma_k J_{n+k}^m \end{cases} \quad (7.96)$$

siendo los valores de γ_k y $\bar{\gamma}_k$ en cada caso los siguientes:

- En el caso del método EBDF: $\gamma_k = \beta_k$ y $\bar{\gamma}_k = 0$.
- En el caso del método MEBDF: $\gamma_k = \hat{\beta}_k$ y $\bar{\gamma}_k = (\beta_k - \hat{\beta}_k)$.

Así, es la expresión (7.96) la que se utiliza en las funciones $calcR$ y $calcJ$ del objeto $supF$.

Objeto mebdf

- Datos: BG .
- Funciones: $inic(objode, objprob, objmalla)$.

El objeto $mebdf$ hereda de $ebdf$ los datos y funciones comunes y añade el dato $BG = (\hat{\beta}_k)$. En la inicialización adecúa la matriz $BB = (\beta_{k+1} \quad \hat{\beta}_k \quad (\beta_k - \hat{\beta}_k))$ que contiene los coeficientes que se utilizarán en el cálculo de \bar{R} .

Objeto mebdflinc

- Datos: BG .
- Funciones: $inic(objode, objprob, objmalla)$.

El objeto $mebdflinc$ consiste en aplicar el método MEBDF al caso lineal de coeficientes constantes (7.52). El objeto $bdflinc$ resuelve el problema lineal de coeficientes constantes mediante BDFs y lo utilizaremos en los pasos predictores. Sólo falta construir la parte correctora del método MEBDF. Aplicando la expresión (5.7) al sistema (7.52) se tiene:

$$\sum_{j=0}^k \alpha_j M y_{n+j} = h \hat{\beta}_k (A y_{n+k} + g(t_{n+k})) + h \beta_{k+1} (A \bar{y}_{n+k+1} + g(t_{n+k+1})) + h (\beta_k - \hat{\beta}_k) (A \bar{y}_{n+k} + g(t_{n+k})) \quad (7.97)$$

Agrupamos en la parte izquierda de la igualdad (7.97) los valores de y_{n+k} :

$$\begin{aligned} (\alpha_k M - h\hat{\beta}_k A) y_{n+k} = & - \sum_{j=0}^{k-1} \alpha_j M y_{n+j} + h\hat{\beta}_k g(t_{n+k}) + h\beta_{k+1} (A\bar{y}_{n+k+1} + g(t_{n+k+1})) \\ & + h(\beta_k - \hat{\beta}_k) (A\bar{y}_{n+k} + g(t_{n+k})) \quad (7.98) \end{aligned}$$

Como en casos anteriores, generalizamos la expresión (7.98) para cualquier orden utilizando las variables C_2 e Y_{n+k-1} :

$$\begin{aligned} (\alpha_k M - h\hat{\beta}_k A) y_{n+k} = & -C_2 Y_{n+k-1} + h\hat{\beta}_k g(t_{n+k}) + h\beta_{k+1} (A\bar{y}_{n+k+1} + g(t_{n+k+1})) \\ & + h(\beta_k - \hat{\beta}_k) (A\bar{y}_{n+k} + g(t_{n+k})) \quad (7.99) \end{aligned}$$

Y despejando y_{n+k} de (7.99) se tiene:

$$\begin{aligned} y_{n+k} = C_1 \left[-C_2 Y_{n+k-1} + h\hat{\beta}_k g(t_{n+k}) + h\beta_{k+1} (A\bar{y}_{n+k+1} + g(t_{n+k+1})) \right. \\ \left. + h(\beta_k - \hat{\beta}_k) (A\bar{y}_{n+k} + g(t_{n+k})) \right] \quad (7.100) \end{aligned}$$

siendo: $C_1 = (\alpha_k M - h\hat{\beta}_k A)^{-1}$, y C_2 y Y_{n+k-1} dados por (7.88).

El objeto *mebdf* hereda de *ebdf* los datos y funciones comunes y añade el dato $BG = (\hat{\beta}_k)$. Igual que la función *inic* del objeto *mebdf*, también el *inic* del objeto *mebdf* adecúa la matriz $BB = (\beta_{k+1} \quad \hat{\beta}_k \quad (\beta_k - \hat{\beta}_k))$.

7.3.5.6. El resto de métodos de las familias EBDF y MEBDF

En el Capítulo 5 hemos ampliado las familias EBDF y MEBDF:

- creando los métodos EBNDF, ENBDF y ENDF dentro de la familia EBDF y
- creando los métodos MEBNDF, MENBDF y MENDF dentro de la familia MEBDF.

Los cuatro métodos de cada familia comparten el mismo corrector. Si comparamos las dos familias entre sí, tenemos que: los métodos EBNDF y MEBNDF utilizan como primer predictor el método BDF y como segundo el método NDF; los métodos ENBDF y MENBDF primero predicen con el método NDF y después con el método BDF y los métodos ENDF y MENDF utilizan NDFs en sus dos predicciones. En cada una de estas parejas el esquema que se sigue es el mismo, siendo la única diferencia entre ellas el corrector.

Objetos no lineales de las familias EBDF y MEBDF

Los seis objetos no lineales creados (*ebndf*, *mebndf*, *enbdf*, *menbdf*, *endf* y *mendf*) tienen la siguiente estructura:

- Creamos los objetos *ebndf*, *enbdf* y *endf* los cuales tienen su inicialización (*inic*) y su paso (*paso*).
- A continuación creamos los tres objetos correspondientes a los tres métodos de la familia MEBDF (*mebndf*, *menbdf* y *mendf*) teniendo cada uno su inicialización (*inic*) y heredando el paso del método con iguales predictores de la familia EBDF.

Así, el objeto *mebndf* hereda el paso del objeto *ebndf*, el objeto *mendf* el del *endf*, etc. Se puede visualizar este esquema en la Figura 7.19.

Los objetos *ebndf* y *enbdf* tienen los mismos datos y funciones y son los siguientes:

- Datos: M , $ykm1$, $fbk1$, $fbk2$, $objbdf$, $objndf$.
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1, yk1] = paso(tk, yk, h, objode, varargin)$.

Los objetos *mebndf* y *menbdf* heredan respectivamente los datos y funciones de *ebndf* y *enbdf* y ambos añaden la propiedad BG y la función $inic(objode, objprob, objmalla)$. El objeto *endf* tiene los mismos datos que los objetos *ebndf* y *enbdf*, a excepción de la propiedad *objdf* ya que las dos predicciones las hace con el método NDF. Y el objeto *mendf* añade la propiedad BG y la función $inic(objode, objprob, objmalla)$ a los datos y funciones que hereda del objeto *endf*.

Objetos lineales de las familias EBDF y MEBDF

Igual que en el caso no lineal, también en el caso lineal tienen los mismos datos y funciones los objetos *ebndflinc* y *enbdfinc*:

- Datos: $C1$, $ykm1$, $gk1$, $objbdfinc$, $objndflinc$.
- Funciones: $inic(objode, objprob, objmalla)$,
 $[tk1, yk1] = paso(tk, yk, h, objode, objprob, objmalla)$.

El objeto *endflinc* no tiene la propiedad *objbdfinc* porque sólo predice utilizando NDFs. Los objetos *mebndflinc*, *menbdfinc* y *mendflinc* heredan de sus respectivos padres las propiedades y las funciones correspondientes, añadiendo únicamente la propiedad BG y la función de inicio.

7.3.6. Método odesuite

Este método sólo es una conexión con la *odesuite* de Matlab donde se proporcionan funciones diversas para la integración numérica de sistemas de EDOs, en particular la *ode45* y la *ode15s* representantes de los métodos explícitos e implícitos respectivamente.

Estas funciones sólo requieren del exterior la función de apoyo que proporciona el segundo miembro $f(t, y)$ de (7.38) o (7.44) del sistema de EDOs que se quiere resolver, ya que en los métodos implícitos implementan internamente ellas mismas el método de Newton para resolver el sistema no lineal de ecuaciones de cada paso.

7.3.6.1. Objeto odesuite

Esta clase hereda directamente de *base_met* haciendo uso de algunos de sus datos y funciones (por ejemplo, *calcf*(objfity)), y añadiendo otros específicos:

- Datos: *metJ*.
- Funciones: *inic*, *resolver*, *actpaso*.

Sólo reutiliza la función *calcf*(objfity) de los desarrollos mostrados anteriormente y deben sobrecargarse algunas funciones *inic*, *resolver* y *actpaso*, a fin de reflejar los aspectos específicos de la interfaz.

```
function options=inic(objjode,objprob,objmalla)
objfity=objjode.objfity; M=inicM(objfity,objjode,objprob,objmalla); tmetJ=objjode.metJ;
if ~isempty(metJ) %ode15s , etc.
    switch metJ
        case 'jder'
            options=odeset('Mass',M,'Jacobian',@Fjac,'stats','on','outputFcn',@actpaso);
            inic_varele(objprob,objmalla)
        case 'lincc' %ode lineal de coef ctes
            j=inicJ(objfity,objjode,objprob,objmalla);
            options= odeset('Mass',objjode.M,'Jacobian',j,'stats','on'); objjode.dat.j=j;
        end
    else % ode45, etc.
        options = odeset('Mass',M,'stats','on','outputFcn',@actpaso);
        inic_varele(objprob,objmalla)
        if strcmp(objjode.moda,'lincc')
            j=objfity.inicJ(objjode,objprob,objmalla); objjode.dat.j=j;
        end
    end
end
```

```
function resolver(objjode,objprob,objmalla)
interv=objjode.interv ; yini=objjode.yini; objfity=objjode.objfity; metodo=objjode.metodo;
fty=@objfity.calcf; options = inic(objjode,objprob,objmalla);
[tsol,ysol]=feval(metodo,fty,interv,yini,options,objjode,objprob,objmalla);
objjode.tsol=tsol; objjode.ysol=ysol;
actfin(objfity,tsol,ysol(end,:),'objjode,objprob,objmalla);
```

```
function status=actpaso(t,y,flag,objjode,varargin)
switch flag
    case 'init',
    case 'done'
    case []
        objfity=objjode.objfity; actpaso(t,y,objfity,objjode,varargin{:})
        end, status=0;
```

La función *inic* se encarga de definir la variable “options” donde se define la forma de trabajo de las funciones de la *odesuite*. En particular la propiedad ‘Mass’ donde se indica que el sistema se resolverá en el formato $My' = f(t, y)$, la propiedad ‘Jacobian’ donde se indica que se proporciona la matriz jacobiana (en caso contrario la *odesuite* la calcularía numéricamente por diferencias finitas) de dos posibles maneras: si es constante, calculándola una sola vez aquí e introduciéndola en la estructura “options”; y si es variable, indicando la función @Fjac que ha de calcular en cada paso. Igualmente en la tercera propiedad que se define, ‘outputFcn’, se especifica el nombre de la función (@actpaso) que la *odesuite* ejecuta al final de cada uno de sus pasos, lo cual es muy importante porque permite a la *odesuite* modificar cualquier entorno exterior de acuerdo con los resultados que va obteniendo, por ejemplo en nuestro caso, actualizando la geometría de la malla en un problema de grandes deformaciones. Por último, en la función *inic* se inicializa la estructura *dat* (dato del objeto base) añadiendo el campo *un* que servirá para disponer de una referencia de partida en cada paso del valor actual de las incógnitas nodales completas.

La función *resolver* no contiene el bucle en pasos como en los otros métodos, sino que simplemente se encarga mediante *feval* de ejecutar la función de la *odesuite* que ha sido especificada en la variable *tipomet* de la entrada de datos y que se encuentra en el dato método del objeto *odesuite*. Asimismo, le proporciona los argumentos adicionales: función de apoyo (*fty*), intervalo, condiciones iniciales, opciones y variables adicionales necesarias (*objode*, *objprob* y *objmalla*).

```
function yp= flin(t,y,objode)
yp=objode.dat.j*y;
```

La función *actpaso* acaba reutilizando la función *actpaso* de *base_fly*, pero lo hace en el formato especificado por la *odesuite*, que es el indicado en la función anterior.

7.3.7. Método alfa

En el método HHT- α (o método alfa) abordamos directamente la resolución del sistema de EDOs de orden 2, obviando su conversión a orden 1 y evitando la duplicación de las funciones incógnita. A partir de las condiciones iniciales (CI) integramos el sistema de EDOs obtenido de la semidiscretización de elementos finitos de un problema de elasticidad (o de Laplace tipo onda):

$$\begin{cases} MU'' + CU' + F_{int}(U) = F_{ext} \\ U(0) = U_0 = d_0 \\ U'(0) = U'_0 = v_0 \end{cases} \quad (7.101)$$

Queremos aproximar calculando paso a paso los valores $\{d_n \equiv U_n, n = 0, 1, \dots\}$ y las derivadas $v_n \equiv d'_n$, $a_n \equiv d''_n$. Para ello, puesto que ya hemos obtenido los valores en el

paso n , obtendremos los valores en el paso $(n + 1)$ resolviendo el sistema de ecuaciones triparamétrico (α, β, γ) (3.23) y considerando $\alpha_m = 0$ (ver Sección 3.2.2):

$$\begin{cases} Ma_{n+1} + Cv_{n+1-\alpha} + F_{int}(d_{n+1-\alpha}) = F_{ext}(t_{n+1-\alpha}) \\ d_{n+1} = d_n + hv_n + \frac{h^2}{2} [(1 - 2\beta)a_n + 2\beta a_{n+1}] \\ v_{n+1} = v_n + h [(1 - \gamma)a_n + \gamma a_{n+1}] \end{cases} \quad (7.102)$$

siendo $h = \Delta t$.

En la primera relación, la de equilibrio dinámico (7.101) planteada en $(n + 1)$, se definen los valores intermedios:

$$\begin{cases} t_{n+1-\alpha} = (1 - \alpha)t_{n+1} + \alpha t_n = t_{n+1} - \alpha h \\ d_{n+1-\alpha} = (1 - \alpha)d_{n+1} + \alpha d_n \\ v_{n+1-\alpha} = (1 - \alpha)v_{n+1} + \alpha v_n \end{cases} \quad (7.103)$$

Reordenamos las dos últimas fórmulas del algoritmo (7.102) en la siguiente forma predictora-correctora:

- Predictores (conocidos a_n, v_n, d_n):

$$\begin{cases} \tilde{d}_{n+1} = d_n + hv_n + \frac{h^2}{2}(1 - 2\beta)a_n \\ \tilde{v}_{n+1} = v_n + h(1 - \gamma)a_n \end{cases} \quad (7.104)$$

- Correctores (en función del valor desconocido a_{n+1}):

$$\begin{cases} d_{n+1} = \tilde{d}_{n+1} + h^2\beta a_{n+1} \\ v_{n+1} = \tilde{v}_{n+1} + h\gamma a_{n+1} \end{cases} \quad (7.105)$$

Sustituyendo (7.105) en la relación de equilibrio de (7.102), el cálculo de a_{n+1} es directo en el caso lineal, donde se verifica $F_{int}(d_{n+1-\alpha}) = Kd_{n+1-\alpha}$, obteniéndose:

$$\begin{aligned} (M + (1 - \alpha)Ch\gamma + (1 - \alpha)Kh^2\beta)a_{n+1} = \\ = F_{ext}(t_{n+1-\alpha}) - C((1 - \alpha)\tilde{v}_{n+1} + \alpha v_n) - K((1 - \alpha)\tilde{d}_{n+1} + \alpha d_n) \end{aligned} \quad (7.106)$$

Que se puede resolver paso a paso $(n = 0, 1, \dots)$, a partir del valor inicial a_0 que verifica:

$$Ma_0 = F_0 - Cv_0 - Kd_0 \quad (7.107)$$

donde F_0, v_0, d_0 son los valores iniciales de la acción exterior.

En el caso general no lineal, la condición de equilibrio (7.102) es un sistema no lineal de ecuaciones donde la incógnita es d_{n+1} y el residuo para una prueba \hat{d}_{n+1} es:

$$\begin{aligned}\bar{R}(\hat{d}_{n+1}, t_{n+1}) &= \underbrace{F_{ext}(t_{n+1-\alpha}) - F_{int}(\hat{d}_{n+1-\alpha})}_{R(\hat{d}_{n+1-\alpha})} - M\hat{a}_{n+1} - C\hat{v}_{n+1-\alpha} \\ &= R(\hat{d}_{n+1-\alpha}) - M\hat{a}_{n+1} - C\hat{v}_{n+1-\alpha}\end{aligned}\quad (7.108)$$

Dada una prueba \hat{d}_{n+1} , para pasar del paso n a $(n+1)$ mediante el método de Newton calcularemos:

1. Los predictores según (7.104).
2. La aceleración de prueba \hat{a}_{n+1} a partir de la corrección de los desplazamientos:

$$\hat{a}_{n+1} = \frac{\hat{d}_{n+1} - \tilde{d}_{n+1}}{h^2\beta} \quad (7.109)$$

la velocidad de prueba \hat{v}_{n+1} a partir de la corrección de las velocidades:

$$\hat{v}_{n+1} = \tilde{v}_{n+1} + h\gamma\hat{a}_{n+1} \quad (7.110)$$

los desplazamientos y velocidades intermedios:

$$\begin{cases} \hat{d}_{n+1-\alpha} = (1-\alpha)\hat{d}_{n+1} + \alpha\hat{d}_n \\ \hat{v}_{n+1-\alpha} = (1-\alpha)\hat{v}_{n+1} + \alpha\hat{v}_n \end{cases} \quad (7.111)$$

3. El vector residuo utilizando (7.108).
4. La matriz jacobiana de (7.108):

$$\bar{j} = \frac{\partial \bar{R}(\hat{d}_{n+1}, t_{n+1})}{\partial \hat{d}_{n+1}} = \frac{\partial R(\hat{d}_{n+1-\alpha}, t_{n+1})}{\partial \hat{d}_{n+1}} - M \frac{\partial \hat{a}_{n+1}}{\partial \hat{d}_{n+1}} - C \frac{\partial \hat{v}_{n+1-\alpha}}{\partial \hat{d}_{n+1}} \quad (7.112)$$

- En el cálculo de (7.112) hay que tener en cuenta:

$$\frac{\partial R(\hat{d}_{n+1-\alpha})}{\partial \hat{d}_{n+1}} = \frac{\partial R(\hat{d}_{n+1-\alpha})}{\partial \hat{d}_{n+1-\alpha}} \cdot \frac{\partial \hat{d}_{n+1-\alpha}}{\partial \hat{d}_{n+1}} = \frac{\partial R(\hat{d}_{n+1-\alpha})}{\partial \hat{d}_{n+1-\alpha}} (1-\alpha) \quad (7.113)$$

- También hay que tener en cuenta que de (7.109) se tiene:

$$\frac{\partial \hat{a}_{n+1}}{\partial \hat{d}_{n+1}} = \frac{1}{h^2\beta} I \quad (7.114)$$

- Y que de (7.111), (7.110) y (7.109) resulta:

$$\frac{\partial \hat{v}_{n+1-\alpha}}{\partial \hat{d}_{n+1}} = \frac{\partial \hat{v}_{n+1-\alpha}}{\partial \hat{v}_{n+1}} \cdot \frac{\partial \hat{v}_{n+1}}{\partial \hat{a}_{n+1}} \cdot \frac{\partial \hat{a}_{n+1}}{\partial \hat{d}_{n+1}} = (1-\alpha) Ih \gamma I \frac{1}{h^2 \beta} I \quad (7.115)$$

5. Las pruebas actualizadas de los desplazamientos:

$$\hat{d}_{n+1}^{m+1} = \hat{d}_{n+1}^m - \left[\bar{J}(\hat{d}_{n+1}^m, t_{n+1}) \right]^{-1} \bar{R}(\hat{d}_{n+1}^m, t_{n+1}) \quad (7.116)$$

repetiendo (iterando) los puntos 1 – 5 hasta que $\left\| \bar{R}(\hat{d}_{n+1}^{m+1}, t_{n+1}) \right\| \leq tol \approx 0$.

6. Los desplazamientos convergidos: $d_{n+1} \equiv \hat{d}_{n+1}$ sirven para iniciar el siguiente paso. Todo ello queda implementado en el objeto alfa.

7.3.7.1. Objeto alfa

- Datos: *alfa*, *beta*, *gamma*, *M*, *K*, *dini*, *vini*, *vn*, *an*.
- Funciones: *inic*(objode, objprob, objmalla),
 $[tn1, yn1] = \textit{paso}(tn, yn, h, \textit{objode}, \textit{objprob}, \textit{objmalla})$,
 $[R, Fext, Reac, Fint] = \textit{calcR}(u, \textit{objode}, \textit{objprob}, \textit{objmalla}_n)$,
 $Jb = \textit{calcJ}(u, \textit{objode}, \textit{objprob}, \textit{objmalla}, \textit{objJ})$.

```
function [tn1,yn1] = paso(tn,yn,h,objode,objprob,objmalla)
beta=objode.beta; gamma=objode.gamma; tn1=tn+h;
dn=yn(1:end/2); vn=objode.vn; an=objode.an;
objode.vnp=vn+h*(1-gamma)*an; objode.inc_dnp=h*vn+(.5)*(h^2)*(1-2*beta)*an;
resolver(objode.objnr,objode,objprob,objmalla); uinc_n=objode.objnr.u;
an1=(uinc_n-inc_dnp)/(beta*h*h); dn1=dn+uinc_n; vn1=vnp+h*gamma*an1;
objode.vn=vn1; objode.an=an1;
yn1=[dn1;vn1]; actpaso(tn1,yn1,objode,objprob,objmalla);
```

```
function Rb=calcR(u,objode,objprob,objmalla_n)
alfa=objode.Alf; beta=objode.beta; gamma=objode.gamma; M=objode.M; C=objode.C;
inc_dnp=objode.dat.inc_dnp; h=objode.h; vn=objode.vn; vnp=objode.dat.vnp;
inc_u=u; inc_u_alfa=(1-alfa)*inc_u; an1=(inc_u-inc_dnp)/(beta*h*h);
vn1=vnp+h*gamma*an1;
R=calcR(objprob,objmalla_n,inc_u_alfa); Rb=R-(M*an1+(1-alfa)*C*vn1+alfa*C*vn);
```

```
function Jb=calcJ(u,objode,objprob,malla_n1)
J=calcJ(objprob,malla_n1); %La parte del problema
[Jm,Jc]=calcJmet(u,objode); %La parte del método
Jb=(1-objode.Alf)*J+Jm+Jc;
```

```
function [Jm,Jc]=calcJmet(u,objode)
alfa=objode.alfa; beta=objode.beta; gamma=objode.gamma;
h=objode.h; M=objode.M; C=objode.C;
Jm=-M/(h*h*beta); Jc=-(1-alfa)*gamma*C/(h*beta);
```

7.3.7.2. Objeto alfalinc

Este objeto implementa la particularización de las fórmulas del método alfa en el caso lineal donde las fórmulas de avance quedan en la forma explícita indicada en (7.106) que es mucho menos costosa ya que no deben iterar. Los datos son los mismos que los del objeto alfa, a excepción de un par que quedan definidos como estructura en el dato *dat* del objeto base. Además de las funciones comunes que residen en la clase padre *base_met*, esta clase implementa los métodos específicos *inic* y *paso*, que reflejan los aspectos específicos del algoritmo (7.104), (7.105), (7.106) y que se describen a continuación:

```
function inic(objode,objprob,objmalla)
K=rigidez(objprob,objmalla); tipomas=objprob.tipomas;
M=calcmasa(objprob,objmalla,tipomas); [ro,d]=get(objmalla.objmat,'ro','d');
C=M*(d/ro); objode.C=C; objode.K=K;
alfa=objode.Alfa; beta=objode.beta; gamma=objode.gamma; h=objode.h;
MCKinv=(M+(1-alfa)*(gamma*h*C+(beta*(h^2))*K))\speye(objprob.ngdla);
Fext=calfext(objprob,objmalla); Fint=objode.K*objode.dini;
R=Fext-Fint; objode.aini=M\ (R-C*objode.vini);
objode.vn1=objode.vini; objode.an1=objode.aini; objode.MCKinv=MCKinv;
```

```
function [tn1,yn1] = paso(tn,yn,h,objode,objprob,objmalla)
alfa=objode.alfa; beta=objode.beta; gamma=objode.gamma;
tn1=tn+h; tn1_a=tn-alfa*h; dn=yn(1:end/2); vn=objode.vn1; an=objode.an1;
vnp=vn+h*(1-gamma)*an; inc_dnp=h*vn+(.5)*(h^2)*(1-2*beta)*an;
uc=solc(objprob,dn); set(objmalla,'u',uc); dnp=dn+inc_dnp;
Fext=calfext(objprob,objmalla,tn1_a);
Fn1=Fext-objode.C*((1-alfa)*vnp+alfa*vn)-objode.K*((1-alfa)*dnp+alfa*dn);
an1=objode.MCKinv*Fn1; dn1=dn+inc_dnp+(h^2)*beta*an1; vn1=vnp+h*gamma*an1;
objode.vn1=vn1; objode.an1=an1; yn1=[dn1;vn1];
```

7.3.8. Conclusiones

Lo característico de esta sección es la organización de un proceso paso a paso incremental, con posibles iteraciones no lineales por paso, donde es posible reutilizar todo lo anterior, cuidando especialmente las actualizaciones entre pasos una vez concluida la posible resolución iterativa no lineal. Todo ello queda gestionado, manteniendo la arquitectura anterior y añadiendo nuevos objetos método que implementan distintas fórmulas o algoritmos de avance dinámico (integración temporal) por paso, incluyendo el cuasi-

estático no lineal cuya fórmula de avance es trivial pero que se beneficia de los procesos comunes de actualización entre pasos.

Cabe señalar el diseño bajo la misma arquitectura de cinco tipos de métodos: los explícitos de paso constante (el método RK4); los implícitos adaptados a problemas lineales, donde se distinguen los que son para EDOs de orden 1 (entre ellos tenemos los métodos unipaso y los métodos multipaso) y los específicos para EDOs de orden 2 (método HHT- α) y los implícitos no lineales donde se distinguen los mismos casos que en los implícitos lineales. Mención aparte merece la clase *odesuite*, que no es más que un enlace entre la arquitectura orientada a objeto aquí propuesta y los métodos de la *odesuite* de Matlab. En total, los distintos tipos de métodos que tenemos son 21: 1 cuasi, 1 RK4, 1 trapezoidal, 11 multipasos (1 BDF- α , 1 BDF, 1 NDF, 4 en la familia EBDF, 4 en la familia MEBDF), las 6 funciones de la *odesuite* de Matlab, 1 HHT- α . Es decir, 20 para resolver EDOs de orden 1 y 1 para resolver EDOs de orden 2. Estas opciones se multiplican, ya que los métodos BDF, NDF y las familias EBDF y MEBDF han sido implementadas para distintos órdenes de precisión.

Contabilizando a grosso modo los posibles experimentos distintos de dinámica que se pueden realizar y sin tener en cuenta el orden de precisión al que algunos métodos pueden trabajar:

- Tenemos 192 experimentos posibles: 24 (casos lineales) + 168 (casos no lineales). Todos estos experimentos resultan en EDOs de orden 1 o EDOs de orden 2 que se podrán reducir a EDOs de orden 1.
- Teniendo en cuenta sólo los casos cuasi-estático y los métodos dinámicos para resolver EDOs de orden 1 (que en total son 20), salen: $192 \times 20 = 3840$ casos.
- A estos 3840 casos se le tendrían que sumar: la posibilidad que presentan las EDOs de orden 2 para resolverlas directamente con métodos para EDOs de orden 2 (HHT- α en nuestro caso), y las posibilidades que presentan los métodos BDF, NDF y las familias EBDF y MEBDF para resolver en distinto orden de precisión.

La Figura 7.21 muestra dos posibles casos entre todos los posibles: el de la izquierda corresponde a un problema de onda 1D lineal con interpolación elemental lineal. El de la derecha corresponde a un problema elástico 2D con material neohookeano y grandes deformaciones, con cálculo exacto de la matriz jacobiana e interpolación bilineal en cuadriláteros.

7.4. Resultados

7.4.1. Ejemplo 1: Ecuación de onda lineal

Mediante la metodología descrita en las líneas anteriores experimentaremos en este ejemplo con el control paramétrico del amortiguamiento algorítmico de los métodos

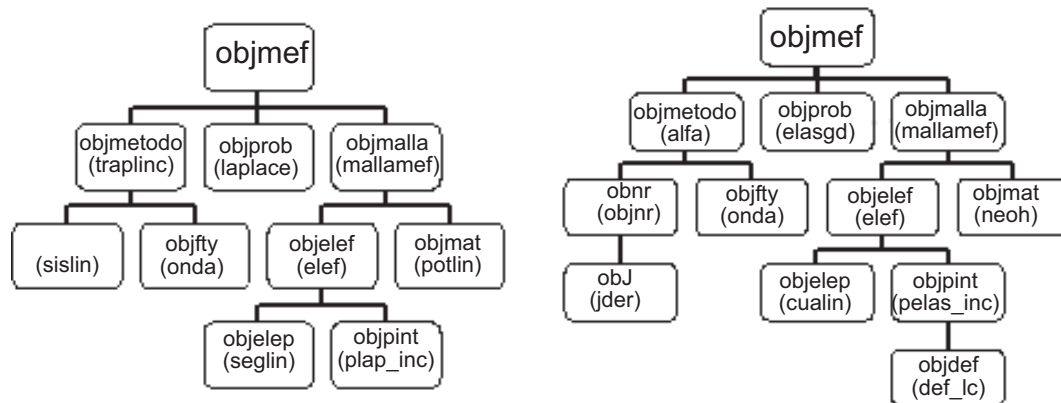


Figura 7.21: Dos posibles casos de la arquitectura elemento finito (caso dinámico).

HHT- α y BDF- α , en el caso de la ecuación de onda con un pulso rectangular. Consideramos la ecuación de onda 1D (1.23) con CI correspondientes al pulso rectangular y CC dadas por (1.23), en una cuerda de 8 unidades de longitud $L = 8$ y $\alpha^2 = 1$. En el Capítulo 1 hemos estudiado este ejemplo para una discretización de 400 elementos, en un intervalo de 16 segundos y dando 1400 pasos y es con estos datos con los que vamos a seguir trabajando. Se trata de un problema con 399 frecuencias naturales que son los autovalores de la matriz jacobiana del sistema de EDOs. Vimos que el método HHT- α con $\alpha = 0,3$ elimina las altas frecuencias (Figura 1.32) y que el método trapezoidal y la función *ode15s* son capaces de integrar con precisión el sistema de EDOs proveniente de la semidiscretización MEF, cuya solución exacta muestra “rizo” debido a que los modos altos discretos difieren sensiblemente de los modos continuos.

Si experimentamos con el método HHT- $\alpha = 0,05$, el resultado que se obtiene elimina menos las altas frecuencias, Figura 7.22, que cuando elegimos $\alpha = 0,3$, Figura 1.32. Sobre la representación gráfica de la evolución de los radios espectrales en función de la frecuencia de métodos diversos, superponemos los radios espectrales correspondientes a las frecuencias propias de vibración de nuestro problema al resolverlo con los métodos HHT- $\alpha = 0,05$ y HHT- $\alpha = 0,3$ y 1400 pasos (ver imagen izquierda de la Figura 7.23). Observamos que los radios espectrales son mayores en el caso del método HHT- $\alpha = 0,05$ y es esta la razón por la que este método amortigua menos que el método HHT- $\alpha = 0,3$ las frecuencias altas. Recordemos que en las figuras de radios espectrales denotamos $\Omega = h\omega$, siendo ω las frecuencias propias de vibración del sistema de EDOs.

También el resultado que se obtiene con el método HHT- $\alpha = 0,3$ pero dando 2000 pasos, Figura 7.24, elimina menos las frecuencias altas que cuando se dan 1400 pasos. Esto es debido a que de nuevo, los radios espectrales del método HHT- $\alpha = 0,3$ y 2000 pasos (ver imagen derecha de la Figura 7.23) son mayores que los que corresponden a dar 1400 pasos (imagen izquierda de la Figura 7.23).

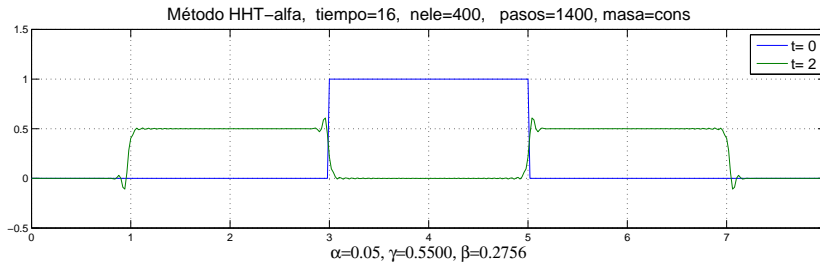


Figura 7.22: Solución de la ec. de onda CI pulso (HHT- $\alpha = 0,05$ 400 elem. 1400 pasos).

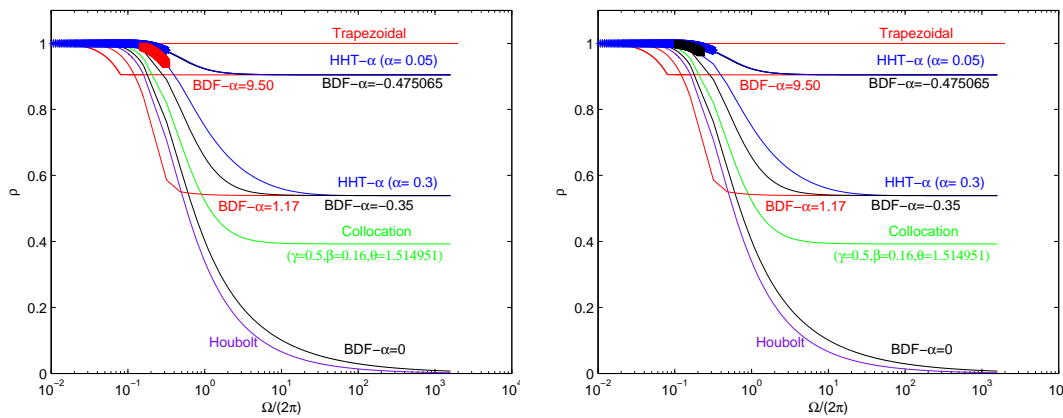


Figura 7.23: Radios espectrales del problema (1.23), método HHT- α .

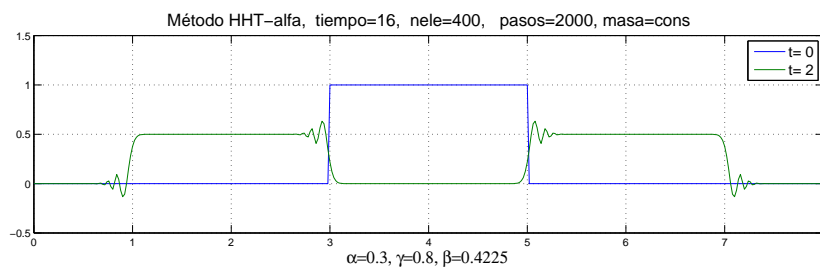


Figura 7.24: Solución de la ec. de onda CI pulso (HHT- $\alpha = 0,3$ 400 elem. 2000 pasos).

En el Capítulo 6 presentamos el método BDF- α y dibujamos su radio espectral para algunos valores del parámetro α , Figura 6.6. Comprobamos que la pareja de métodos

HHT- $\alpha = 0,05$ y BDF- $\alpha = -0,475065$ tiene parecido radio espectral según aumenta el valor de la frecuencia y el mismo ρ_∞ . En la Figura 7.25 se puede ver que el resultado obtenido con el BDF- $\alpha = -0,475065$ es muy similar al que se obtiene con su pareja HHT- $\alpha = 0,05$, Figura 7.22. ¿Sucederá lo mismo con la pareja HHT- $\alpha = 0,3$ y BDF- $\alpha = -0,35$? ¿Será capaz el método BDF- α de conseguir el resultado que nos proporciona el método de Hilber-Hughes-Taylor?

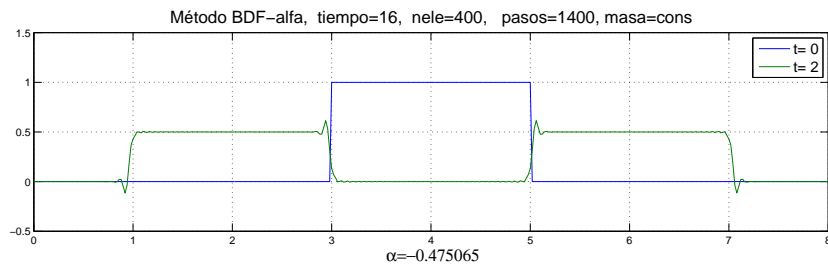


Figura 7.25: Solución ec. onda CI pulso (BDF- $\alpha = -0,475065$ 400 elem. 1400 pasos).

En la Figura 7.26 hemos representado el resultado que se obtiene con el método BDF- $\alpha = -0,35$ y efectivamente es capaz de eliminar las altas frecuencias de manera similar a su pareja HHT- $\alpha = 0,3$. En la imagen izquierda de la Figura 7.27 se han dibujado los radios espectrales del método BDF- α para las frecuencias de este problema.

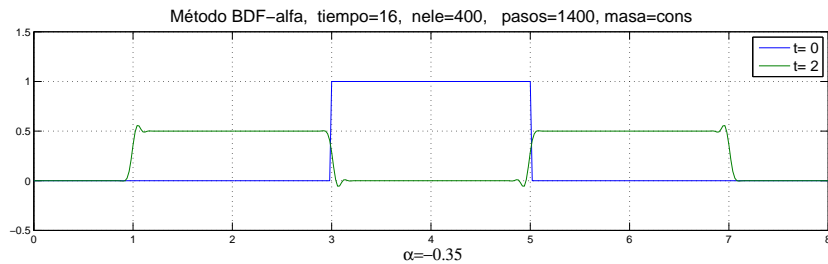


Figura 7.26: Solución ec. onda CI pulso (BDF- $\alpha = -0,35$ 400 elem. 1400 pasos).

Entre los métodos BDF- α corresponden a $\alpha = -0,5$ (método trapezoidal) y $\alpha = 0$ (método BDF2) los dos valores extremos de ρ_∞ . En las Figuras 7.28 y 7.29 hemos dibujado los resultados que se obtienen con ambos métodos. Observamos que el método trapezoidal proporciona un resultado con mucho “rizo” y que el método BDF2 consigue

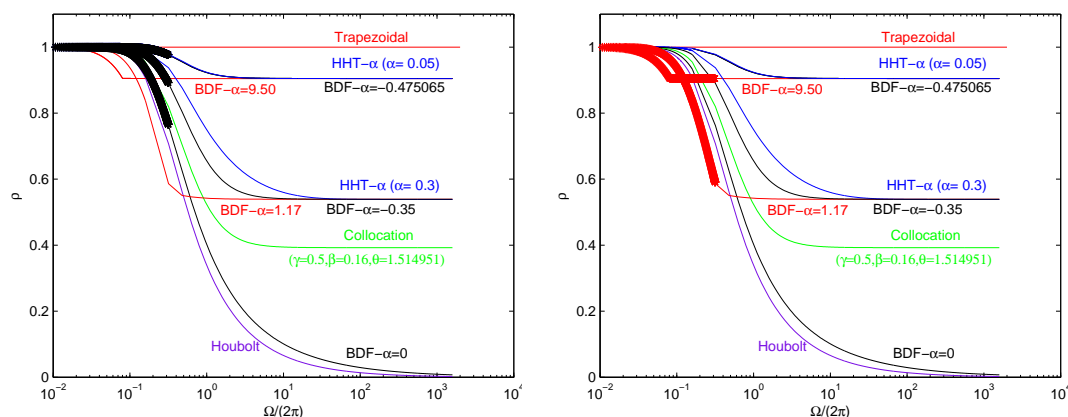


Figura 7.27: Radios espectrales del problema (1.23), método BDF- α .

un resultado “suave” a excepción de en las esquinas, ya que no es capaz de reproducirlas adecuadamente al haber eliminado no sólo las frecuencias altas sino también las intermedias.

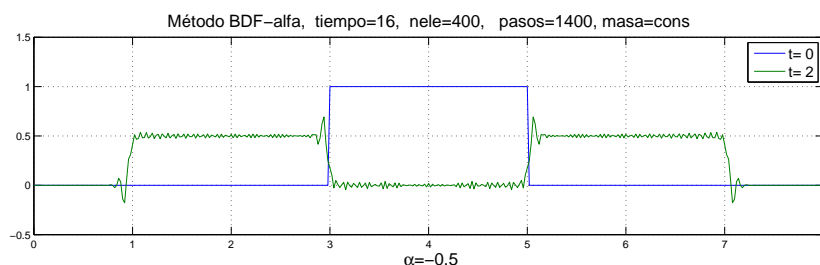


Figura 7.28: Solución de la ec. de onda CI pulso (BDF- $\alpha = -0,5$ 400 elem. 1400 pasos).

También hemos analizado lo que sucede con los métodos BDF- $\alpha = 9,50$ y BDF- $\alpha = 1,17$, con similares radios espectrales que los métodos BDF- $\alpha = -0,475065$ y BDF- $\alpha = -0,35$ en frecuencias altas. Debido a que las primeras también eliminan las frecuencias intermedias, los resultados que se obtienen con éstas pierden precisión, Figura 7.30. Los valores de los radios espectrales de éstas se han representado en la imagen derecha de la Figura 7.27.

Finalmente, vemos que los métodos EBDF con $k = 1$ y $k = 3$ (órdenes $p = 2$ y $p = 4$ respectivamente) eliminan las altas frecuencias y también las intermedias, y esto hace que el resultado pierda precisión en las esquinas, Figura 7.31. Se han representado en la

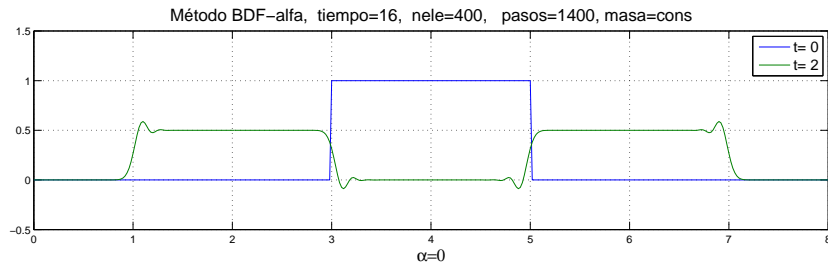


Figura 7.29: Solución de la ec. de onda CI pulso (BDF- $\alpha = 0$ 400 elem. 1400 pasos).

Figura 7.32 los radios espectrales correspondientes a las frecuencias propias de vibración de nuestro problema al resolverlo con los métodos EBDF1 y EBDF3.

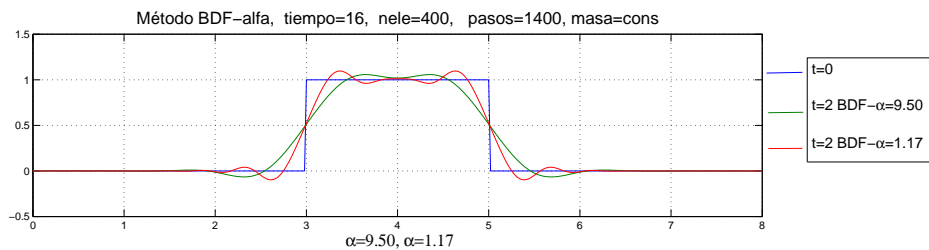


Figura 7.30: Solución ec. onda CI pulso (BDF- $\alpha = 9,51$, $\alpha = 1,17$ 400 elem. 1400 pasos)

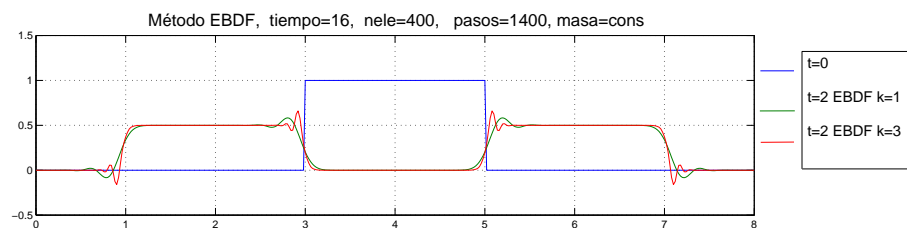


Figura 7.31: Solución de la ec. de onda CI pulso (EBDF 400 elem. 1400 pasos).

Cuando resolvemos (1.23) utilizando los nuevos métodos ENDF y MENDF con $k = 3$ (orden $p = 4$), conseguimos soluciones parecidas que con el método EBDF. En la Figura 7.33 se han representado las soluciones que se obtienen con 1400 pasos y con 64 pasos. Aunque no hayamos dibujado los radios espectrales de estos nuevos métodos, el resultado que se obtiene con ellos nos hace pensar que éstos serán parecidos a los del método EBDF.

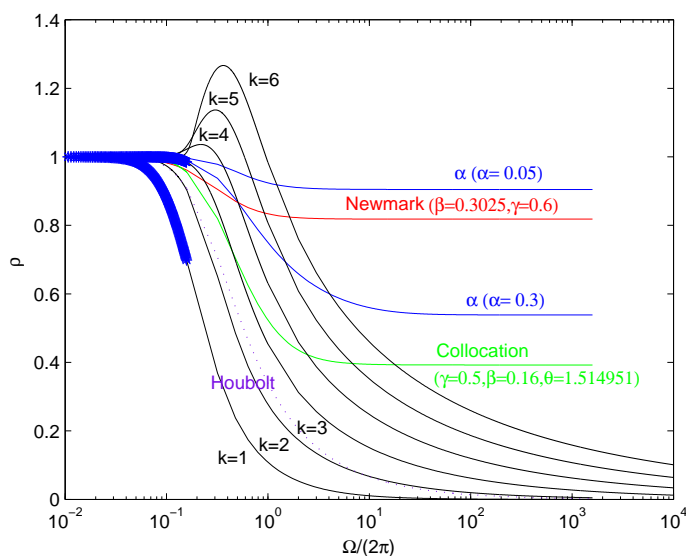


Figura 7.32: Radios espectrales del problema (1.23), método EBF.

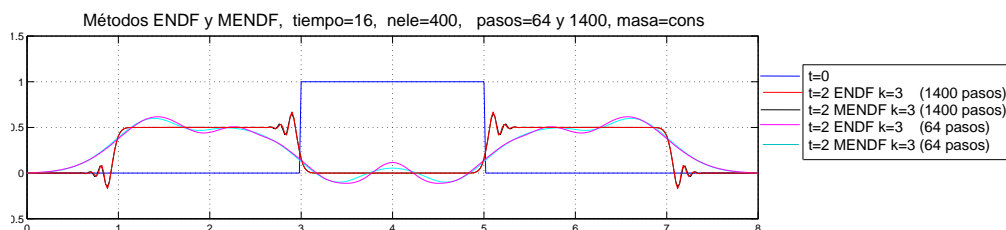


Figura 7.33: Solución de la ec. de onda CI pulso (ENDF y MENDF 400 elem.).

En la Tabla 7.1 se han recogido los tiempos computacionales (en segundos) de los métodos HHT- α y BDF- α al resolver el problema (1.23) con una discretización de 400 elementos y dando 1400 pasos (utilizando la misma máquina que en el Capítulo 1). Se han agrupado las parejas de métodos HHT- α y BDF- α con iguales radios espectrales ρ_∞ . Se observa que el método HHT- α requiere del orden de 3 veces menos tiempo computacional que el método BDF- α . En el caso del método HHT- α la operación con más coste computacional es el producto que hay que hacer en (7.106) para despejar el valor de a_{n+1} :

$$a_{n+1} = MCK^{-1} \cdot F_{n+1} \quad (7.117)$$

siendo:

$$\begin{cases} MCK = (M + (1 - \alpha)Ch\gamma + (1 - \alpha)Kh^2\beta) \\ F_{n+1} = F_{ext}(t_{n+1-\alpha}) - C((1 - \alpha)\tilde{v}_{n+1} + \alpha v_n) - K((1 - \alpha)\tilde{d}_{n+1} + \alpha d_n) \end{cases} \quad (7.118)$$

Y en el caso del método BDF- α el producto que se realiza en la expresión (7.84) es el que más coste computacional supone:

$$y_{n+2} = C_1 \cdot \underbrace{[-C_2 Y_{n+1} - h\alpha A y_{n+1} + h(1 + \alpha)g(t_{n+2}) - h\alpha g(t_{n+1})]}_{TI} \Rightarrow y_{n+2} = C_1 \cdot TI \quad (7.119)$$

donde $C_1 = (\beta_2 M - h(1 + \alpha)A)^{-1}$, $C_2 = (\beta_1 M \quad \beta_0 M)$, $Y_{n+1} = (y_{n+1} \quad y_n)^T$.

Para comprender la diferencia que existe en tiempos, se debe tener en cuenta que la dimensión de la matriz $(\beta_2 M - h(1 + \alpha)A)$ es el doble que el de la matriz MCK . Aunque ambas matrices son dispersas, las inversas de estas matrices, $C_1 = (\beta_2 M - h(1 + \alpha)A)^{-1}$ y MCK^{-1} , no lo son y esto hace que en el producto (7.119) se realicen del orden de 4 veces más operaciones que en el producto (7.117).

HHT	BDF	HHT	BDF	HHT	BDF
$\alpha = 0$	$\alpha = -0,5$	$\alpha = 0,05$	$\alpha = -0,475065$	$\alpha = 0,3$	$\alpha = -0,35$
1,61	5,18	1,74	5,54	1,88	6,17

Tabla 7.1: Tiempo computacional al resolver (1.23) 400 elem., 1400 pasos.

También hemos comparado los tiempos computacionales empleados por el método BDF- $\alpha = 0$ (BDF2) y los métodos EBDF, ENDF y MENDF con $k = 3$ (orden $p = 4$) al resolver este problema, ver Tabla 7.2. El tiempo computacional de los tres últimos métodos es parecido y 3 veces mayor que el que requiere el método BDF2. La relación entre los tiempos computacionales de los métodos EBDF, ENDF y MENDF es lógica, ya que la cantidad de operaciones y la complejidad de éstas es parecida en los tres métodos. También resulta lógico que éstos requieran del orden de tres veces más tiempo que el BDF2, ya que en cada paso realizan tres *operaciones* por cada una que realiza el método BDF2. En todos ellos resultan los productos (7.64), (7.76), (7.92) y (7.100) los que más coste computacional suponen.

BDF	EBDF	ENDF	MENDF
$\alpha = 0$	$k = 3$	$k = 3$	$k = 3$
5,88	17,68	17,06	18,24

Tabla 7.2: Tiempo computacional al resolver (1.23) 400 elem., 1400 pasos.

7.4.2. Ejemplo 2: Una versión no lineal de la ecuación de onda

Consideramos el ejemplo de la cuerda de guitarra tratado en la Sección 1.2.3:

$$\rho u_{tt}(x, t) = \left(T + E \cdot S \left(\sqrt{1 + u_x^2(x, t)} - 1 \right) \right) u_{xx}(x, t)$$

En el Capítulo 1, hemos visto que con una discretización de 20 elementos y dando 1000 pasos constantes, el método HHT- $\alpha = 0,3$ proporciona resultados aceptables y que es capaz de eliminar la aportación de las altas frecuencias en la solución.

Hemos resuelto este ejemplo utilizando el método BDF- α , y lo hemos comparado con los resultados que se obtienen al utilizar el método HHT- α . Empezamos resolviendo este problema con el método BDF- $\alpha = 0$ que es el método BDF2. En la Figura 7.34 se muestra el resultado obtenido y se comprueba que efectivamente este método amortigua las altas frecuencias.

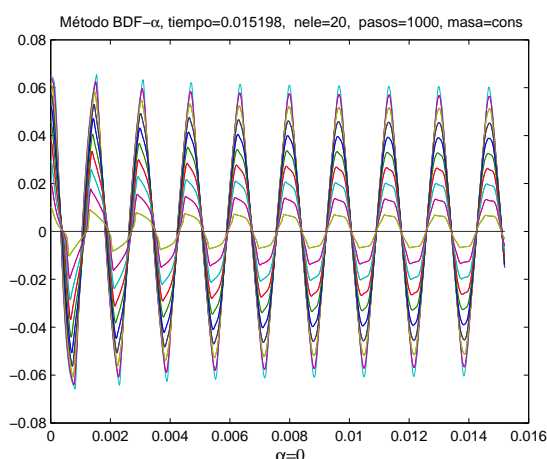


Figura 7.34: Solución ec. onda no lineal (método BDF- $\alpha = 0$).

A continuación resolvemos el problema utilizando los métodos HHT- $\alpha = 0,05$ y BDF- $\alpha = -0,475065$, con similar radio espectral ρ_∞ (ver Figura 6.6). En la Figura 7.35 se puede observar que ambos métodos proporcionan resultados similares.

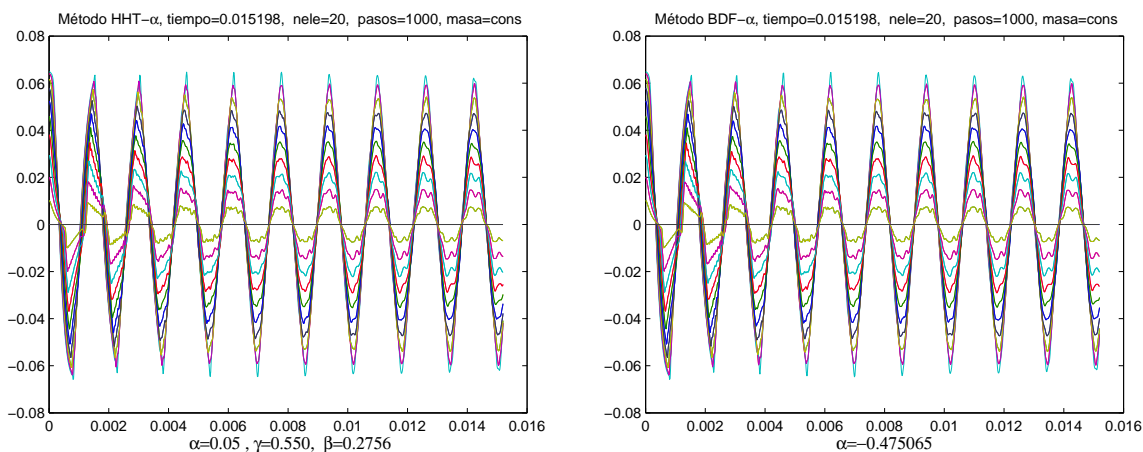


Figura 7.35: Solución ec. onda no lineal (HHT- $\alpha = 0,05$ y BDF- $\alpha = -0,475065$).

En el Capítulo 6 también hemos comprobado que los métodos $\text{BDF-}\alpha = 1,17$, $\text{BDF-}\alpha = -0,35$ y $\text{HHT-}\alpha = 0,3$ tienen similar radio espectral ρ_∞ (ver Figura 6.6). Se pueden ver en la Figura 7.36 los resultados que se obtienen para este problema con los métodos $\text{BDF-}\alpha = 1,17$ y $\text{BDF-}\alpha = -0,35$. El resultado obtenido con el método $\text{BDF-}\alpha = -0,35$ es muy similar al que se obtiene con el método $\text{HHT-}\alpha = 0,3$ (Figura 1.42). El método $\text{BDF-}\alpha = 1,17$ proporciona un resultado que va perdiendo en amplitud y en el que aparecen menos ciclos. Esto se debe a que el método $\text{BDF-}\alpha = 1,17$ actúa más que los otros dos métodos en las frecuencias bajas (eliminando su aportación en el resultado) y a que su error relativo en el periodo es mayor (ver Figuras 6.6 y 6.9).

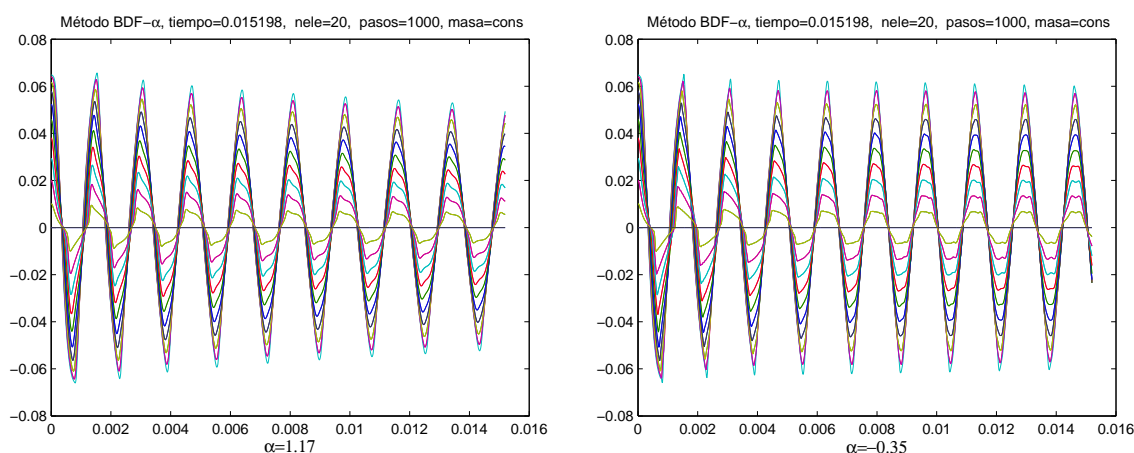


Figura 7.36: Solución ec. onda no lineal (métodos $\text{BDF-}\alpha = 1,17$ y $\text{BDF-}\alpha = -0,35$).

Comparando más en detalle los resultados que se obtienen con los métodos $\text{BDF-}\alpha = -0,35$ y $\text{HHT-}\alpha = 0,3$, se observa que el resultado que proporciona el segundo de ellos, al principio presenta más rizo que el que proporciona el primer método, ver Figura 7.37. Este rizo se hace más liso hacia el final, pareciéndose aún más los resultados que se obtienen con ambos métodos, Figura 7.38. Esto se debe a que el método $\text{BDF-}\alpha = -0,35$ amortigua más que el método $\text{HHT-}\alpha = 0,3$ las frecuencias intermedias (ver Figura 6.6).

En este caso, los tiempos computacionales requeridos por ambos métodos son parecidos: el método $\text{HHT-}\alpha = 0,3$ necesita 244,76 segundos y el método $\text{BDF-}\alpha = -0,35$ 259,89 segundos. También es parecido el número total de iteraciones que da cada método: 9479 iteraciones del método $\text{HHT-}\alpha = 0,3$, frente a 9516 del método $\text{BDF-}\alpha = -0,35$.

Cabe preguntarse la causa por la que ahora los tiempos son similares cuando antes en el caso lineal había un factor diferencial de 4 debido al dimensionamiento doble del método $\text{BDF-}\alpha$ frente al método $\text{HHT-}\alpha$. La respuesta está en que ahora la mayor parte del tiempo se emplea en el cálculo de las matrices \bar{R} y \bar{J} (*calcR* y *calcJ*) del sistema de ecuaciones a resolver en cada iteración: alrededor del %40 y %59 del tiempo total respectivamente. La construcción es muy similar en ambos casos, puesto que se calculan el mismo vector residuo y la misma matriz de rigidez tangente.

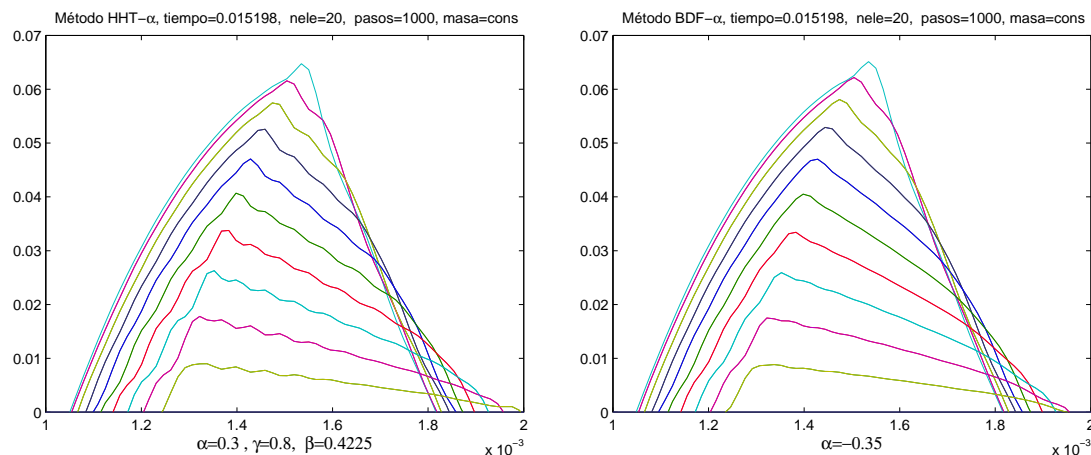


Figura 7.37: Detalle de las Figuras correspondientes a $\text{HHT-}\alpha = 0,3$ y $\text{BDF-}\alpha = -0,35$.

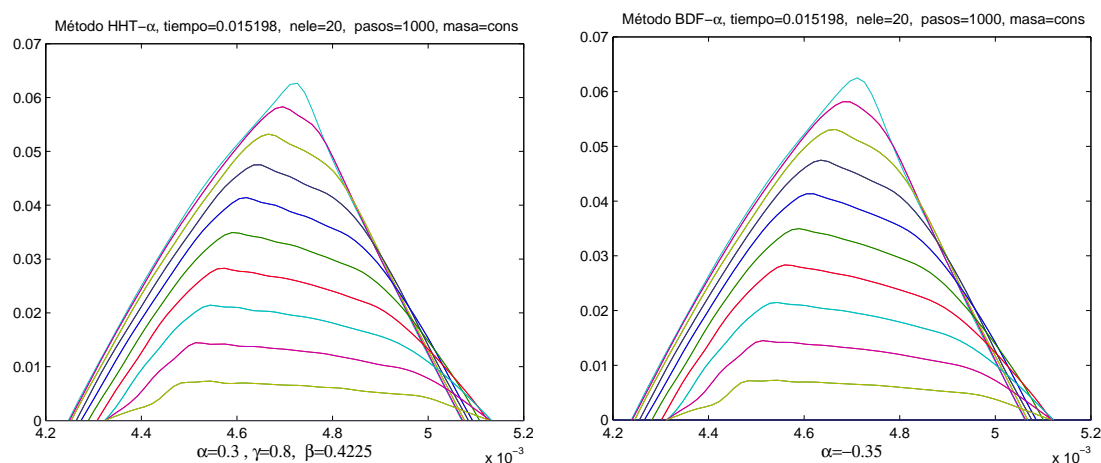


Figura 7.38: Detalle de las Figuras correspondientes a $\text{HHT-}\alpha = 0,3$ y $\text{BDF-}\alpha = -0,35$.

- En el caso del método $\text{HHT-}\alpha$, el cálculo de $R(\hat{d}_{n+1-\alpha})$ en \bar{R} y el de $\frac{\partial R(\hat{d}_{n+1-\alpha})}{\partial \hat{d}_{n+1-\alpha}}$ en \bar{J} son los que más coste computacional suponen, términos que aparecen en las expresiones (7.108) y (7.112) respectivamente.
- En el caso del método $\text{BDF-}\alpha$, las operaciones más costosas resultan el cálculo de f_{n+2}^m en \bar{R} y el cálculo de j_{n+2}^m en \bar{J} , que son los términos que aparecen en la expresión (7.80). También en este caso, la dimensión de las matrices del método $\text{BDF-}\alpha$ es el doble que las del método $\text{HHT-}\alpha$. Así, el método $\text{BDF-}\alpha$ necesita 1,54 segundos para resolver el sistema de ecuaciones del conjunto total de iteraciones, frente a 0,1 segundos que necesita el método $\text{HHT-}\alpha$. Esta diferencia no pesa en el balance final dado que es el cálculo de \bar{R} y \bar{J} lo que se lleva la mayor parte del tiempo.

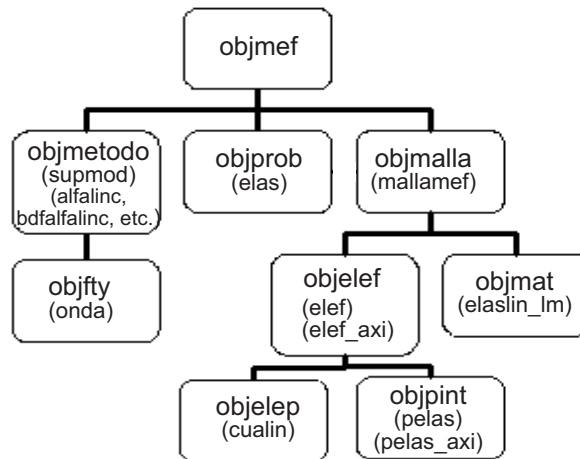


Figura 7.39: Arquitectura del ejemplo 3.

7.4.3. Ejemplo 3: Elasticidad dinámica lineal

En este ejemplo veremos cómo se atenúan los modos altos de vibración en un bloque de sección cuadrada y longitud de lado de 2 unidades, que va a vibrar libremente en régimen de deformación plana a partir de unas CI correspondientes a un alargamiento vertical de una unidad de longitud (tracción simple).

Se trata de un problema académico cuya finalidad es mostrar las funcionalidades de la metodología adoptada y el comportamiento de los métodos. Se han tomado los valores $E = 1000$, $\nu = 0,25$ y $\rho = 1$ para el módulo de Young, de Poisson y para la densidad del material respectivamente. A pesar de las dimensiones de la deformada inicial y de la escasa rigidez del material, el problema se trata linealmente, por lo que los resultados obtenidos deberían multiplicarse por el factor de escala adecuado.

Se ha realizado una discretización mediante cuadriláteros bilineales de 10 elementos por lado y se han impuesto condiciones Dirichlet nulas para los desplazamientos verticales del eje de simetría horizontal y para los desplazamientos horizontales del eje de simetría vertical. Se ha calculado la solución del sistema de EDOs de la semidiscretización MEF, de dos maneras distintas: mediante superposición modal y utilizando diversos métodos numéricos (HHT- α , BDF- α , ENDF y MENDF). Para ello, basta con definir como *objmetodo* en la Figura 7.39, el objeto *supmod* o *alfalinc*, *bdfalfalinc*, *endflinc*, *mendflinc* respectivamente.

Realizado el cálculo en el primer caso, el objeto *supmod* proporciona entre sus datos, las frecuencias y modos de vibración, los factores de participación modal, y la solución temporal para las CI especificadas y el número de modos de participación indicados (por orden de mayor factor de participación).

7.4.3.1. Solución modal y paso a paso

En la Figura 7.40 se muestran los modos 1, 5, 8 y 13 (simétricos), del bloque completo (10×10). Son los 4 modos con mayor factor de participación para las CI de este problema. En particular corresponden a las frecuencias 44,78, 51,79, 60,60, 89,10 y los factores de participación son $0,33037$, $3,3893 \cdot 10^{-15}$, $7,9993 \cdot 10^{-15}$ y $8,8053 \cdot 10^{-15}$ respectivamente.

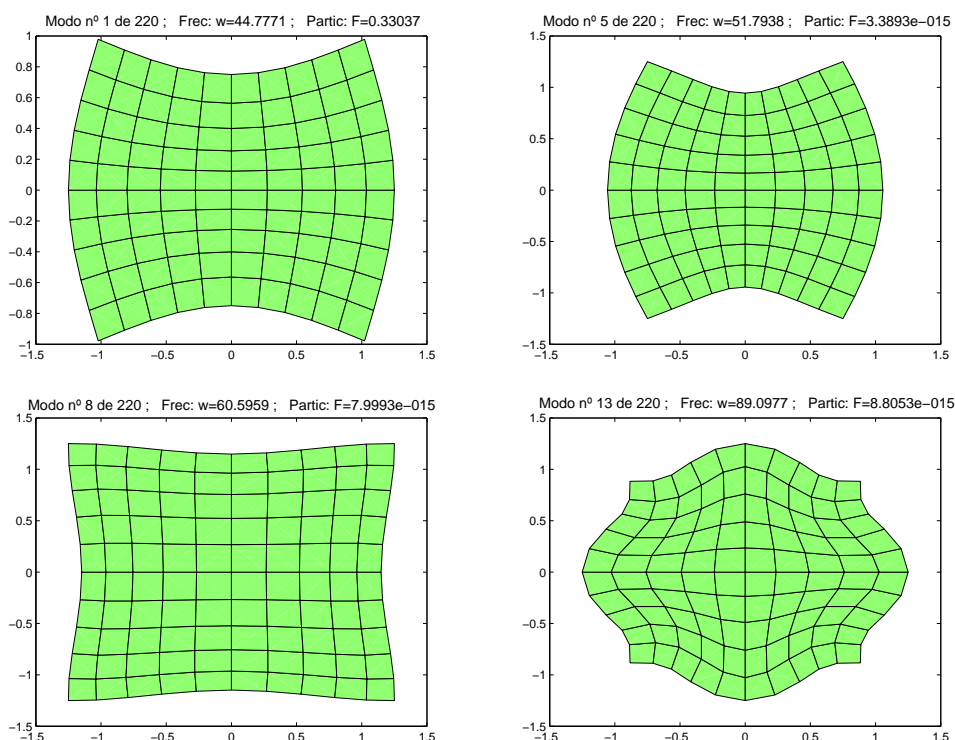


Figura 7.40: Modos 1, 5, 8 y 13 del bloque del ejemplo 3.

Dada la simetría, en lo que sigue los cálculos se realizan sobre la cuarta parte de la discretización. Se muestran los resultados correspondientes a cuatro tipos de cálculo: superposición modal (considerando una discretización temporal de 1000 puntos), métodos HHT- α , BDF- α , ENDF y MENDF dando 200 pasos, 1000 pasos y 10000 pasos. En la Figura 7.41 se muestran las tensiones y la deformada en el instante final ($t = 0,5$).

En la Figura 7.42 se puede ver la evolución temporal de todos los desplazamientos nodales. Se ha utilizado el método trapezoidal (es decir, HHT- $\alpha = 0$) y se han dado 200 pasos. En la parte inicial se observa el fenómeno de transmisión de onda (cómo los nodos inferiores precisan un tiempo para iniciar su movimiento).

Nos fijaremos en el movimiento vertical de un único nodo significativo de la discretización. En particular, nos fijamos en el nodo superior del eje de simetría vertical y calculamos su evolución utilizando los métodos citados. En la Figura 7.43 se puede

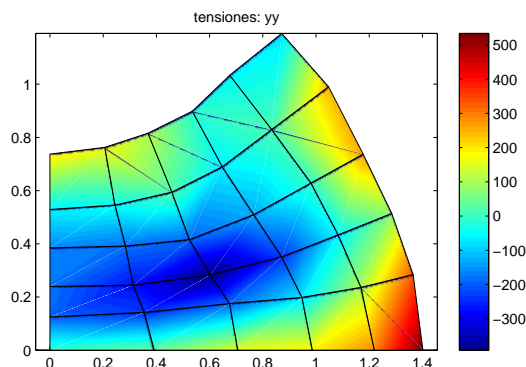


Figura 7.41: Tensiones y la deformada del ejemplo 3 en el instante final ($t = 0,5$).

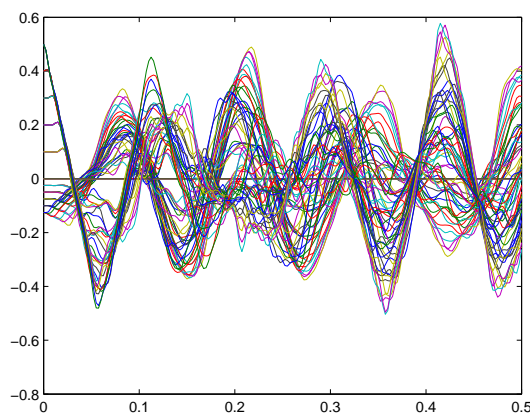


Figura 7.42: Evolución temporal de los desplazamientos nodales (mét. trap. 200 pasos).

ver la evolución de este nodo al resolver el problema con el método $\text{HHT-}\alpha = 0$ y por superposición modal. Se confirma la hipótesis de que la superposición modal, a pesar del cálculo numérico de los modos, es prácticamente la solución exacta del sistema de EDOs (error de discretización). Se puede observar con más nitidez el detalle correspondiente al 2º pico superior en la imagen derecha de la Figura 7.43.

Se obtienen resultados muy parecidos cuando utilizamos los métodos $\text{BDF-}\alpha = -0,5$ y el método ENDF3 para resolver el mismo problema (ver Figura 7.44). También el método MENDF3 proporciona resultados muy similares a los anteriores.

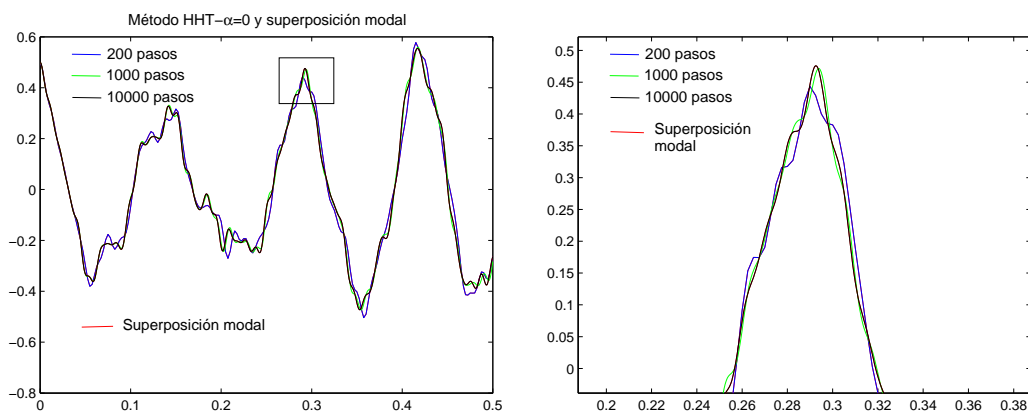


Figura 7.43: Evolución temporal nodo superior (HHT- $\alpha = 0$ y *supmod*) y detalle.

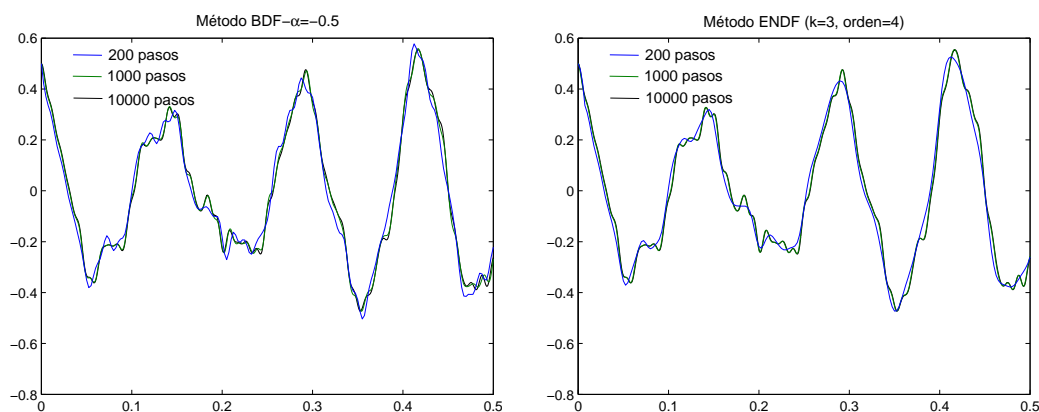


Figura 7.44: Evolución temporal nodo superior (BDF- $\alpha = -0,5$ y ENDF3).

7.4.3.2. Atenuación, amortiguamiento modal

Ahora cambiamos nuestro foco de interés, de los modos de mayor participación modal a los de menor y mayor frecuencia. Imponiendo como CI la deformada correspondiente al modo principal (es decir, menor frecuencia) Figura 7.45, y aplicando el método trapecoidal (HHT- $\alpha = 0$) con 200 pasos, obtenemos la evolución temporal de los desplazamientos nodales Figura 7.46. En esta figura se observa la respuesta armónica propia de la excitación modal y se aprecia la diferencia con la respuesta de la Figura 7.42. Las soluciones que se obtienen con los métodos BDF- $\alpha = -0,5$, HHT- $\alpha = 0,3$, ENDF3 y MENDF3 son muy parecidas.

No ocurre lo mismo sin embargo cuando la CI que se impone es el último modo, el de mayor frecuencia. Imponiendo la deformada de la Figura 7.47 como condición inicial, con el método trapecoidal, se obtiene la evolución temporal de los nodos de la Figura

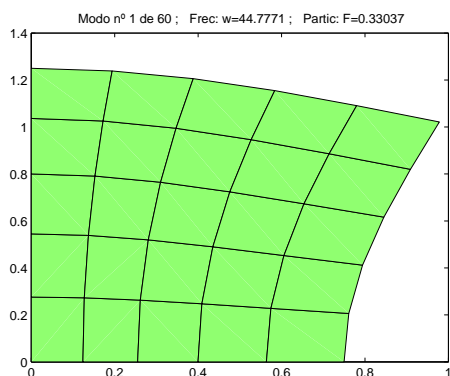


Figura 7.45: Modo 1 del bloque del ejemplo 3.

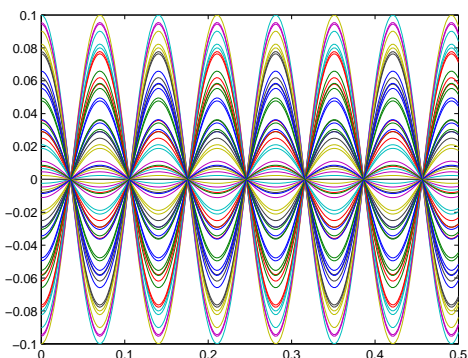


Figura 7.46: Evolución temporal de los desplazamientos nodales (mét. trap. 200 pasos).

7.48, donde puede observarse que ha habido que reducir el tiempo en un factor de 10 (y utilizar 1000 pasos) para poder observar la alta frecuencia de la respuesta armónica a esta excitación modal.

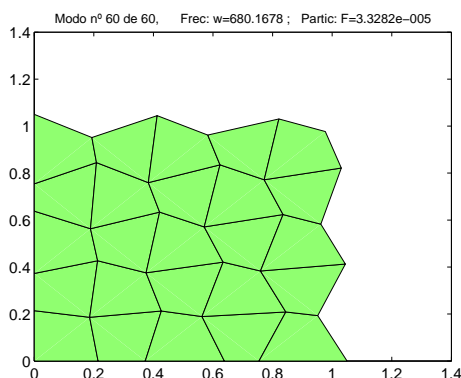


Figura 7.47: Modo 60 del bloque del ejemplo 3.

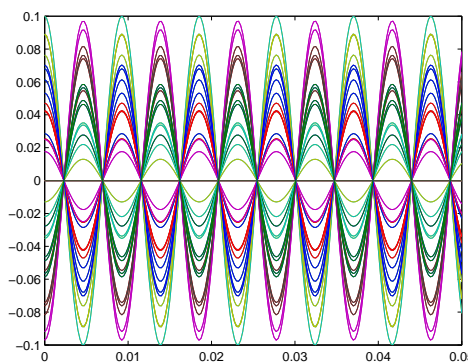


Figura 7.48: Evolución temporal de los nodos (mét. trap. 1000 pasos).

Para los métodos $\text{HHT-}\alpha = 0,3$ y $\text{BDF-}\alpha = -0,35$ y dando 317 pasos (valor crítico) se observa la reducción drástica (aniquilamiento) de la vibración libre del último modo. En la imagen izquierda de la Figura 7.49 se ha incluido la solución obtenida con el método $\text{BDF-}\alpha = -0,35$. Para 1000 pasos se mantiene la propiedad de aniquilamiento, imagen derecha de la Figura 7.49 aunque no tan rápidamente; tendencia que se mantiene para 2000 pasos como se observa en la Figura 7.50. Sin embargo, para las condiciones más severas (317 pasos) el primer modo pierde muy poca precisión, ver Figura 7.51.

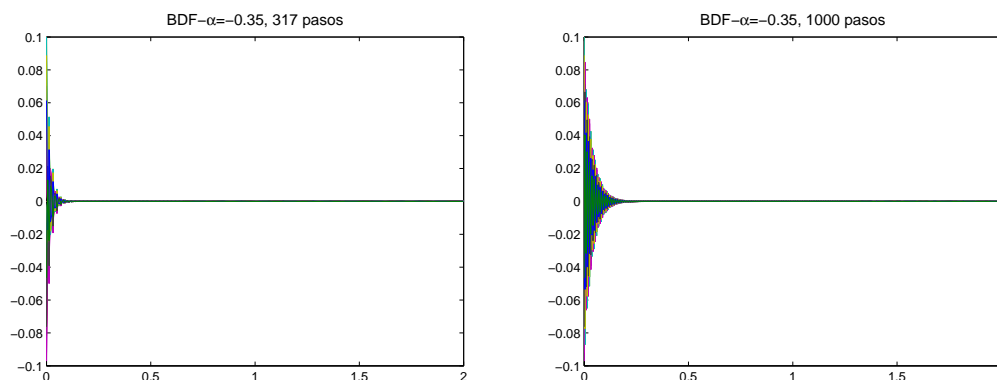


Figura 7.49: Evolución temporal de los nodos BDF- $\alpha = -0,35$ (317 y 1000 pasos).

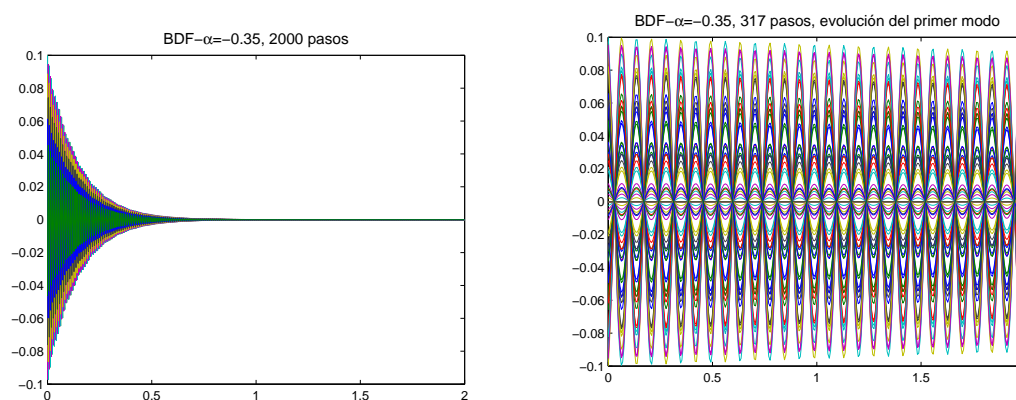


Figura 7.50: BDF- $\alpha = -0,35$ y 2000 pasos. Figura 7.51: Evolución del primer modo BDF- $\alpha = -0,35$ y 317 pasos.

Todo ello se puede observar con más perspectiva (más modos) en la Figura 7.52, donde se muestra la evolución temporal del nodo superior central del bloque, pero para un espectro amplio de modos.

Para los 3 primeros modos de mayor participación modal, que son los modos 1, 5 y 8 y dando 317 pasos con el método BDF- $\alpha = -0,35$, se obtienen los resultados de la imagen izquierda de la Figura 7.53 y cuando se dan 1000 pasos, se obtienen los resultados de la imagen derecha de la misma figura. Las figuras que se obtienen con el método HHT- $\alpha = 0,3$ son muy parecidas.

Al final probablemente compensa el aumentar el número de pasos, ya que se amortiguan menos las frecuencias altas, pero también las bajas y se pierde menos precisión.

En la Figura 7.54 se muestra la solución a las CI del problema, para los métodos HHT- $\alpha = 0,3$ y BDF- $\alpha = -0,35$ dando 317, 1000 y 2000 pasos; para la superposición

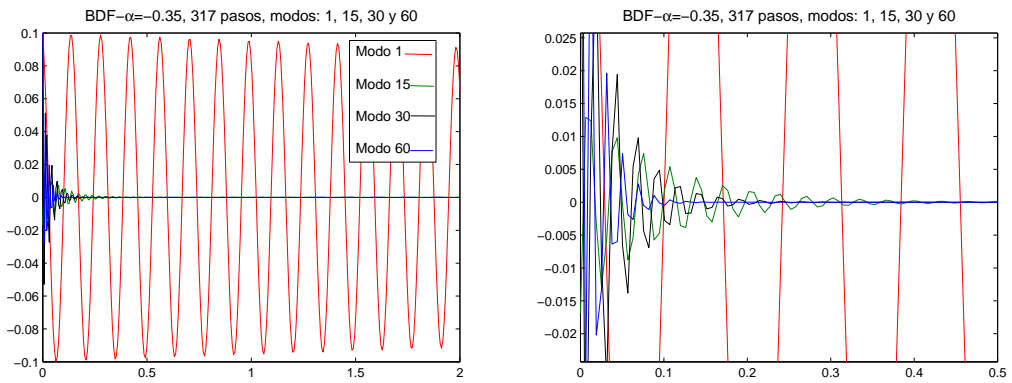


Figura 7.52: Evolución temporal de un nodo BDF- $\alpha = -0,35$, 317 pasos (dcha. detalle).

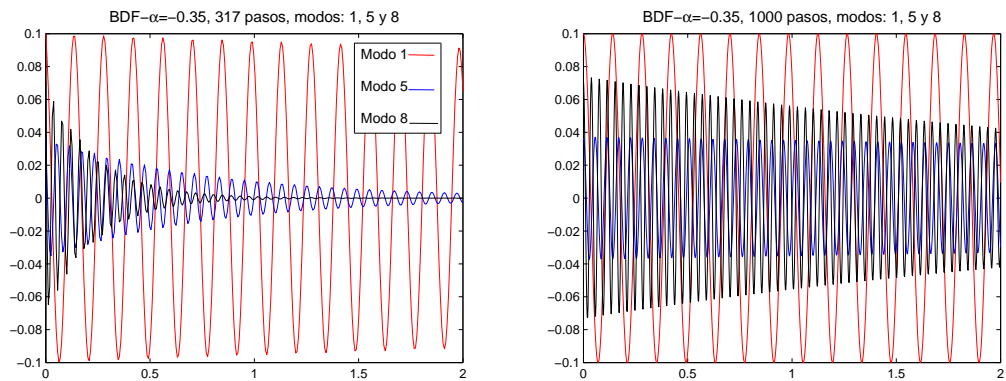


Figura 7.53: Evolución temp. de un nodo BDF- $\alpha = -0,35$, 317 y 1000 pasos.

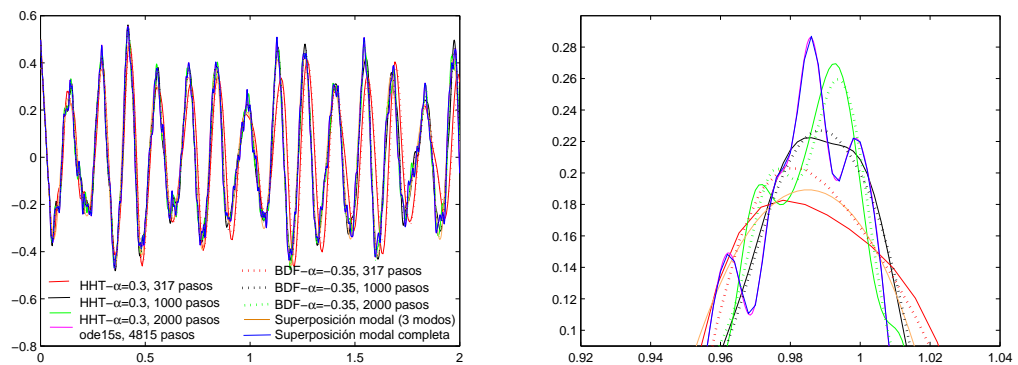


Figura 7.54: Evolución temporal nodo superior con varios métodos, derecha detalle.

modal completa y para los 3 modos de mayor participación, y para la *ode15s* (*objmetodo* objeto *odesuite-ode15s*). Cuando resolvemos con los métodos HHT- α y BDF- α dando 2000 pasos, con la *ode15s* y mediante superposición modal completa, vemos que no se eliminan las altas frecuencias. A las soluciones obtenidas con los métodos HHT- α y BDF- α con 317 pasos les falta un poco de los modos 2 y 3 para coincidir con el resultado de los tres modos de mayor participación modal. Los resultados obtenidos con HHT- α y BDF- α con 1000 pasos, pierden menos de los modos 1, 2 y 3 pero eliminan menos las altas frecuencias. Los resultados obtenidos mediante superposición modal completa y con la *ode15s* son muy parecidos.

7.4.4. Ejemplo 4: Tracción elastoplástica de bloque axisimétrico

Hemos considerado la tracción elastoplástica de bloque axisimétrico con cavidad interior. Dentro de este ejemplo analizaremos dos casos: el comportamiento estático y el comportamiento dinámico no lineal.

Se pretende calcular la evolución temporal de un cilindro de 20 unidades de radio y 20 unidades de altura que se deforma en tracción simple con una fuerza axial $P = 3 \cdot 10^5$, módulo de elasticidad $E = 2 \cdot 10^6$, módulo de Poisson $\nu = 0,25$, módulo de endurecimiento $H = 10^6$ y tensión de fluencia $\sigma_e = 2000$. Dada la simetría modelizaremos sólo la mitad de la altura del cilindro que analizaremos en 2D en el plano axisimétrico, mediante la agregación de objetos que se muestra en la Figura 7.55. En esta figura se observa la definición del problema como elástico (objeto *elas*) sin grandes deformaciones, deformación lineal (*deflin*) y material no lineal (objeto *plasxy*).

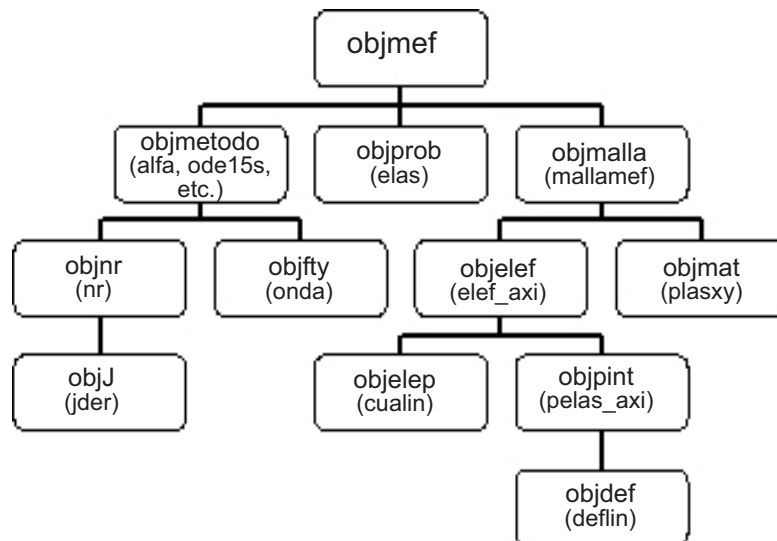


Figura 7.55: Arquitectura del ejemplo 4, caso dinámico.

7.4.4.1. Caso estático

Como paso previo, se resuelve en 5 pasos, un caso similar pero sin cavidad interna, representando en la Figura 7.56 el diagrama fuerzas-desplazamientos que es manifestación del diagrama de tracción del material.

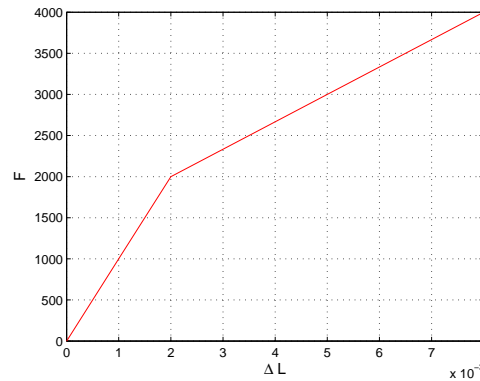


Figura 7.56: Diagrama fuerzas-desplazamientos del ejemplo 4.

El cilindro con cavidad se resuelve en 10 pasos, mostrándose en la Figura 7.57 la discretización y la deformada (factor de escala de 5) sobre la configuración inicial, observándose el efecto de la concentración de tensiones. En la Figura 7.58 se pueden ver las isóneas de deformación plástica: las más altas lógicamente en la base, por el efecto de concentración de tensiones, y las más bajas sobre el taladro por el efecto sombra o de aislamiento que el agujero ejerce sobre la laja, ya que las líneas de fuerza deben circular el agujero. Se observa también el alto valor de las deformaciones plásticas obtenidas, 3 % de media.

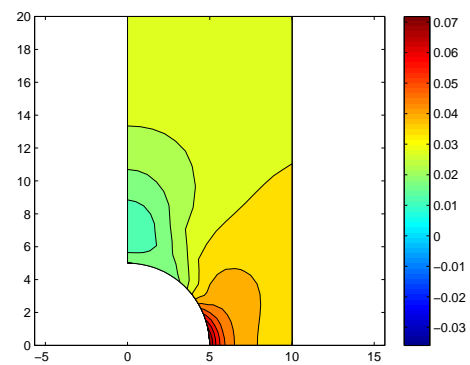
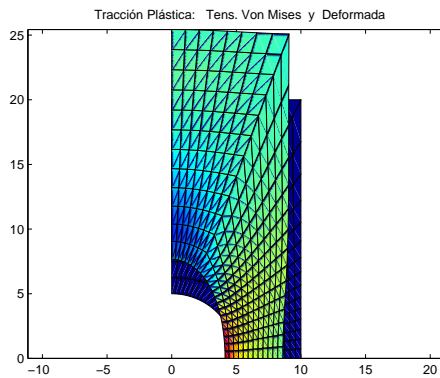


Figura 7.57: Discretización y deformada del ejemplo 4.

Figura 7.58: Isóneas de deformación plástica del ejemplo 4.

En la Figura 7.59, se muestra el diagrama fuerzas-desplazamientos, correspondiente al punto central superior de la laja. Se observa el cambio de pendiente al inicio, cuando toda la laja ya deforma plásticamente. Ello se debe al valor muy elevado de la fuerza de tracción ($P = 3 \cdot 10^5$ unidades), por lo que la mayor parte del proceso de carga discurre con deformación plástica sobre toda la pieza.

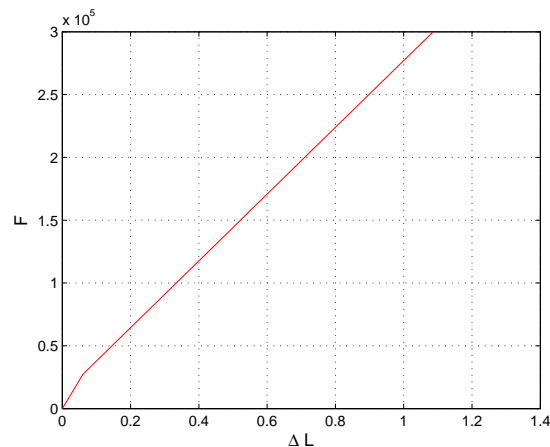


Figura 7.59: Diagrama fuerzas-desplazamientos del punto central superior de la laja.

7.4.4.2. Caso dinámico

Resolvemos el ejemplo 4 en régimen dinámico, condiciones iniciales nulas y carga constante de tracción. En la Figura 7.60 se muestra la evolución temporal del desplazamiento vertical en el nodo central de la parte superior de la pieza obtenido con el método trapezoidal y 50 pasos (225,31 segundos). Se observa la oscilación elástica en torno a la deformación estática, como ocurre en los casos de carga dinámica constante sin amortiguamiento. Nos preguntamos si la forma de la oscilación es consecuencia de la baja resolución, sólo 50 pasos. Con este fin, en la Figura 7.61 se muestran los resultados obtenidos con 500 pasos (2204,36 segundos) utilizando el mismo método.

Se confirma la hipótesis planteada: con más pasos el rizo es uniforme y de alta frecuencia. La viga inicia su deformación elastoplástica desde su configuración inicial recta, y cuando se acerca a la deformada estática prosigue su deformación por la inercia acumulada. Cuando finalmente para, se descarga y vibra elásticamente manteniendo la deformación plástica permanente que observamos en la Figura 7.62.

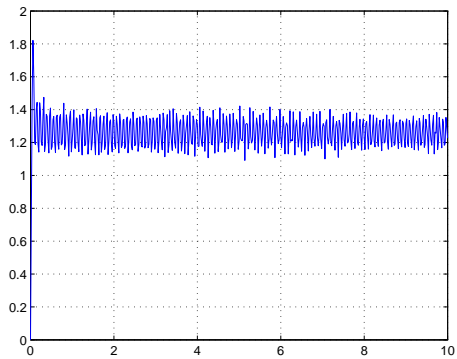
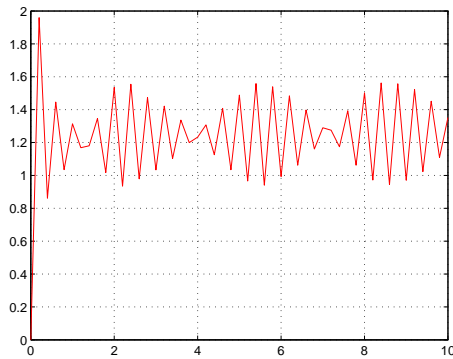


Figura 7.60: Evolución temporal del desplazamiento vertical (nodo central sup.).
Figura 7.61: Resultados con el método de capturas 500 pasos.

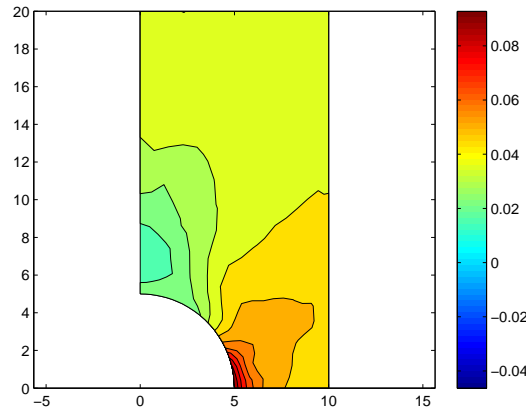


Figura 7.62: Isolíneas de deformación plástica del ejemplo 4 (caso dinámico).

7.4.5. Ejemplo 5: Flexión de viga hiperelástica incompresible

También en este ejemplo analizaremos los siguientes dos casos: el comportamiento estático y el comportamiento dinámico no lineal. Los materiales que soportan grandes deformaciones suelen deformar sin variación de volumen, es decir, con módulo de Poisson $\nu = 0,5$ o muy cercano a este valor. En ese caso es preciso modificar la formulación neo-hookeana para que funcione correctamente en incompresibilidad, lo cual se ha hecho en el nuevo material *neoh_nc*, válido para elasticidad 3D y 2D axisimétrica y en deformación plana. Para tensión plana, la formulación adaptada admite una versión simplificada que ha quedado reflejada en el material *neoh_nc_tp*.

En este ejemplo, primero se trata la incompresibilidad del material en la viga en tensión plana, y en segundo lugar en deformación plana.

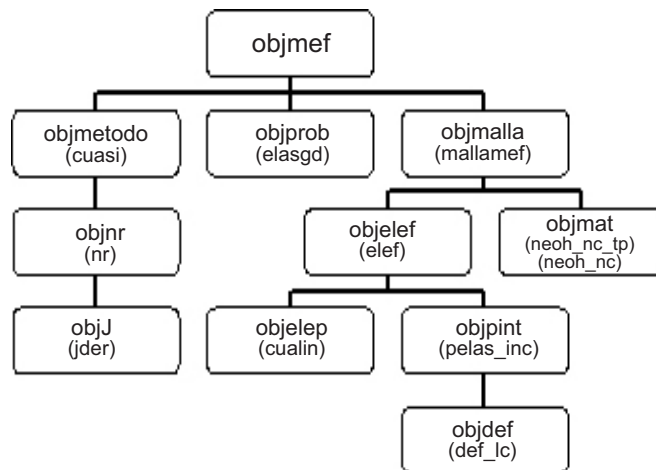


Figura 7.63: Arquitectura del ejemplo 5, caso estático.

7.4.5.1. Caso estático en tensión plana

Se emplea el material neohookeano incompresible para tensión plana, definido en el objeto *neoh_nc_tp*, obteniéndose las tensiones y deformada de la Figura 7.64. En la Figura 7.65 se han dibujado los desplazamientos nodales de este ejemplo y se observa de nuevo la no linealidad del material.

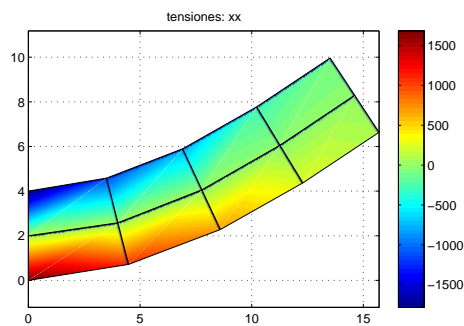


Figura 7.64: Tensiones y deformada del ejemplo 5, caso estático.

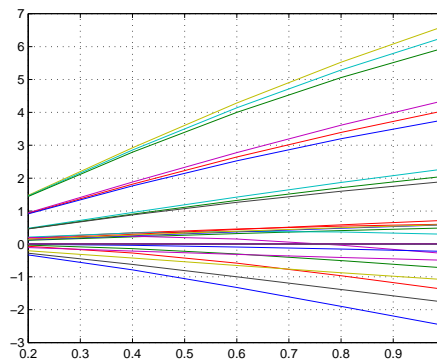


Figura 7.65: Desplazamientos nodales del ejemplo 5, caso estático.

7.4.5.2. Caso dinámico en tensión plana

En la Figura 7.66 se pueden ver las tensiones y la deformada del ejemplo 5 caso dinámico. Se observa que la flecha dinámica viene a ser el doble que la estática y que las

tensiones son mayores en el caso dinámico, debido a que la curvatura de la deformada es mayor. En la Figura 7.67 se muestran los desplazamientos nodales.

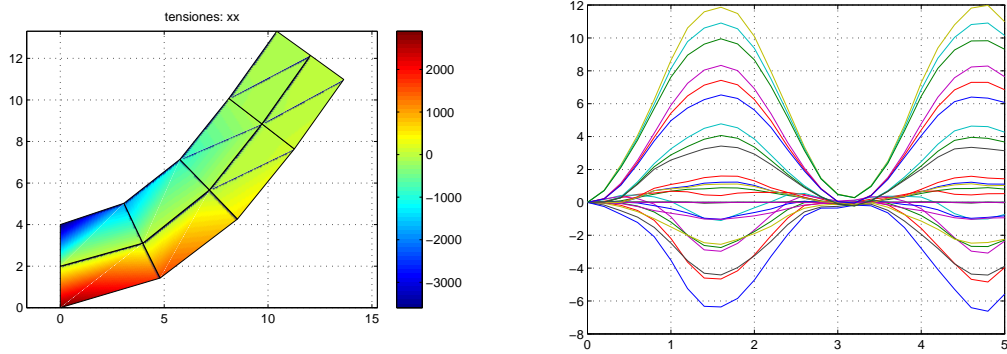


Figura 7.66: Tensiones y deformada del ejemplo 5, caso dinámico. Figura 7.67: Desplazamientos nodales del ejemplo 5, caso dinámico.

En la Figura 7.68 se observan los verticales del nodo central del extremo del voladizo. Se han comparado con las obtenidas para el caso de elasticidad lineal y con las calculadas por el método trapezoidal y la *ode15s* (5934 pasos). Por último, y con el fin de observar si la solución trapezoidal (50 pasos) converge a la solución de la *ode15s* se realiza un nuevo cálculo con el método trapezoidal dando 500 pasos, cuyo resultado se observa en la misma figura. Se observa que la solución de la *ode15s* y la solución del método trapezoidal (500 pasos) casi coinciden, lo cual confirma la convergencia a una solución única con el aumento del número de pasos.

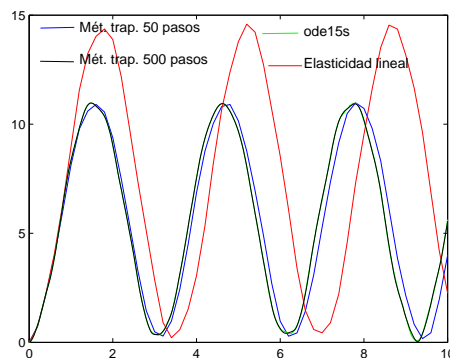


Figura 7.68: Verticales del nodo central del extremo del voladizo, ejemplo 5.

7.4.5.3. Caso dinámico en deformación plana

En la Figura 7.69 se muestra la solución obtenida en deformación plana y material neohookeano utilizando el método trapezoidal 50 pasos, para $\nu = 0,25$ y $\nu = 0,4999$. Se aprecia claramente el bloqueo de malla en incompresibilidad. La reformulación del material en un nuevo objeto *neoh_nc*, permite obtener en 500 pasos y $\nu = 0,4999$ la solución (ya que no converge en 50 pasos), donde claramente se observa la mejora con respecto al bloqueo de malla.

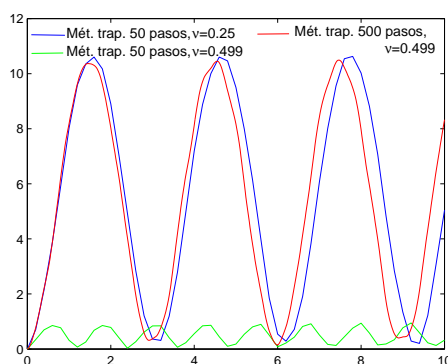


Figura 7.69: Solución del ejemplo 5 en deformación plana y material neohookeano.

7.4.6. Ejemplo 6: Movimiento de rotación de muelle no lineal

Consideraremos un muelle no lineal fijado en un centro de rotación O , con una masa $m = 2$ concentrada en el extremo final del mismo, con un parámetro de rigidez del muelle $k = 15$, $l_0 = 10$ la posición inicial que ocupa la masa y condiciones iniciales [4]:

$$d_0 = [0 \ 10]^T, \quad (m \cdot v_0) = [-20 \ 0]^T \quad (7.120)$$

siendo d la posición y $(m \cdot v)$ el momento lineal. La energía potencial de este sistema viene dada por $V(l) = 1/2k(l - l_0)^2$.

Resolveremos este problema utilizando los métodos energía-momento, HHT- $\alpha = 0,111$, BDF- $\alpha = -0,445$ y EDMC-1 $\chi_1 = \chi_2 = 0,11$. Estos tres últimos tienen el mismo $\rho_\infty = 0,8$. El problema se ha resuelto en dos casos distintos: cuando el intervalo temporal es $T = [0, 2000]$ y el tamaño de paso $h = 1$; y para $T = [0, 50 \cdot h]$ y tamaño de paso $h = 1,6775$.

Se pueden ver en las Figuras 7.70, 7.71, 7.72 y 7.73 los resultado que se han obtenido para el primer caso ($T = [0, 2000]$ y $h = 1$). Vemos que el método energía-momento conserva tanto el momento angular como la energía, el método EDMC-1 conserva el momento angular y disipa la energía, y los métodos HHT- α y BDF- α disipan ambas cosas y además de similar manera.

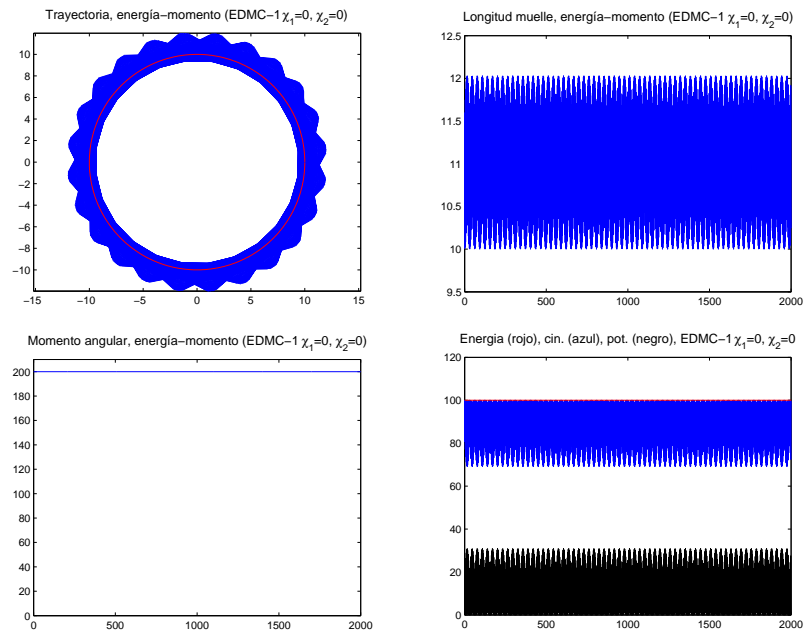


Figura 7.70: Solución del sistema no lineal masa-muelle, energía-momento $h = 1$.

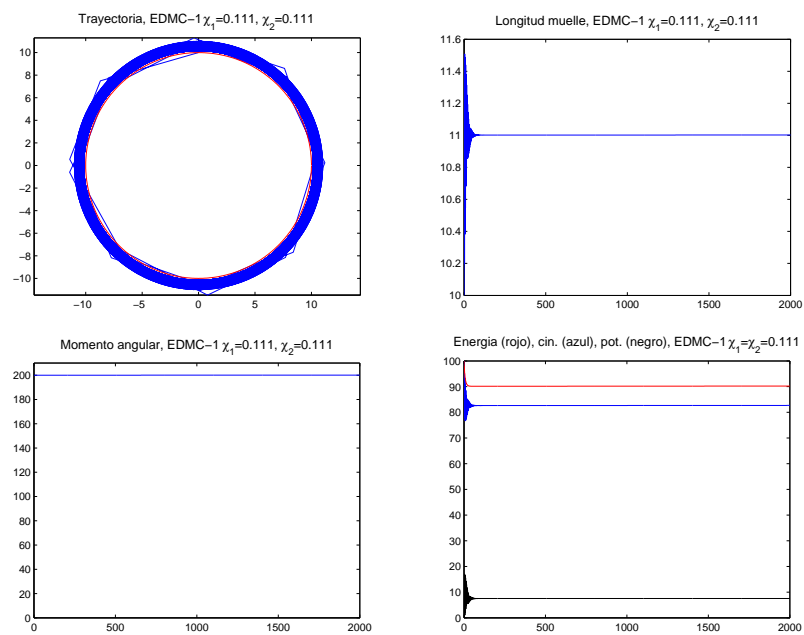


Figura 7.71: Solución del sistema no lineal masa-muelle, método EDMC-1 $h = 1$.

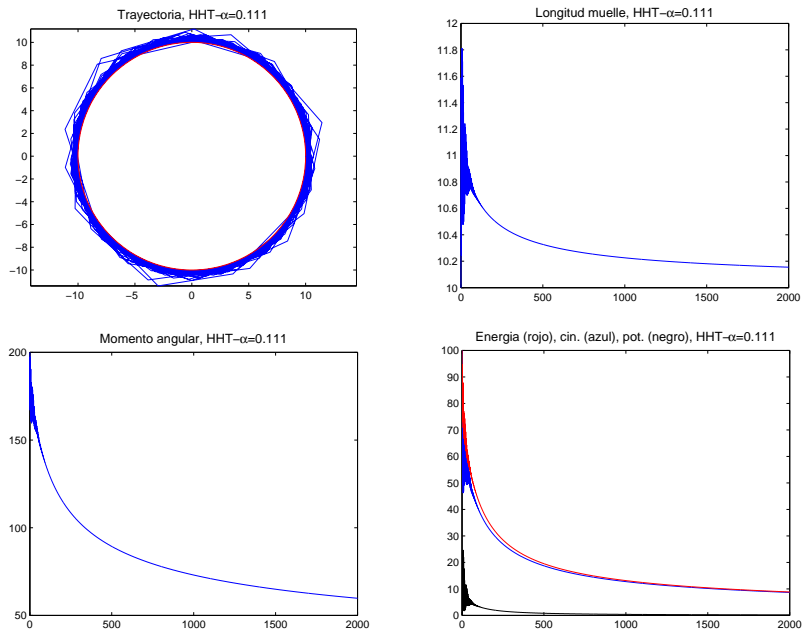


Figura 7.72: Solución del sistema no lineal masa-muelle, método HHT- α $h = 1$.

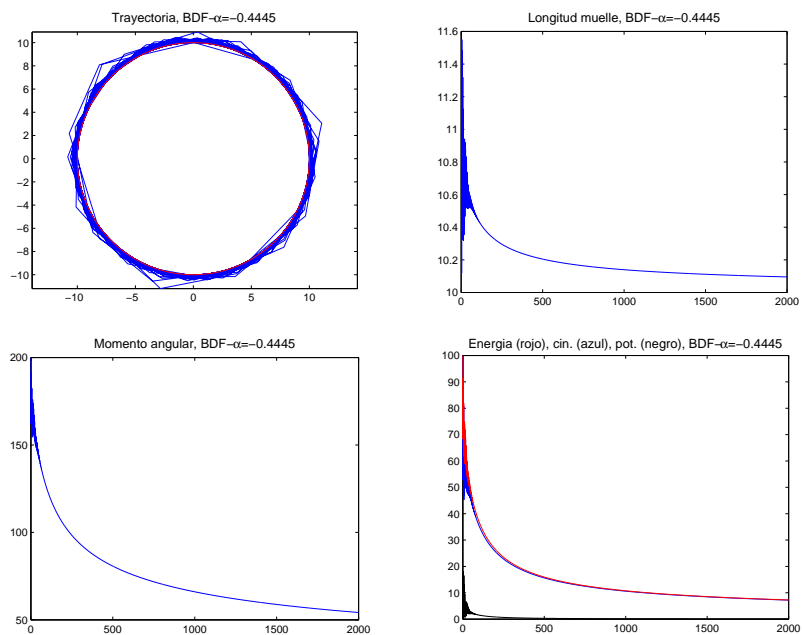


Figura 7.73: Solución del sistema no lineal masa-muelle, método BDF- α $h = 1$.

Considerando un tamaño de paso mayor, $h = 1,6775$ y un intervalo temporal $T = [0, 50 \cdot h]$, se observa que los métodos energía-momento y EDMC-1 funcionan de similar manera que con un tamaño de paso menor, Figuras 7.74 y 7.75. El método HHT- α presenta un aumento de la energía, ver Figura 7.76, mientras que en el método BDF- α aunque ésta aumenta un poco al principio tiende a disiparse según avanzamos en el tiempo, Figura 7.77.

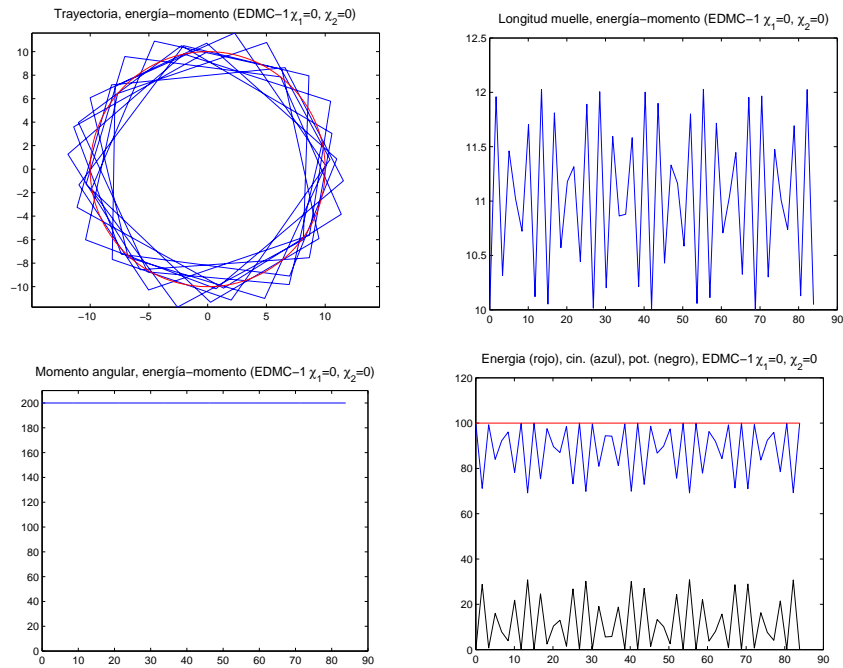


Figura 7.74: Solución del sistema no lineal masa-muelle, energía-momento $h = 1,6775$.

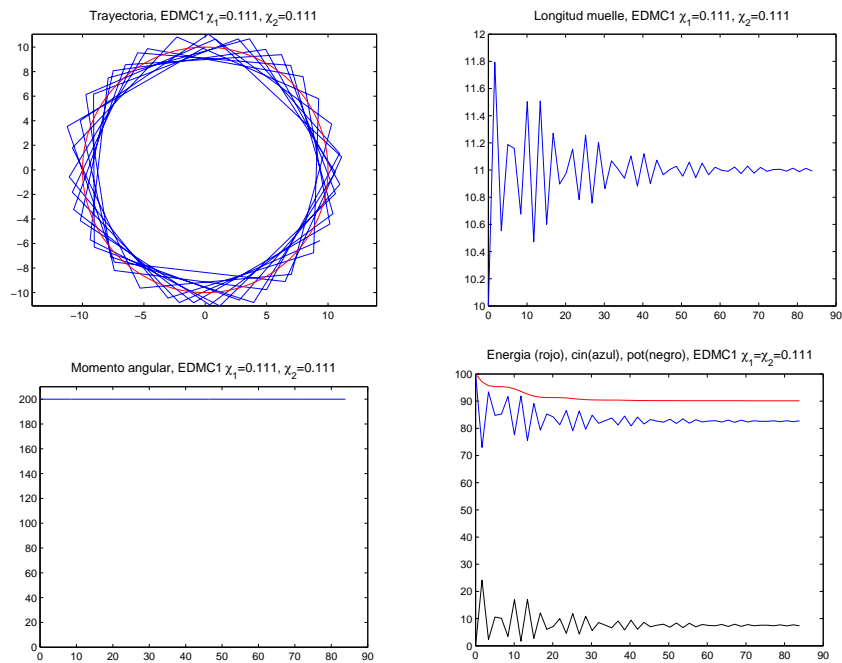


Figura 7.75: Solución del sistema no lineal masa-muelle, método EDMC-1 $h = 1,6775$.

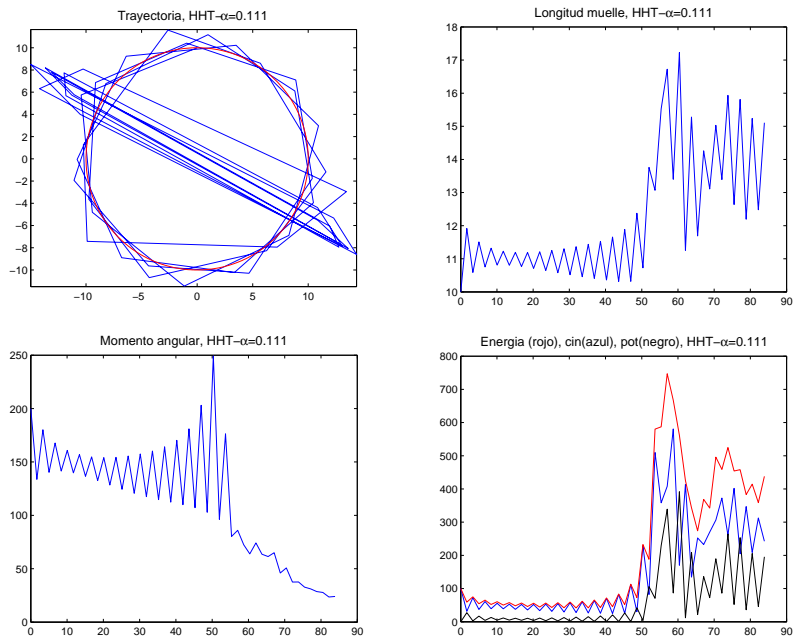


Figura 7.76: Solución del sistema no lineal masa-muelle, método HHT- α $h = 1,6775$.

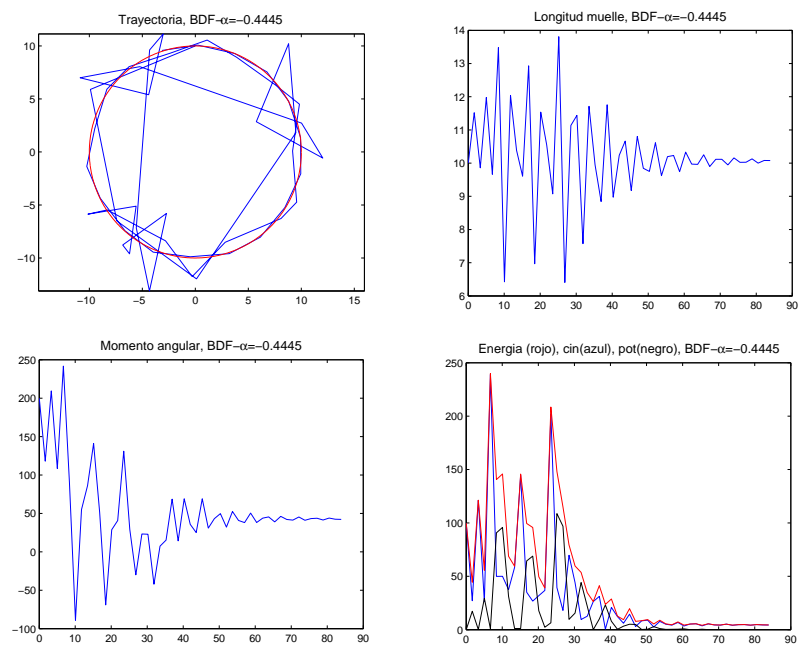


Figura 7.77: Solución del sistema no lineal masa-muelle, método BDF- α $h = 1,6775$.

Capítulo 8

Conclusiones y futuras líneas de investigación

Nunca te das cuenta de lo que ya has hecho; sólo puedes ver lo que te queda por hacer.

MARIE CURIE

Este trabajo de tesis se apoya en los siguientes cuatro puntos:

- La existencia en la actualidad de herramientas computacionales modernas como Matlab para el cálculo matemático.
- La penetración de tales herramientas, que cada vez son más utilizadas, tanto en el ámbito industrial como en el académico, como ocurrió anteriormente con las calculadoras de bolsillo.
- La posibilidad que ofrecen estas herramientas para realizar cálculos complejos, combinando recursos resolutivos propios de carácter general con las facilidades para la programación de necesidades más específicas.
- En particular, la posibilidad que ofrecen estas herramientas de integrar directamente el sistema de EDOs rígido proveniente de la semidiscretización del método de los elementos finitos, mediante funciones específicas de la *odesuite* de Matlab.

En este contexto se expusieron en el Capítulo 1 los objetivos de la tesis, entre los que sobresalen los siguientes:

- Conocer la arquitectura y contrastar la algorítmica de las funciones *ode15s* y *ode45* de Matlab.
- Analizar las contradicciones observadas en la aplicación de la *ode15s* a la resolución de sistemas de EDOs rígidos de orden 2 que se producen en los problemas vibratorios.

- Proponer alternativas BDF para los sistemas de EDOs de orden 1 que ensanchen las regiones de estabilidad, en particular con métodos BDF extendidos.
- Conocer la algorítmica de los métodos clásicos de la mecánica computacional para EDOs de orden 2: método Newmark y método HHT- α . En particular, los aspectos relacionados con la eliminación de las altas frecuencias de la semidiscretización MEF.
- Comparar los métodos BDF de paso variable de la *ode15s*, y los de paso constante con los métodos clásicos de la mecánica computacional.
- Proponer alternativas BDF para los sistemas de EDOs de orden 2 que como el método HHT- α permitan parametrizar la eliminación de altas frecuencias.
- Profundizar en el desarrollo de alternativas de programación, proponiendo una metodología orientada a objeto que aprovecha la modularidad del MEF para generar un entorno de desarrollo que facilite la experimentación y donde sea posible añadir nuevas ideas y desarrollos sin modificar las anteriores.

8.1. Conclusiones y aportaciones de la tesis

Las conclusiones a las que hemos llegado tras realizar este trabajo son las siguientes:

- El proceso de crear un método numérico para la resolución de EDOs de orden 1 o de orden 2 no es trivial. Requiere conocer conceptos como el orden de precisión, la estabilidad, etc., calcular estos parámetros para el método numérico que se propone, programar el método para que funcione y adaptarlo adecuadamente, en el caso de que se quiera incorporar a un entorno de programación orientado a objeto (POO).
- La importancia de utilizar un método numérico con buena precisión y con una región de estabilidad extensa al resolver sistemas de EDOs rígidas.
- El hecho de que en Matlab estén implementadas las funciones *ode45* y *ode15s* no es casual. Ambas son de paso adaptativo y pueden trabajar en órdenes relativamente altos (orden 5). La primera es explícita y con región de estabilidad limitada y la segunda es implícita y con una región de estabilidad mayor. En el caso de las funciones *ode45* y *ode15s* la estimación del error local está ideada y gestionada eficazmente: la primera utiliza la diferencia entre los valores numéricos obtenidos con los métodos Runge-Kutta de orden 5 y 4 (ambos disponibles) y la segunda, utiliza como estimación el error local de truncamiento del método BDF/NDF, que al ser una diferencia hacia atrás resulta fácilmente calculable debido al almacenamiento de la matriz de diferencias.
- La ineficiencia computacional de la *ode15s* en problemas vibratorios de gran radio de rigidez, frente a los métodos de paso constante incondicionalmente estables,

debido a que en los métodos BDF de orden de precisión superior a 2 una parte del eje imaginario no pertenece a la región de estabilidad.

- Ha sido posible ampliar la región de estabilidad mediante métodos BDF extendidos, alcanzando hasta orden de precisión 4 con estabilidad incondicional. Esta extensión mejora sensiblemente la eficiencia computacional de los métodos BDF de orden superior en los problemas vibratorios, a costa de mayor coste computacional que los métodos incondicionalmente estables de orden 2, pero con mayor orden de precisión.
- La necesidad de estudiar la evolución del factor de amplificación con la frecuencia en los problemas de vibraciones y onda, para conocer parámetros como el amortiguamiento algorítmico, el error relativo en el periodo y en particular, la capacidad que tiene el método para eliminar las altas frecuencias provenientes de la semidiscretización MEF.
- La importancia de utilizar métodos numéricos capaces de eliminar las altas frecuencias provenientes de la semidiscretización MEF, mejorando la precisión y facilitando la convergencia en los problemas no lineales.
- Ha sido posible incorporar control paramétrico del amortiguamiento en los problemas vibratorios al método BDF2, de manera similar a como lo hace el método HHT- α . En este sentido, se ha constatado la incapacidad de los métodos de paso adaptativo para el control paramétrico de la disipación de las altas frecuencias.
- La similar eficiencia computacional de los métodos BDF- α y HHT- α en los problemas vibratorios no lineales, y la superioridad del segundo de ellos en los lineales, a pesar de que en este caso ambos son muy rápidos.
- La capacidad de los métodos BDF- α y HHT- α para conservar la energía y el momento reduciendo el tamaño de paso y por tanto, reduciendo su capacidad para la eliminación de frecuencias altas. En este sentido, ambos métodos resultan menos eficientes que los métodos de la familia energía-momento que conservan estas propiedades con independencia del tamaño de paso y en particular, los métodos EDMC que conservan el momento y disipan las frecuencias altas. En esta línea y dada la estructura de los métodos EDMC, tampoco parece factible incorporar estas características en el método BDF- α .
- La capacidad y la eficiencia de la programación orientada a objeto (POO) para mantener, reutilizar y extender más posibilidades a los códigos que implementan los algoritmos de resolución, sin alterar desarrollos previos.

Las aportaciones de este trabajo son éstas:

- Hemos caracterizado la rigidez de las EDPs de difusión y de onda con Condiciones Iniciales (CI) y Condiciones de Contorno (CC) en 1D, en función del número de elementos de la semidiscretización, la longitud del dominio (L) y del parámetro α^2 (siendo éste la difusividad térmica en el caso de la EDP de difusión y el cuadrado

de la velocidad de propagación de la onda en la EDP de onda). También hemos demostrado cuál es la relación entre los autovalores de ambos problemas.

- Hemos presentado el estudio de precisión, estabilidad y amortiguamiento algorítmico de los métodos clásicos de la mecánica computacional (Newmark, HHT- α , etc.) para problemas de vibraciones y onda, en el marco de los métodos multipaso lineales para EDOs de orden 2. En paralelo, hemos realizado el mismo estudio de los métodos multipaso lineales para EDOs de orden 1 (BDF, EBDF, etc.) que también son aplicables a los problemas de vibraciones y onda reduciendo previamente el sistema de EDOs de orden 2 a orden 1.
- Hemos generado herramientas para la definición y representación gráfica de las regiones de estabilidad de los métodos numéricos, lo cual ha permitido el estudio comparativo de sus características.
- Hemos mostrado las razones por las que la *ode15s* no es eficiente al resolver sistemas rígidos de EDOs provenientes de la semidiscretización MEF de problemas de onda y vibraciones.
- Hemos detallado los mecanismos de control del paso adaptativo de la *ode45* y *ode15s*, observando que la estabilidad no está garantizada, y se han presentado ejemplos extremos con tamaños de paso que hacen que $\hat{h} = \lambda h$ esté fuera de la región de estabilidad y que degradan la precisión de los resultados.
- Hemos ampliado los métodos EBDF y MEBDF hasta orden 4, cambiando uno o ambos predictores BDF por NDFs. Se han creado los métodos EBNDF, ENBDF, ENDF y MEBNDF, MENBDF, MENDF. Para cada uno de ellos se ha hecho el estudio analítico de precisión y estabilidad, y hemos conseguido mantener o mejorar la estabilidad de los métodos de origen manteniendo el orden de precisión.
- Se ha desarrollado una metodología para la caracterización del amortiguamiento algorítmico de los problemas vibratorios, aplicable a los métodos generales para sistemas de EDOs de orden 1 (BDFs, etc.) y a los específicos para sistemas de EDOs de orden 2 (Newmark, HHT- α , Collocation, etc.).
- Hemos ampliado los métodos BDF introduciendo un parámetro de control del amortiguamiento algorítmico de los problemas vibratorios. Hemos creado el método BDF- α con el objeto de disipar las altas frecuencias de manera similar a como lo hace el método HHT- α . El método BDF- α surge al introducir un parámetro α en el método BDF2 y resulta de orden 2, incondicionalmente estable si $\alpha \geq -0,5$ y que permite el control paramétrico del radio espectral en el infinito, donde ρ_∞ toma todos los valores del intervalo $(0, 1]$, mientras que en el método HHT- α sólo lo hace en el intervalo $[0,5, 1]$.
- Hemos implementado un entorno para la resolución de EDOs utilizando las metodologías de la Programación Orientada a Objeto (POO), extendiendo un trabajo previo desde el MEF cuasi-estático al dinámico, que permite resolver del

orden de 4000 experimentos distintos (lineales y no lineales). Además, se trata de un entorno que permite extensiones futuras.

8.2. Futuras líneas de investigación

Teniendo en cuenta el recorrido realizado en esta tesis, las direcciones que podrían seguir los trabajos futuros derivados de la misma podrían ser las siguientes:

- Seguir estudiando si hay alguna posibilidad de generar métodos numéricos lineales multipaso de alta precisión y con buenas características de estabilidad:
 - utilizando el método BDF- α como predictor de los esquemas EBDF2 y MEBDF2.
 - utilizando el método A-BDF como predictor del esquema MEBDF. Obsérvese que el método A-EBDF en el que se utiliza el método A-BDF como predictor del esquema EBDF ya existe y que presenta buenas características.
 - utilizando NDFs en los esquemas E2BD y New SDMM que utilizan segundas derivadas dentro del esquema de los métodos extendidos.
 - creando una combinación lineal entre el método de Adams de orden 3 y el método BDF- α de orden 2 similar a los métodos *blended*.
- Intentar proponer un binomio similar al BDF2-BDF- α , pero de orden de precisión mayor que 2. Los métodos EBDF2 y MEBDF2 de orden de precisión 3 e incondicionalmente estables, podrían resultar buenos candidatos donde apoyarnos para la creación de un método EBDF2- α o MEBDF2- α capaz de parametrizar las altas frecuencias.
- Dado que la *ode15s* es de propósito general y por tanto conservadora en la adaptación del paso para garantizar precisión, se puede aprovechar la descripción detallada del Capítulo 4 del algoritmo adaptativo para estudiar su modificación cuando la *ode15s* se aplica a problemas vibratorios, de forma que el algoritmo adaptativo tienda al orden y al paso constante más conveniente para eliminar las frecuencias altas y la ineficiencia computacional.
- Seguir trabajando en el campo no lineal, incorporando en la metodología orientada a objeto métodos conservativos del tipo energía-momento para problemas elásticos, con el objeto de potenciar la experimentación y la comparación con los métodos HHT- α y BDF- α .

Apéndice A

Ecuación de onda estándar

La ecuación de onda estándar viene dada por la expresión:

$$\rho u_{tt}(x, t) = T u_{xx}(x, t) \quad (\text{A.1})$$

donde ρ es la densidad por unidad de longitud y T el tiro o fuerza axial constante, que por construcción debe mantener la cuerda tensionada para que introducida una perturbación inicial a la configuración recta de la cuerda, ésta tienda a recuperar su forma recta inicial y se produzca la vibración que la inercia de la masa (ρ) se encargará de mantener si no existe amortiguamiento.

Si T se sustituye por E (módulo elástico del material), entonces (A.1) representa las vibraciones longitudinales de una barra elástica y E proporciona la fuerza recuperadora elástica que se mantiene en equilibrio en cada sección de la barra con la fuerza de inercia. Pero toda esta descripción sólo tiene sentido en el caso de pequeñas vibraciones donde las deformadas prácticamente coinciden con la configuración rectilínea inicial de la cuerda.

Si la perturbación inicial es considerable con respecto a las dimensiones de la cuerda (por ejemplo, cuando el desplazamiento transversal del centro de la cuerda es una décima parte de su longitud), las deformaciones longitudinales provocadas por el desplazamiento transversal empiezan a ser considerables y la fuerza axial asociada resulta ser comparable con la del tiro inicial T , por lo que habrá que considerarla. Dada una sección recta diferencial de la cuerda situada en la posición x de longitud dx en su configuración recta inicial, si consideramos que la deformada transversal es $y(x)$, la nueva longitud diferencial tras la deformación será:

$$ds = \sqrt{dx^2 + dy^2} \Rightarrow \frac{ds}{dx} = \sqrt{\frac{dx^2 + dy^2}{dx^2}} = \sqrt{1 + (y'(x))^2} \quad (\text{A.2})$$

Si además, consideramos que el comportamiento del material es elástico lineal, la ecuación constitutiva que relaciona la fuerza F con la deformación para un elemento

longitudinal de longitud L , viene dada por la ley de Hooke:

$$\Delta L = \frac{F \cdot L}{E \cdot S} \quad (\text{A.3})$$

siendo E el módulo de Young del material y S el área de la sección recta. Entonces:

$$\begin{cases} \frac{\Delta [dx]}{dx} = \frac{ds-dx}{dx} = \sqrt{1 + (y'(x))^2} - 1 \\ \Delta [dx] = \frac{\bar{T}(x)dx}{E \cdot S} \end{cases} \Rightarrow \bar{T}(x) = E \cdot S \left(\sqrt{1 + (y'(x))^2} - 1 \right) \quad (\text{A.4})$$

siendo $\bar{T}(x)$ la fuerza axial en la sección x asociada a la deformación.

Teniendo en cuenta (A.4), la expresión (A.1) queda:

$$\rho u_{tt}(x, t) = \left(T + E \cdot S \left(\sqrt{1 + u_x^2(x, t)} - 1 \right) \right) u_{xx}(x, t) \quad (\text{A.5})$$

que es el modelo no lineal con el que trabajaremos a continuación.

Efectuando la semidiscretización MEF, el sistema de EDOs resultante es:

$$\sum_{j=2}^{n-1} \left[\int_0^L N_i(x) \rho N_j(x) dx \right] d_j''(t) = - \sum_{j=2}^{n-1} \left[\int_0^L N_i'(x) \left(T + \bar{T}(u^h) \right) N_j'(x) dx \right] d_j(t)$$

Que matricialmente se escribe:

$$M \mathbf{d}''(t) = -K(\mathbf{d}) \mathbf{d}(t) \quad (\text{A.6})$$

Convertimos el sistema de EDOs de orden 2 (A.6) en un sistema de EDOs de orden 1:

$$\underbrace{\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}}_{\hat{M}} \mathbf{y}' = \underbrace{\begin{pmatrix} 0 & I \\ -K(\mathbf{y}) & 0 \end{pmatrix}}_{\hat{K}} \mathbf{y} \Rightarrow \hat{M} \mathbf{y}' = \hat{K}(\mathbf{y}) \mathbf{y} \quad (\text{A.7})$$

siendo $\mathbf{y} = \begin{pmatrix} \mathbf{d} \\ \mathbf{d}' \end{pmatrix}$.

Para resolver este sistema de EDOs mediante la *ode15s* deberemos proporcionarle al inicio la matriz de masa \hat{M} y durante el cálculo una función que calcule el 2º miembro del sistema: $\hat{K}(\mathbf{y}) \mathbf{y}$ para los distintos valores de \mathbf{y} .

Otra opción es programar un algoritmo de resolución, por ejemplo el trapezoidal, que es incondicionalmente estable y por tanto apropiado para los sistemas rígidos procedentes de la semidiscretización MEF. En Matlab se puede implementar con sencillez un algoritmo completo donde cada paso se resuelve iterativamente por el método de Newton Raphson, incluyendo una función para calcular las matrices jacobianas por diferencias finitas de los residuos.

Apéndice B

Autovalores de las EDP lineales de difusión y onda

B.1. Difusión

Consideramos la ecuación de difusión 1D dada por la EDP (1.22):

$$\begin{cases} u_t = \alpha^2 u_{xx} \\ CC : u(0, t) = 0 = u(L, t) \\ CI : u(x, 0) = g(x) \end{cases} \quad (\text{B.1})$$

siendo $\alpha^2 = \frac{k}{c\rho}$ la difusividad térmica, k la conductividad termal, c la capacidad termal y ρ la densidad.

En el Capítulo 1 hemos visto que la semidiscretización MEF de la EDP (B.1) con n nodos y $(n - 1)$ elementos conduce a un sistema de EDOs que en la forma $y' = f(t, y)$ viene dada por:

$$\mathbf{d}'(t) = -M^{-1}K\mathbf{d}(t), \quad \mathbf{d}(0) = \mathbf{d}_0 = (g(x_2), \dots, g(x_{n-1}))^T \quad (\text{B.2})$$

En este apéndice estudiamos la rigidez del problema (B.1), es decir los autovalores de la matriz jacobiana $\frac{\partial \mathbf{f}}{\partial \mathbf{d}} = -M^{-1}K$, en función de tres variables: el número de elementos de la semidiscretización, la longitud de la varilla L y la difusividad térmica.

B.1.1. Rigidez en función del n° de elementos de la semidiscretización

Hemos considerado los materiales epoxi y epoxi alúmina (100/188) cuyas difusividades térmicas son $\alpha^2 = 0,129 \cdot 10^{-2} \text{ cm}^2/s$ y $\alpha^2 = 0,449 \cdot 10^{-2} \text{ cm}^2/s$ respectivamente. Fijados α^2 y la longitud de la varilla ($L = 20\text{cm}$) hacemos variar el número de elementos

y analizamos lo que les sucede a los autovalores máximo y mínimo en módulo, ver Tabla B.1. Los autovalores máximos de cada caso los hemos representado en la Figura B.1.

Nº de elementos	$ \lambda_{max} $ epoxi	$ \lambda_{min} $ epoxi	$ \lambda_{max} $ epoxi alúm.	$ \lambda_{min} $ epoxi alúm.
5	$7,3478 \cdot 10^{-4}$	$3,2890 \cdot 10^{-5}$	$2,5575 \cdot 10^{-3}$	$1,1448 \cdot 10^{-4}$
10	$3,5991 \cdot 10^{-3}$	$3,2092 \cdot 10^{-5}$	$1,2527 \cdot 10^{-2}$	$1,1170 \cdot 10^{-4}$
20	$1,5198 \cdot 10^{-2}$	$3,1895 \cdot 10^{-5}$	$5,2897 \cdot 10^{-2}$	$1,1101 \cdot 10^{-4}$
40	$6,1635 \cdot 10^{-2}$	$3,1846 \cdot 10^{-5}$	$2,1453 \cdot 10^{-1}$	$1,1084 \cdot 10^{-4}$
80	$2,4739 \cdot 10^{-1}$	$3,1834 \cdot 10^{-5}$	$8,6108 \cdot 10^{-1}$	$1,1080 \cdot 10^{-4}$
160	$9,9043 \cdot 10^{-1}$	$3,1830 \cdot 10^{-5}$	$3,4473 \cdot 10^0$	$1,1079 \cdot 10^{-4}$
320	$3,9626 \cdot 10^0$	$3,1830 \cdot 10^{-5}$	$1,3792 \cdot 10^1$	$1,1079 \cdot 10^{-4}$
640	$1,5851 \cdot 10^1$	$3,1830 \cdot 10^{-5}$	$5,5172 \cdot 10^1$	$1,1079 \cdot 10^{-4}$
1280	$6,3406 \cdot 10^1$	$3,1829 \cdot 10^{-5}$	$2,2069 \cdot 10^2$	$1,1079 \cdot 10^{-4}$

Tabla B.1: Autovalores EDP difusión epoxi y epoxi alúm. en función del nº de elem.

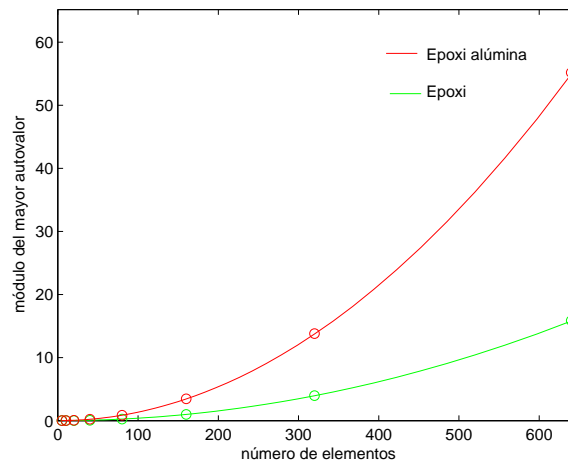


Figura B.1: Mayor autov. en módulo en función del nº elem. con epoxi, epoxi alúmina.

Conclusiones:

- En cada uno de los materiales, al aumentar el número de elementos de la semidiscretización aumenta el autovalor más grande y queda igual el autovalor mínimo.
- Si tenemos una semidiscretización de m elementos y lo multiplicamos por n , obteniendo así una nueva semidiscretización de $n \cdot m$ elementos, el autovalor más grande de la segunda semidiscretización es el autovalor máximo de la 1ª semidiscretización multiplicado por n^2 . Por ejemplo, en el material epoxi para 10 elementos

$|\lambda_{max}| = 3,5991 \cdot 10^{-3}$. Y si multiplicamos por 2 el número de elementos, pasando a tener una semidiscretización de 20 elementos, el autovalor más grande se multiplica por 2^2 : $|\lambda_{max}| = 4 \cdot 3,5991 \cdot 10^{-3} = 1,4396 \cdot 10^{-2} \approx 1,5198 \cdot 10^{-2}$.

B.1.2. Rigidez en función de la longitud de la varilla L

Fijados α^2 y el número de elementos de la semidiscretización, hacemos variar la longitud de la varilla y analizamos lo que les sucede a los autovalores máximo y mínimo en módulo. Para el estudio se han considerado los materiales epoxi y epoxi alúmina y una semidiscretización de 20 elementos. En la Tabla B.2 hemos recogido los valores de los autovalores máximo y mínimo en ambos materiales.

Longitud de la varilla (cm)	$ \lambda_{max} $ epoxi	$ \lambda_{min} $ epoxi	$ \lambda_{max} $ epoxi alúm.	$ \lambda_{min} $ epoxi alúm.
0,02	$1,5198 \cdot 10^4$	$3,1895 \cdot 10^1$	$5,2897 \cdot 10^4$	$1,1101 \cdot 10^2$
0,2	$1,5198 \cdot 10^2$	$3,1895 \cdot 10^{-1}$	$5,2897 \cdot 10^2$	$1,1101 \cdot 10^0$
2	$1,5198 \cdot 10^0$	$3,1895 \cdot 10^{-3}$	$5,2897 \cdot 10^0$	$1,1101 \cdot 10^{-2}$
20	$1,5198 \cdot 10^{-2}$	$3,1895 \cdot 10^{-5}$	$5,2897 \cdot 10^{-2}$	$1,1101 \cdot 10^{-4}$
200	$1,5198 \cdot 10^{-4}$	$3,1895 \cdot 10^{-7}$	$5,2897 \cdot 10^{-4}$	$1,1101 \cdot 10^{-6}$
2000	$1,5198 \cdot 10^{-6}$	$3,1895 \cdot 10^{-9}$	$5,2897 \cdot 10^{-6}$	$1,1101 \cdot 10^{-8}$
20000	$1,5198 \cdot 10^{-8}$	$3,1895 \cdot 10^{-11}$	$5,2897 \cdot 10^{-8}$	$1,1101 \cdot 10^{-10}$

Tabla B.2: Autovalores EDP difusión para epoxi y epoxi alúm. en función de L.

Conclusiones:

- Al disminuir la longitud de la varilla el autovalor más grande (y todos ellos) aumentan.
- Si dada una varilla de longitud l , ésta la multiplicamos por x teniendo así una varilla de longitud de $l \cdot x$, los autovalores se multiplican por $\frac{1}{x^2}$.

B.1.3. Rigidez en función de la difusividad térmica

Fijados el número de elementos de la semidiscretización y la longitud de la varilla, si utilizamos difusividades térmicas α_1^2 y α_2^2 relacionadas de manera que $\alpha_1^2 = k \cdot \alpha_2^2$, las matrices resultantes $-M_1^{-1}K_1$ y $-M_2^{-1}K_2$ también cumplen la misma relación de proporcionalidad:

$$-M_1^{-1}K_1 = k \cdot (-M_2^{-1}K_2) \quad (\text{B.3})$$

Veremos cuál es la relación entre los autovalores de dos matrices A y B proporcionales. Dada la matriz A denotaremos por $\chi_A(\lambda) = \det(\lambda I - A)$ su polinomio característico.

Teorema: Sea A una matriz cuadrada de dimensión n y $B = k \cdot A$, matriz que se consigue multiplicando A por un escalar $k \neq 0$. Entonces, los polinomios característicos de ambas matrices están relacionados de la siguiente manera:

$$\chi_B(\lambda) = k^n \chi_A \left(\frac{\lambda}{k} \right) \quad (\text{B.4})$$

Como consecuencia, los autovalores de B son los autovalores de A multiplicados por el escalar k .

Demostración: Por la definición de polinomio característico tenemos:

$$\chi_B(\lambda) = \det(\lambda I - B) = \begin{vmatrix} \lambda - b_{11} & -b_{12} & \cdots & -b_{1n} \\ -b_{21} & \lambda - b_{22} & \cdots & -b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -b_{n1} & -b_{n2} & \cdots & \lambda - b_{nn} \end{vmatrix} \quad (\text{B.5})$$

siendo: $b_{ij} = k \cdot a_{ij}$.

Si dividimos cada una de la filas del determinante (B.5) entre la constante k , el valor del determinante queda dividido entre k^n :

$$\chi_B(\lambda) = k^n \begin{vmatrix} \frac{\lambda - b_{11}}{k} & \frac{-b_{12}}{k} & \cdots & \frac{-b_{1n}}{k} \\ \frac{-b_{21}}{k} & \frac{\lambda - b_{22}}{k} & \cdots & \frac{-b_{2n}}{k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-b_{n1}}{k} & \frac{-b_{n2}}{k} & \cdots & \frac{\lambda - b_{nn}}{k} \end{vmatrix} = k^n \begin{vmatrix} \frac{\lambda}{k} - a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \frac{\lambda}{k} - a_{22} & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \frac{\lambda}{k} - a_{nn} \end{vmatrix} \quad (\text{B.6})$$

De (B.6) se tiene:

$$\chi_B(\lambda) = k^n \det \left(\frac{\lambda}{k} I - A \right) = k^n \chi_A \left(\frac{\lambda}{k} \right) \Rightarrow \chi_B(\lambda) = k^n \chi_A \left(\frac{\lambda}{k} \right) \quad (\text{B.7})$$

Por lo que queda demostrada la igualdad (B.4). \square

Una vez demostrada la igualdad (B.4), obtenemos la siguiente cadena de equivalencias:

$$\lambda \text{ autovalor de } B \Leftrightarrow \chi_B(\lambda) = 0 \Leftrightarrow \chi_A \left(\frac{\lambda}{k} \right) = 0 \Leftrightarrow \frac{\lambda}{k} \text{ autovalor de } A \quad (\text{B.8})$$

Así, si λ_j es autovalor de A con multiplicidad p , $\lambda_j \cdot k$ es autovalor de B de multiplicidad p . \square

Para realizar este análisis hemos considerado una varilla de longitud $L = 20\text{cm}$ de cobre y otra de epoxi alúmina con difusividades térmicas: $\alpha_{ep.al.}^2 = 0,449 \cdot 10^{-2} \text{ cm}^2/\text{s}$ y $\alpha_{cobre}^2 = 1,0745 \text{ cm}^2/\text{s}$. La relación entre ambas difusividades es:

$$k = \alpha_{cobre}^2 / \alpha_{ep.al.}^2 = 2,39 \cdot 10^2 \quad (\text{B.9})$$

Nº de elementos	$ \lambda_{max} $ epoxi alúm.	$ \lambda_{min} $ epoxi alúm.	$ \lambda_{max} $ cobre	$ \lambda_{min} $ cobre
5	$2,5575 \cdot 10^{-3}$	$1,1448 \cdot 10^{-4}$	$6,1203 \cdot 10^{-1}$	$2,7395 \cdot 10^{-2}$
10	$1,2527 \cdot 10^{-2}$	$1,1170 \cdot 10^{-4}$	$2,9979 \cdot 10^0$	$2,6731 \cdot 10^{-2}$
20	$5,2897 \cdot 10^{-2}$	$1,1101 \cdot 10^{-4}$	$1,2659 \cdot 10$	$2,6567 \cdot 10^{-2}$
40	$2,1453 \cdot 10^{-1}$	$1,1084 \cdot 10^{-4}$	$5,1338 \cdot 10$	$2,6526 \cdot 10^{-2}$
80	$8,6108 \cdot 10^{-1}$	$1,1080 \cdot 10^{-4}$	$2,0607 \cdot 10^2$	$2,6516 \cdot 10^{-2}$
160	$3,4473 \cdot 10^0$	$1,1079 \cdot 10^{-4}$	$8,2498 \cdot 10^2$	$2,6513 \cdot 10^{-2}$
320	$1,3792 \cdot 10$	$1,1079 \cdot 10^{-4}$	$3,3006 \cdot 10^3$	$2,6512 \cdot 10^{-2}$
640	$5,5172 \cdot 10$	$1,1079 \cdot 10^{-4}$	$1,3203 \cdot 10^4$	$2,6512 \cdot 10^{-2}$
1280	$2,2069 \cdot 10^2$	$1,1079 \cdot 10^{-4}$	$5,2814 \cdot 10^4$	$2,6512 \cdot 10^{-2}$

Tabla B.3: Autovalores EDP difusión epoxi alúm. y cobre en función del nº de elem.

Así, hemos comprobado experimentalmente que (B.4) se cumple. Es decir, que multiplicando los autovalores del material epoxi por la constante k se consiguen los autovalores del cobre, ver Tabla B.3.

Conclusiones:

- Cuanto más grande es la difusividad térmica, mayores son los autovalores del problema.
- Dado un valor de α^2 , si éste lo multiplicamos por k , los autovalores del nuevo sistema también se multiplican por k . Como la relación entre las difusividades térmicas de los materiales epoxi alúmina y cobre es (B.9), la relación que existe entre los autovalores de las matrices jacobianas correspondientes es la misma:

$$\begin{cases} |\lambda_{max,cobre}| = 2,3931 \cdot 10^2 \cdot |\lambda_{max,ep.al.}| \\ |\lambda_{min,cobre}| = 2,3931 \cdot 10^2 \cdot |\lambda_{min,ep.al.}| \end{cases} \quad (B.10)$$

B.2. Onda

Consideramos la ecuación de onda 1D dada por la EDP (1.23):

$$\begin{cases} u_{tt} = \alpha^2 u_{xx} \\ CC : u(0, t) = 0 = u(L, t) \\ CI : u(x, 0) = g(x), u_t(x, 0) = 0 \end{cases} \quad (B.11)$$

siendo $\alpha = \sqrt{\frac{T}{\rho}}$ la velocidad de propagación de la onda, T el tiro y ρ la densidad lineal de masa.

En el Capítulo 1 también hemos visto que la semidiscretización MEF de la EDP (B.11) con n nodos y $(n - 1)$ elementos conduce al siguiente sistema de EDOs que en la forma $y' = f(t, y)$ viene dada por:

$$\begin{pmatrix} \mathbf{y}'_1(t) \\ \mathbf{y}'_2(t) \end{pmatrix} = \begin{pmatrix} 0 & I \\ -M^{-1}K & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y}_1(t) \\ \mathbf{y}_2(t) \end{pmatrix}, \quad (\text{B.12})$$

siendo: $\begin{cases} \mathbf{y}_1(t) = \mathbf{d}(t), \mathbf{y}_2(t) = \mathbf{d}'(t) \\ \mathbf{y}_1(0) = \mathbf{d}_0, \mathbf{y}_2(0) = \mathbf{d}'_0 \end{cases}$, y la matriz jacobiana $\frac{\partial \mathbf{f}}{\partial \mathbf{d}} = \begin{pmatrix} 0 & I \\ -M^{-1}K & 0 \end{pmatrix}$.

Hemos estudiado los autovalores de la matriz jacobiana de (B.11):

- Hemos demostrado la relación que existe entre los autovalores de las matrices jacobianas de la EDP de difusión y onda.
- Y utilizando la relación entre los autovalores de las dos matrices jacobianas y los resultados que se han obtenido para la EDP de difusión, hemos deducido la rigidez del problema (B.11) en función del número de elementos de la semidiscretización, la longitud de la cuerda L y la velocidad de propagación de la onda.

B.2.1. Relación entre los autovalores del problema de difusión y onda

Las matrices jacobianas de las EDP de difusión y onda son:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{d}} = A, \quad \frac{\partial \mathbf{f}}{\partial \mathbf{d}'} = \begin{pmatrix} 0 & I \\ A & 0 \end{pmatrix} \quad (\text{B.13})$$

siendo $A = -M^{-1}K$. En el siguiente teorema demostraremos cuál es la relación entre los autovalores de dos matrices de la forma (B.13).

Teorema: Sea A una matriz cuadrada y $B = \begin{pmatrix} 0 & I \\ A & 0 \end{pmatrix}$. Entonces, los polinomios característicos de ambas matrices cumplen:

$$\chi_B(\lambda) = \chi_A(\lambda^2) \quad (\text{B.14})$$

En consecuencia, los autovalores de B son las raíces cuadradas de los autovalores de A .

Demostración: El polinomio característico de B viene dado por:

$$\chi_B(\lambda) = \det \begin{pmatrix} \lambda I & -I \\ -A & \lambda I \end{pmatrix} \quad (\text{B.15})$$

Supongamos que la matriz A es una matriz cuadrada de dimensión n y que la matriz B es de dimensión $2n \times 2n$. Si en el determinante (B.15) a la columna i -ésima le sumamos

la columna $(n+i)$ -ésima multiplicada por λ , para $i = 1, 2, 3, \dots, n$, obtenemos el siguiente determinante:

$$\chi_B(\lambda) = \det \begin{pmatrix} 0 & -I \\ C & \lambda I \end{pmatrix} \quad (\text{B.16})$$

siendo:

$$C = \begin{pmatrix} -a_{11} + \lambda^2 & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & -a_{22} + \lambda^2 & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & -a_{nn} + \lambda^2 \end{pmatrix} = \lambda^2 I - A \quad (\text{B.17})$$

Teniendo en cuenta (B.17), la expresión (B.16) queda:

$$\chi_B(\lambda) = \det \begin{pmatrix} 0 & -I \\ \lambda^2 I - A & \lambda I \end{pmatrix} \quad (\text{B.18})$$

Desarrollaremos el determinante (B.18) empezando por la primera fila. Todos los elementos de esa fila son 0, menos el de la posición $(n+1)$ que es -1 . Después seguiremos el desarrollo por la segunda fila y seguiremos así hasta desarrollarlo por la fila n -ésima. Entonces, se consigue:

$$\begin{aligned} \chi_B(\lambda) &= (-1)^{1+(n+1)}(-1)(-1)^{2+(n+2)}(-1)\dots(-1)^{n+2n}(-1)\det(\lambda^2 I - A) \\ &= (-1)^n(-1)(-1)^n(-1)\dots(-1)^n(-1)\det(\lambda^2 I - A) \\ &= (-1)^{n^2+n}\det(\lambda^2 I - A) = \det(\lambda^2 I - A) = \chi_A(\lambda^2) \end{aligned} \quad (\text{B.19})$$

Durante la demostración (B.19) hay que tener en cuenta que $n^2 + n = n(n+1)$ es par.

Una vez demostrado (B.14), obtenemos la siguiente cadena de equivalencias:

$$\lambda \text{ autovalor de } B \Leftrightarrow \chi_B(\lambda) = 0 \Leftrightarrow \chi_A(\lambda^2) = 0 \Leftrightarrow \lambda^2 \text{ autovalor de } A \quad (\text{B.20})$$

Así, si λ_j autovalor de A y p su multiplicidad, entonces, $\pm\sqrt{\lambda_j}$ es autovalor de B de multiplicidad p .

En la Tabla B.4 se han recogido los autovalores máximos y mínimos de las EDPs de difusión y onda cuando $\alpha^2 = 1$ y considerando $L = 8$. Se cumple que los autovalores de la EDP de onda son la raíz cuadrada de los autovalores de la EDP de difusión.

Nº de elementos	λ_{max} difusión	λ_{min} difusión	λ_{max} onda	λ_{min} onda
50	$-4,6737 \cdot 10^2$	$-1,5426 \cdot 10^{-1}$	$\pm 2,1619 \cdot 10^1 i$	$\pm 3,9270 \cdot 10^{-1} i$
200	$-7,4986 \cdot 10^3$	$-1,5422 \cdot 10^{-1}$	$\pm 8,6595 \cdot 10^1 i$	$\pm 3,9270 \cdot 10^{-1} i$
800	$-1,2000 \cdot 10^5$	$-1,5421 \cdot 10^{-1}$	$\pm 3,4641 \cdot 10^2 i$	$\pm 3,9270 \cdot 10^{-1} i$

Tabla B.4: Autovalores EDP difusión y onda, con $\alpha^2 = 1$ y $L = 8$.

B.2.2. Rigidez en función del nº de elementos de la semidiscretización

Fijados α^2 y la longitud de la cuerda, hacemos variar el número de elementos y analizamos lo que les sucede a los autovalores máximo y mínimo en módulo. Hemos calculado los autovalores máximo y mínimo de una cuerda de guitarra (considerando el problema como si fuese lineal), con una cuerda de longitud $L = 64,8cm$, y hemos considerado la quinta cuerda de la misma que para dar la nota LA necesita una frecuencia de $110Hz$ [29], ver Tabla B.5. La relación entre el parámetro α^2 de la expresión (B.11) y la frecuencia viene dada por la siguiente fórmula:

$$\omega_n = \frac{n\pi\alpha}{L} = \frac{n\pi}{L} \sqrt{\frac{T}{\rho}} \Rightarrow \alpha = \frac{\omega_n L}{n\pi} \quad (\text{B.21})$$

Nº de elementos	$ \lambda_{max} $	$ \lambda_{min} $
50	$2,3986 \cdot 10^5$	$4,3577 \cdot 10^3$
100	$4,8020 \cdot 10^5$	$4,3572 \cdot 10^3$
200	$9,6101 \cdot 10^5$	$4,3571 \cdot 10^3$
400	$1,9221 \cdot 10^6$	$4,3570 \cdot 10^3$
800	$3,8450 \cdot 10^6$	$4,3570 \cdot 10^3$

Tabla B.5: Autovalores EDP onda para dar la nota LA.

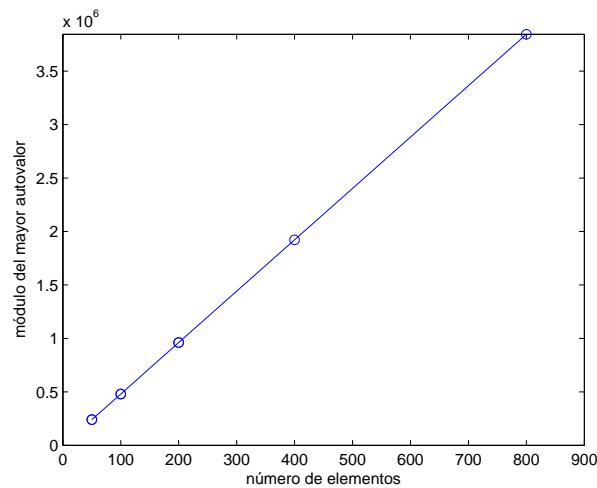


Figura B.2: Mayor autovalor de la EDP onda en función del nº elem, nota LA.

Conclusiones:

- Al aumentar el número de elementos aumenta el autovalor más grande del sistema.

- Si dada una semidiscretización de m elementos, ésta cantidad la multiplicamos por n , teniendo así una semidiscretización de $n \cdot m$ elementos, el autovalor más grande queda multiplicado por n .

B.2.3. Rigidez en función de la longitud de la cuerda L

Fijados α^2 y el número de elementos de la semidiscretización, hacemos variar la longitud de la cuerda y analizamos lo que les sucede a los autovalores máximo y mínimo en módulo. Teniendo en cuenta la relación entre los autovalores de las matrices jacobianas de las EDP de difusión y onda y las conclusiones de la Sección B.1.2 se tiene:

- Al disminuir la longitud de la cuerda los autovalores aumentan.
- Si dada una cuerda de longitud l , ésta la multiplicamos por x teniendo así una cuerda de longitud $l \cdot x$, los autovalores se multiplican por $\frac{1}{x}$.

B.2.4. Rigidez en función de la velocidad de propagación

Fijados la longitud de la cuerda y el número de elementos de la semidiscretización, hacemos variar el parámetro α^2 . Teniendo en cuenta la relación entre los autovalores de las matrices jacobianas de las EDP de difusión y onda y las conclusiones de la Sección B.1.3, se tiene:

- Al aumentar el valor de α^2 , aumentan los autovalores.
- Dado un valor de α^2 , si éste lo multiplicamos por k , los autovalores quedan multiplicados por \sqrt{k} .

Apéndice C

Datos de las funciones de los objetos

Objeto donde aparece	Nombre del dato	Significado
Objeto método elementos finitos (<i>objmef</i>)	<i>objmetodo</i> <i>objprob</i> <i>objmalla</i>	Objeto método. Objeto problema. Objeto malla.
Objeto sistema lineal de ecuaciones (<i>sislin</i>)	K F u	Matriz K del sistema lineal $KU = F$. Matriz F del sistema lineal $KU = F$. Incógnita nodal U del sistema lineal $KU = F$.
Objeto problema (<i>objprob</i>)	<i>ngdl</i> <i>gdla</i> <i>ngdla</i> <i>tension</i> (o <i>flujos</i>) <i>objresYcc</i>	Nº de grados de libertad. Grados de libertad activos. Nº de grados de libertad activos. Tensiones σ (o flujos ϕ). Objeto restricciones y Cond. Contorno (CC).
Objeto restricciones y condiciones de contorno (<i>objresYcc</i>)	<i>ccd</i> <i>valccd</i> <i>ccn</i> <i>valccn</i>	CC Dirichlet. Valores de CCD. CC Neumann. Valores CCN.
Objeto malla (<i>objmalla</i>)	<i>nodos</i> <i>conec</i> <i>objeief</i> <i>objmat</i>	Coordenadas nodales. Conectividad de los elementos. Objeto elemento finito. Objeto material.
Objeto elemento finito (<i>objeief</i>)	<i>nodos_e</i> <i>conec_e</i> <i>u_e</i> <i>xint</i> N ND <i>detjacobpesos</i> <i>objeiep</i>	Coord. nodales del elemento. Conectividad del elemento. Incógnita nodal u en el elemento. Coordenadas de los puntos de integración. Funciones de forma en el elemento. Derivadas de las funciones de forma en elem. Pesos del det. jac. en los puntos de integr. Objeto elemento patrón.

Continúa en la siguiente página

Tabla C.1 – viene de la página anterior

Objeto donde aparece	Nombre del dato	Significado
	<i>objpint</i>	Objeto punto de integración.
Objeto elemento patrón (<i>objelep</i>)	<i>cint</i> <i>NP</i> <i>NDP</i> <i>pesos</i>	Coordenadas puntos de integr. elem. patrón. Funciones de forma en el elemento patrón. Derivadas de las funciones <i>NP</i> . Pesos de la integración numérica de Gauss.
Objeto punto de integración (<i>objpint</i> en Laplace)	<i>NI</i> <i>NDI</i> <i>u</i> <i>pint</i>	Funciones de forma reales. Derivadas de las funciones de forma reales. Valor de la incógnita <i>u</i> en el pto. de integr. Nº o índice del pto. de integración.
Objeto material (caso <i>elastin_lm</i>)	<i>E</i> <i>Nu</i> <i>Landa</i> <i>Mu</i> <i>Kapa</i> <i>f</i> <i>ro</i> <i>dilat</i> <i>d</i>	Módulo de Young. Módulo de Poisson. Coeficiente de Lamé λ . Coeficiente de Lamé μ . Coeficiente volumétrico. Fuerzas de volumen. Densidad. Coeficiente de dilatación. Amortiguamiento.
Objeto método Newton Raphson (<i>objnr</i>)	<i>R</i> <i>Fint</i> <i>Fext</i> <i>u</i> <i>tol</i> <i>maxiter</i> <i>objJ</i>	Residuo del mét. Newton Raphson. Fuerzas internas F_{int} . Fuerzas externas F_{ext} . Valor de la incógnita <i>u</i> . Tolerancia establecida. Número máximo de iteraciones. Objeto Jacobiano.
Objeto jacobiano (<i>objJ</i>)	<i>J</i>	Jacobiano
Objeto problema (<i>objprob</i>)	<i>tension</i> <i>temp</i> <i>tensele_n</i> <i>tensele_k</i>	Tensión. Temperatura. Tensión en los ptos. de integr. paso <i>n</i> . Tensión ptos. de integr. paso <i>n</i> , iteración <i>k</i> .
Objeto punto de integración (caso <i>pelas_inc</i>)	<i>N</i> <i>ND</i> <i>u</i> <i>num_ele</i> <i>num_pint</i> <i>objdef</i> <i>tipoelas</i>	Funciones de formas reales. Derivadas de las funciones de forma reales. Valor de la incógnita <i>u</i> en el pto. de integr. Nº o índice del elemento. Nº o índice del pto. de integración. Objeto deformación. Tipo de elasticidad.
Objeto deformación (caso <i>deflin</i>)	<i>def</i> <i>deft</i> <i>objprob</i>	Deformación total. Deformación térmica. Objeto problema.
Objeto deformación (caso <i>def_lc</i>)	<i>def</i> <i>J</i> <i>nodos0</i> <i>objdef</i>	Deformación total. Jacobiano. Coord. de los nodos en el instante inicial. Objeto deformación.
Objeto material (caso <i>plasxy</i>)	<i>sig-y</i> <i>H</i>	Tensión de fluencia. Coeficiente de endurecimiento.

Continúa en la siguiente página

Tabla C.1 – viene de la página anterior

Objeto donde aparece	Nombre del dato	Significado
	<i>beta</i> <i>H1</i> <i>H2</i> <i>factor</i> <i>tipotang</i> <i>Cvol</i> <i>Cdesv</i> <i>varintn</i> <i>varintk</i>	Coefficiente cinemático. Coefficiente endurecimiento isótropo. Coefficiente endurecimiento cinemático. $\frac{2\mu}{2\mu+2/3H}$ Tipo módulo plástico (continuo o consistente). Parte volumétrica del tensor de elasticidad. Parte desviadora del tensor de elasticidad. Variables internas en el paso n . Variables internas en el paso n , iteración k .
Objeto método ode (1) <i>base_met</i>	<i>objfty</i> <i>interv</i> <i>yini</i> <i>npasos</i> <i>h</i> <i>tsol</i> <i>ysol</i> <i>dat</i>	Objeto fty. Intervalo de integración temporal $T = [a, b]$. Valor inicial $y(a) = y_0$. Número de pasos. Tamaño de paso $h = \Delta t$. Valores t_n donde se calcula y_n . Solución numérica en instante t_n . Conjunto de datos.
Objeto <i>odesuite</i>	<i>metJ</i>	Forma en la que se calcula el jacobiano: analíticamente, por diferencias finitas o indicando si el problema es lineal de coeficientes constantes.
Objeto método trapezoidal (<i>objtrap</i>)	<i>M</i> <i>fk</i>	Matriz de masa. Valor $f(t_n, y_n)$ al dar el paso de t_n a t_{n+1} .
Objeto método trapezoidal lineal (<i>objtraplinc</i>)	<i>C1</i> <i>C2</i> <i>gk</i>	Matriz C_1 de la expresión (7.55). Matriz C_2 de la expresión (7.55). Valor $g(t_n)$ de (7.55).
Objeto alfa lineal y no lineal (<i>objalfalinc</i> y <i>objalfa</i>)	<i>alfa</i> (lin.+no lin.) <i>beta</i> (lin.+no lin.) <i>gamma</i> (lin.+no lin.) <i>M</i> (lin.+no lin.) <i>K</i> (lin.+no lin.) <i>dini</i> (en no lineal) <i>vini</i> (en no lineal) <i>vn</i> (lin.+no lin.) <i>an</i> (lin.+no lin.)	Parámetro α del método HHT- α . Parámetro β del método HHT- α . Parámetro γ del método HHT- α . Matriz de masa. Matriz de rigidez. Valor de d en el instante inicial $d(0) = d_0$. Valor de v en el instante inicial $d'(0) = v_0$. Valor de v en el instante t_n $d'(t_n) = v_n$. Valor de a en el instante t_n $d''(t_n) = a_n$.
Objeto cuasi (<i>objcuasi</i>)	<i>xini</i> <i>paso_ad</i>	Prueba inicial para el método de Newton. Paso adaptativo.
Objeto bdf lineal y no lineal (<i>objbdfinc</i> y <i>objbdf</i>)	<i>M</i> (en no lineal) <i>C1</i> (en lineal) <i>alfas</i> (lin.+no lin.) <i>C2</i> (lin.+no lin.) <i>ykm1</i> (lin.+no lin.)	Matriz de masa. Matriz C_1 de (7.64). Matriz de coeficientes $\hat{\alpha}_j$ del mét. BDF. Matriz C_2 dada por (7.60). Matriz Y_{n+k-1} dada por (7.60).
Objeto ndf lineal y no lineal (<i>objndfinc</i> y <i>objndf</i>)	<i>kappa</i> <i>gamma</i> <i>C4</i>	Constantes κ de los mét. NDF. Constantes γ_k de los mét. NDF. Matriz C_4 dada por (7.69).

Continúa en la siguiente página

Tabla C.1 – viene de la página anterior

Objeto donde aparece	Nombre del dato	Significado
Objeto bdf alfa lineal y no lineal (<i>objbdfalfalinc</i> y <i>objbdfalfa</i>)	M (en no lineal) fk (en no lineal) C_1 (en lineal) gk (en lineal) C_2 (lin.+no lin.) $beta$ (lin.+no lin.) $alfa$ (lin.+no lin.) $ykm1$ (lin.+no lin.)	Matriz de masa. Valor $f(t_n, y_n)$ al dar el paso de t_n a t_{n+1} . Matriz C_1 dada por (7.85). Valor $g(t_{n+1})$ de (7.82). Matriz C_2 dada por (7.85). Matriz que contiene los valores β_i de (7.77). Constante α del método BDF- α . Matriz Y_{n+1} dada por (7.85).
Objetos puntos super-futuros lineal y no lineal	AA BB C_2	Matriz de coeficientes α_j del mét. EBDF. Matriz de coeficientes $(\beta_k \beta_{k+1})$ EBDF. Matriz C_2 dada por (7.88).
Objeto ebdf no lineal (<i>objebdf</i>)	M $ykm1$ $fbk1$ $fbk2$ <i>objbdf</i>	Matriz de masa. Matriz Y_{n+k-1} dada por (7.88). Valor $\bar{f}_{n+k} = f(t_{n+k}, \bar{y}_{n+k})$ de (7.96). Valor $\bar{f}_{n+k+1} = f(t_{n+k+1}, \bar{y}_{n+k+1})$ de (7.96). Objeto bdf.
Objeto ebdf lineal (<i>objebdflinec</i>)	C_1 $ykm1$ $gk1$ <i>objbdflinec</i>	Matriz C_1 de la expresión (7.92). Y_{n+k-1} dada por (7.88). Valor $g(t_{n+k})$ de (7.91). Objeto bdf lineal.
Objeto ebndf y enbdf no lineales (<i>objebndf</i> y <i>objenbdf</i>)	M $ykm1$ $fbk1$ $fbk2$ <i>objbdf</i> <i>objndf</i>	Matriz de masa. Matriz Y_{n+k-1} dada por (7.88). $\bar{f}_{n+k} = f(t_{n+k}, \bar{y}_{n+k})$ $\bar{f}_{n+k+1} = f(t_{n+k+1}, \bar{y}_{n+k+1})$ Objeto bdf. Objeto ndf.
Objeto mebdf lineal y no lineal	BG	Coefficientes $\hat{\beta}_k$ del mét. BDF.
Objeto ebndf y enbdf lineales (<i>objebndflinec</i> y <i>objenbdflinec</i>)	C_1 $ykm1$ $gk1$ <i>objbdflinec</i> <i>objndflinec</i>	Matriz C_1 de la expresión (7.92). Matriz Y_{n+k-1} dada por (7.88). $g(t_{n+k})$ de (7.91). Objeto bdf lineal. Objeto ndf lineal.

Apéndice D

Programación Objeto

D.1. Objeto métodos multipaso no lineales

Funciones del objeto mp: *pasos_ini*, *actpaso*.

```
function pasos_ini(objode,objprob,objmalla,pasos)
%pasos_ini(objode,objprob,objmalla,pasos)

%Función arranque de los métodos multipaso no lineales. En el caso de que
%haya 3 argumentos de entrada repite el bucle desde 2 hasta pasos=orden. Si hay 4
%argumentos de entrada, se le indica en "pasos" hasta dónde avanza el bucle
%(este es el caso de los métodos NDF que utilizan un valor más en su
%fórmula de avance).
yini=objode.yini; tini=objode.interv(1); h=objode.h; orden=objode.orden;
objtrap=trap2_h(); datostrap=objtrap.hdat;
objtrap=trap2_h(datostrap); inic(objtrap,objprob,objmalla);
if nargin==3
    pasos=orden;
end
ykm1=yini;
for j1=2:pasos
    [tini,yini]=paso(tini, yini, h, objtrap, objprob, objmalla);
    objode.tsol(j1)=tini; objode.ysol(j1,:)=yini;
    ykm1=[yini; ykm1];
end
objode.ykm1=ykm1; objode.dat.un=objtrap.dat.un; objode.npasosini=pasos;
```

```
function actpaso(tk1,yk1,objode,varargin)
%actpaso(tk1,yk1,objode,varargin)

%Función que actualiza el paso en los métodos multipaso no lineales.
actpaso@base_met_h(tk1,yk1,objode,varargin{:})
yk1=objode.yk1; n=size(yk1,1); %El número de filas de yk.
objode.yk1=[yk1; yk1(1:end-n,:)];
```

D.1.1. Objeto bdf

Funciones del objeto bdf: *inic*, *paso*, *calcR*, *calcJ*.

```
function inic(objode,objprob,objmalla)
% inic(objode,objprob,objmalla)

%Coeficientes del método BDF escritos de esta manera: alfas=[alf0 alf1 alf2..... ].
%Los coeficientes del método de orden 1 en la primera fila, etc.
alfas=[-1 1 0 0 0 0 0 0;...
        1/2 -2 3/2 0 0 0 0 0;...
        -1/3 3/2 -3 11/6 0 0 0 0;...
        1/4 -4/3 3 -4 25/12 0 0 0;...
        -1/5 5/4 -10/3 5 -5 137/60 0 0;...
        1/6 -72/60 225/60 -4600/60 450/60 -6 147/60 0 0;...
        -1/7 7/6 -21/5 35/4 -35/3 21/2 -7 1089/420 0];
%%-----fin de los coeficientes del método BDF-----

objode.objfty.inicM(objode,objprob,objmalla)
%Inicialización del problema físico, antes de los pasos de inicio:
objode.objfty.inic_nl(objode,objprob,objmalla)

objode.alfas=alfas; k=objode.orden; M=objode.M; objode.C2=alfas(k,k)*M;
for j1=2:k
    objode.C2=[objode.C2 alfas(k,k+1-j1)*M];
end
pasos_ini(objode,objprob,objmalla)
```

```
function [tk1,yk1] = paso(tk,yk,h,objode,varargin)
% [tk1,yk1] = paso(tk,yk,h,objode,...)

% Paso BDF para un sistema de EDOs no lineal: My'=f(t,y). Avanzamos de t_{n+k-1}
% a t_{n+k}.
% yk: solución aprox. de My'=f(t,y) en t=t_{n+k-1}.
% h: amplitud del paso.
% [tk1,yk1]: valores actualizados de (tk,yk).

% Paso de (n+k-1) a (n+k):
tk1 = tk+h; % t -> n+k

objode.objnr.xini=yk;
[~,~,yk1]=resolver(objode.objnr,tk1,yk,objode,varargin{:});
yk1=yk1(:,end);

actpaso(tk1,yk1,objode,varargin{:})
```

```
function Rb=calcR(yk1,tk1,yk,objjode,varargin)
% Rb=calcR(yk1,tk1,yk,objjode,...)

objjfty=objjode.objjfty;
fk1=objjfty.calcf(tk1,yk1,objjode,varargin{:});
M=objjode.M; h=objjode.h; k=objjode.orden; ykm1=objjode.ykm1;
alfas=objjode.alfas; C2=objjode.C2;

%Cálculo del Residuo Rbarra:
Rb=alfas(k,k+1)*M/h*yk1+C2*ykm1/h-fk1;
```

```
function Jb=calcJ(u,varargin)
% Jb=calcJ(u,varargin)

[objjode,objjprob,objjmall,objjJ]=varargin{end-3:end};
n=objjprob.ngdla; u=u(1:n);
u=u-objjode.dat.un(objjprob.gdla);
M=objjode.M; h=objjode.h; objjfty=objjode.objjfty; k=objjode.orden;
%Cálculo de df/dy (yk1 ha actualizado la malla en calcR):
% j=objjfty.calcj(u,varargin{:});
j=objjfty.calcj(u,objjode,objjprob,objjmall,objjJ);

%Cálculo de la matriz jacobiana (del residuo) Jbarra:
Jb=objjode.alfas(k,k+1)*M/h-j;
```

D.1.2. Objeto ndf

Funciones del objeto ndf: *inic*, *calcR*, *calcJ*.

```
function inic(objjode,objjprob,objjmall)
% inic(objjode,objjprob,objjmall)

inic@bdf(objjode,objjprob,objjmall)
k=objjode.orden;
n=size(objjode.yini,1);
%El método NDF en la inicialización da un paso más que el resto de métodos
%multipaso.
pasos_ini(objjode,objjprob,objjmall,k+1)

kappa=objjode.kappa; gamma=objjode.gamma; M=objjode.M;
C4=kappa(k)*gamma(k)*M*(-1)*(k+1);
for j1=2:(k+1)
    C4=[C4 kappa(k)*gamma(k)*M*(-1)^(j1)*...
        factorial(k+1)/(factorial(k+1-j1)*factorial(j1))];
end
objjode.C2=[objjode.C2 zeros(n)];
objjode.C4=C4;
```

```
function Rb=calcR(yk1,tk1,yk,objode,varargin)
% Rb=calcR(yk1,tk1,yk,objode,...)

%Hereda la Rb del objeto bdf:
k=objode.orden; ykm1=objode.ykm1; h=objode.h;
kappa=objode.kappa; gamma=objode.gamma; M=objode.M;
Rb1=calcR@bdf(yk1,tk1,yk,objode,varargin{:});
Rb2=-kappa(k)*gamma(k)*M/h*yk1-objode.C4*ykm1/h;
Rb=Rb1+Rb2;
```

```
function Jb=calcJ(u,varargin)
% Jb=calcJ(u,varargin)

[objode,~,~,~]=varargin{end-3:end};

k=objode.orden; M=objode.M;
h=objode.h;
kappa=objode.kappa; gamma=objode.gamma;
Jb1=calcJ@bdf(u,varargin{:});
Jb2=-kappa(k)*gamma(k)*M/h;
Jb=Jb1+Jb2;
```

D.2. Objeto métodos multipaso lineales

Funciones del objeto mplinc: *pasos_ini*, *actpaso*.

```
function pasos_ini(objode,objprob,objmalla,pasos)
%pasos_ini(objode,objprob,objmalla)

%Función arranque de los métodos multipaso lineales. En el caso de que
%haya 3 argumentos de entrada repite el bucle desde 2 hasta pasos=orden. Si hay 4
%argumentos de entrada, se le indica en "pasos" hasta dónde avanza el bucle
%(este es el caso de los métodos NDF que utilizan un valor más en su
%fórmula de avance).
yini=objode.yini; tini=objode.interv(1); h=objode.h; orden=objode.orden;
objtrap=trap2linc_h(); datostrap=objtrap.hdat;
objtrap=trap2linc_h(datostrap); inic(objtrap,objprob,objmalla);
if nargin==3
    pasos=orden;
end
ykm1=yini;
for j1=2:pasos
    [tini,yini]=paso(tini, yini, h, objtrap, objprob, objmalla);
    objode.tsol(j1)=tini;
    objode.ysol(j1,:)=yini;
    ykm1=[yini; ykm1];
end
objode.ykm1=ykm1; objode.npasosini=pasos;
```

```
function actpaso(tk1,yk1,objjode,objprob,objmalla)
%actpaso(tk1,yk1,objjode,objprob,objmalla)

%actpaso@base_met_h(tk1,yk1,objjode,varargin{:})
actpaso@base_met_h(tk1,yk1,objjode,objprob,objmalla)
ykm1=objjode.ykm1;
n=size(yk1,1); %El número de filas de yk.
objjode.ykm1=[yk1; ykm1(1:end-n,:)];
```

D.2.1. Objeto bdfinc

Funciones del objeto bdfinc: *inic*, *paso*.

```
function inic(objjode,objprob,objmalla)
% inic(objjode,objprob,objmalla)

% A es la matriz de coeficientes del sistema de EDOs My'=Ay+g(t).
% La matriz C2 multiplica a Y_{n+k-1}: [y_{n+k-1} y_{n+k-2} ... y_n].
% Denotamos por ykm1 la matriz Y_{n+k-1}.

%Coeficientes del método BDF escritos de esta manera: alfas=[alf0 alf1 alf2..... ]
%Los coeficientes del método de orden 1 en la primera fila, etc.
objjode.alfas=[-1 1 0 0 0 0 0 0 0;...
    1/2 -2 3/2 0 0 0 0 0 0;...
    -1/3 3/2 -3 11/6 0 0 0 0 0;...
    1/4 -4/3 3 -4 25/12 0 0 0 0;...
    -1/5 5/4 -10/3 5 -5 137/60 0 0 0;...
    1/6 -72/60 225/60 -4600/60 450/60 -6 147/60 0 0;...
    -1/7 7/6 -21/5 35/4 -35/3 21/2 -7 1089/420 0];
alfas=objjode.alfas;
%-----fin de los coeficientes del BDF-----

A=objjode.objfty.Alin(objprob,objmalla);
M=objjode.objfty.inicM(objjode,objprob,objmalla);
tini=objjode.interv(1); h=objjode.h; yini=objjode.yini;
g0=objjode.objfty.g(tini,objprob,objmalla); objjode.gk=g0;
I=speye(size(A)); k=objjode.orden;

objjode.C1=(alfas(k,k+1)*M-h*A)\I;
objjode.C2=alfas(k,k)*M;
for j1=2:k
    objjode.C2=[objjode.C2 alfas(k,k+1-j1)*M];
end

%Arrancamos con el trapezoidal lineal:
pasos_ini(objjode,objprob,objmalla)
```

```

function [tk1,yk1] = paso(tk,yk,h,objode,objprob,objmalla)
%function [tk1,yk1] = paso(tk,yk,h,objode,objprob,objmalla)

% Paso del método BDF de varios órdenes para un sistema de EDOs lineal de Coef. const.
% ftylin contiene el nombre de un fichero (._m) donde se define el
% 2º miembro del sistema de EDOs en la forma especial para el probl. lineal: A, G.
% yk: solución aprox. en t=t_{n+k-1}.
% h: amplitud del paso.
% [tk1,yk1]: valores actualizados de (tk,yk).

%%%%%%%% Paso de (n+k-1) a (n+k) para sistema de n EDOs lineales My'=Ay+G %%%%%%%%%
k=objode.orden; objfty=objode.objfty;
tk1=tk+h;
s=size(yk,1); %El número de filas de yk.
TI=-objode.C2*objode.ykml;
gk1=objfty.g(tk1,objprob,objmalla);
if ~isempty(gk1) %Ojo inicio nulo de g(t).
    TI=TI+gk1*h;
end
yk1=objode.C1*TI;
%Actualización del paso:
actpaso(tk1,yk1,objode,objprob,objmalla)

```

D.2.2. Objeto ndflinc

Funciones del objeto ndflinc: *inic*.

```

function inic(objode,objprob,objmalla)
% inic(objode,objprob,objmalla)

% A es la matriz de coeficientes del sistema de EDOs My'=Ay+g(t).
A=objode.objfty.Alin(objprob,objmalla);
M=objode.objfty.inicM(objode,objprob,objmalla);
h=objode.h; yini=objode.yini; I=speye(size(A));

inic@bdfinc(objode,objprob,objmalla)
k=objode.orden;
n=size(objode.yini,1);
pasos_ini(objode,objprob,objmalla,k+1)
kappa=objode.kappa; gamma=objode.gamma; alfas=objode.alfas;
objode.C1=(alfas(k,k+1)*M-h*A-kappa(k)*gamma(k)*M)\I;
C4=kappa(k)*gamma(k)*M*(-1)^(k+1);
for j1=2:(k+1)
    C4=[C4 kappa(k)*gamma(k)*M*(-1)^(j1)*factorial(k+1)/(factorial(k+1-j1)*...
                                                factorial(j1)];
end
objode.C4=C4; objode.C2=[objode.C2 zeros(n)];
objode.C2=objode.C2-objode.C4;

```

Apéndice E

Publicaciones y participación en Congresos

E.1. Publicaciones científicas

- E. Alberdi, J. J. Anza. A predictor modification to the EBDF method for stiff systems. *J. Comput. Math.*, 29 (2):199-214, 2011.
- E. Alberdi, J. J. Anza. The integration of stiff systems of ODEs using NDFs and MEBDFs. *J. Comput. Math.* Aceptado y en espera de ser publicado.
- E. Alberdi, J. J. Anza. Changing the local error estimation in ode23 and ode45. *Appl. Comput. Math.* Aceptado y en espera de ser publicado.
- E. Alberdi, J. J. Anza. BDF- α : A multistep method with numerical damping control. *IMA J. Appl. Math.* En revisión.
- E. Alberdi, J. J. Anza. Construction and performance of the MATLAB ODE solvers *ode45* and *ode15s*. Terminado, en espera de enviarlo.

E.2. Participación en Congresos

- E. Alberdi, J. J. Anza. International Congress of Mathematicians 2010 (ICM 2010), *short communication* titulado: Reasons why the ode45 performs better than the ode15s in some stiff problems. Hyderabad (India), 19-27 agosto de 2010.
- E. Alberdi, J. J. Anza. Jornadas RSME de transferencia y Matemática Industrial, *communication* titulado: BDF- α method: modification to the BDF2. Santiago de Compostela (España), 12-14 julio 2011.

E.3. Publicaciones de divulgación

- E. Alberdi. Ekuazio diferentzialak egonkortasun bila. Elhuyar zientzia eta teknologia aldizkaria, 258:49-52, noviembre 2009.
- E. Alberdi. Relación entre el resultado de una ecuación diferencial tras aplicar un método numérico y las sumas de Riemann, Revista matemática SIGMA del Gobierno Vasco, 35:133-142, abril 2010.
- E. Alberdi. Hiru aldiz erratu eta zuzen bukatzearen magia, Ekaia aldizkaria, 23:79-88, 2010.

Índice de figuras

1.1. Condición inicial del caso del pulso ($t = 0$).	12
1.2. Solución del caso del pulso para $t = 2$	12
1.3. Solución del caso triangular para $t = 0, 1, 2$	13
1.4. Factor de amplificación modal (A_k).	13
1.5. Frecuencias del modelo continuo y discreto (masa consistente).	16
1.6. Modos 1, 2 y 10 (continuos y discretos).	17
1.7. Modo 99 del continuo.	17
1.8. Modo 99 del discreto.	17
1.9. Participación modal $ A_k $, $ Y_i(0) $ para la CI pulso.	18
1.10. Participación modal para la CI pulso (detalle de la Figura 1.9).	18
1.11. Solución del pulso por superposición modal (400 elem., 399 modos).	19
1.12. Solución del pulso por superposición modal (1600 elem., 1599 modos).	19
1.13. Solución del pulso por superposición modal (1600 elem., 399 modos).	19
1.14. Solución del pulso por superposición modal (1600 elem., 99 modos).	20
1.15. Solución del pulso por superposición modal (1600 elem., 25 modos).	20
1.16. Solución del pulso por superposición modal (25 modos continuos).	21
1.17. Solución de la ecuación de difus. CI senoidal (<i>ode15s</i> , <i>ode45</i> , RK4, trap.).	22
1.18. Solución de la ecuación de difusión CI triangular (métodos <i>ode45</i> , <i>ode15s</i>).	24
1.19. Solución de la ecuación de difusión CI pulso (métodos <i>ode45</i> , <i>ode15s</i>).	25
1.20. Solución de la ecuación de onda CI senoidal (<i>ode15s</i> , <i>ode45</i> , RK4, trap.).	27
1.21. Solución de la ecuación de onda CI triangular (métodos <i>ode15s</i> y <i>ode45</i>).	27
1.22. Solución de la ecuación de onda CI triangular (método <i>ode45</i>).	28
1.23. Solución de la ec. de onda CI triangular (método trap. 1000 pasos).	28
1.24. Solución de la ec. de onda CI triangular (método trap. 100 pasos).	29
1.25. Solución de la ec. de onda CI pulso (método <i>ode15s</i> 100 elem.).	30
1.26. Solución de la ec. de onda CI pulso (método trap. 100 elem. 1000 pasos).	30
1.27. Solución de la ec. de onda CI pulso (método <i>ode15s</i> 1000 elem.).	31
1.28. Solución de la ec. de onda CI pulso (método trap. 1000 elem. 1000 pasos).	31
1.29. Solución de la ec. de onda CI pulso (método <i>ode15s</i> 400 elem.).	32
1.30. Solución de la ec. de onda CI pulso (método trap. 400 elem. 1000 pasos).	32
1.31. Solución de la ec. de onda CI pulso (método trap. 400 elem. 2000 pasos).	32
1.32. Solución de la ec. de onda CI pulso (mét. HHT- α 400 elem. 1400 pasos).	33

1.33. Solución ec. onda CI pulso (mét. Newmark $\beta = \frac{1}{6}$, $\gamma = 0,5$ 400 elem.).	33
1.34. Solución ec. onda lineal (método trapezoidal 2 elem.).	35
1.35. Solución ec. onda lineal (método trapezoidal 100 elem.).	35
1.36. Solución ec. onda no lineal (método trapezoidal y <i>ode15s</i> 2 elem.).	36
1.37. Solución ec. onda no lineal (método trapezoidal y <i>ode15s</i> 20 elem.).	37
1.38. Solución ec. onda no lineal (método <i>ode15s</i> 100 elem.).	37
1.39. Solución ec. onda no lineal (mét. trapezoidal 100 elem. 200 y 1000 pasos).	38
1.40. Solución ec. onda no lineal (HHT- α , 20 elementos).	38
1.41. Solución ec. onda no lineal (mét. trap. y <i>ode15s</i> , 20 elementos).	39
1.42. Solución ec. onda no lineal (HHT- α , 20 elementos).	40
2.1. Ilustración del error local y global de truncamiento.	47
2.2. A-estabilidad y $A(\alpha)$ -estabilidad.	50
2.3. Regiones de estabilidad de Adams Bashforth y Moulton (interior a curva).	56
2.4. Regiones de estabilidad de los métodos BDF (exterior a curva).	58
2.5. Regiones de estabilidad de los métodos de Enright (exterior a curva).	62
2.6. Regiones de estabilidad de los métodos SDBDF (exterior a curva).	62
2.7. Regiones de estabilidad de los métodos EBDF (exterior a curva).	64
2.8. Regiones de estabilidad de los métodos combinados (exterior a curva).	65
2.9. Regiones de estabilidad Runge-Kutta con $s = p$ (interior a curva).	72
2.10. Regiones de estabilidad RKF45 y DOPRI(5,4) (interior a curva).	75
3.1. Autovalores del método de Newmark $\gamma = 1/2, \beta = 1/4$	89
3.2. Autovalores y radio espectral del método de Newmark $\gamma = 0,4, \beta = 0,6$	90
3.3. Autovalores y radio espectral del método de Newmark $\gamma = 0,5, \beta = 0,2$	90
3.4. Estabilidad del método de Newmark.	91
3.5. Autovalores y radio espectral del método de Newmark $\gamma = 0,6, \beta = 0,3025$	92
3.6. Autovalores y radio espectral del método de Newmark $\gamma = 0,6, \beta = 0,3$	92
3.7. Autovalores y radio espectral del método HHT- $\alpha = 0,3$	93
3.8. Autovalores y radio espectral del método HHT- $\alpha = 0,05$	93
3.9. Radios espectrales de algunos métodos.	94
3.10. Amortiguamiento algorítmico de algunos métodos.	98
3.11. Error relativo en el periodo de algunos métodos.	98
4.1. Regiones $ R_5 - R_4 \leq Rtol$ del método DOPRI(5,4).	105
4.2. Regiones de estabilidad DOPRI(5,4), BDF y NDF.	110
4.3. Posición \hat{h} de (4.92) en la región de estab. DOPRI(5,4), con $\lambda = -1$	121
4.4. Estimación del error y tamaños de paso en (4.92), con $\lambda = -1$	121
4.5. Funcionamiento del contador <i>ode15s</i> en (4.92), con $\lambda = -1$	122
4.6. Siguiete tamaño de paso tras paso bueno en <i>ode15s</i> para (4.92), $\lambda = -1$	123
4.7. Siguiete tamaño de paso tras paso fallido en <i>ode15s</i> para (4.92), $\lambda = -1$	123
4.8. Posición \hat{h} de (4.92) en la región de estab. DOPRI(5,4), con $\lambda = -100$	124
4.9. Tamaños de paso en (4.92), con $\lambda = -100$ (dcha. detalle).	124
4.10. Estimación del error en (4.92), con $\lambda = -100$ (dcha. detalle).	124

4.11. Estimación del error y tamaños de paso en (4.94), con $\lambda^2 = -1$ 126

4.12. Posición \hat{h} de (4.94) en la región de estab. DOPRI(5,4), con $\lambda^2 = -1$. . . 126

4.13. Posición \hat{h} de (4.94) en región de estab. BDF, con $\lambda^2 = -1$ (dcha. detalle). 126

4.14. Posición \hat{h} de (4.94) en región de estab. BDF, (detalle de la Figura 4.13). 127

4.15. Funcionamiento del contador *ode15s* en (4.94), con $\lambda^2 = -1$ 127

4.16. Siguiete tamaño de paso tras paso bueno en *ode15s* para (4.94), $\lambda^2 = -1$. 128

4.17. Siguiete tamaño de paso tras paso fallido en *ode15s* para (4.94), $\lambda^2 = -1$. 128

4.18. Soluciones de (4.94), $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$ 128

4.19. Soluciones de (4.94), $\lambda^2 = -100^2$ y $Rtol=10^{-2}$ 129

4.20. Tamaños de paso para el problema (4.94), $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$. . . 129

4.21. Estimación del error en (4.94), con $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$ 129

4.22. Posición \hat{h} de (4.94) en la región de estab. DOPRI(5,4), con $\lambda^2 = -100^2$. . 130

4.23. Posición \hat{h} de (4.94) en región de estab. BDF, $\lambda^2 = -100^2$ y $Rtol = 10^{-3}$. 130

4.24. Tamaños de paso para el problema (4.94), $\lambda^2 = -100^2$ y $Rtol = 10^{-2}$. . . 131

4.25. Estimación del error en (4.94), con $\lambda^2 = -100^2$ y $Rtol = 10^{-2}$ 131

4.26. Posición \hat{h} de (4.94) en región de estab. BDF, $\lambda^2 = -100^2$ y $Rtol = 10^{-2}$. 132

4.27. Solución y estimación del error en la 1ª componente para (4.94), $\lambda^2 = -1$. 133

4.28. Solución y estimación del error en la 2ª componente para (4.94), $\lambda^2 = -1$. 133

4.29. Solución y estimación del error en la 1ª componente para (4.94), $\lambda^2 = -2$. 133

4.30. Solución y estimación del error en la 2ª componente para (4.94), $\lambda^2 = -2$. 134

4.31. Solución y estimación del error en la 1ª componente para (4.94), $\lambda^2 = -4$. 134

4.32. Solución y estimación del error en la 2ª componente para (4.94), $\lambda^2 = -4$. 134

4.33. Solución y estimación del error con norma euclídea (4.94), $\lambda^2 = -1, -2$. . 135

4.34. Solución y estimación del error con norma euclídea (4.94), $\lambda^2 = -4$ 135

4.35. Tamaños de paso en (1.23) con CI pulso. 136

4.36. Estimación del error en (1.23) con CI pulso. 136

4.37. Detalle de las regiones de estabilidad BDF y NDF. 137

5.1. Regiones de estab. ENDF y EBDF, MENDF y MEBDF (exterior a curva). 146

6.1. Amort. algrít. varios métodos en $y'' = -10^2y$, $(y(0), y'(0)) = (0, 1)$ 156

6.2. Radios espectrales de los métodos BDF. 158

6.3. Radios espectrales de los métodos EBDF. 159

6.4. Constantes de error del método BDF- α en valor absoluto ($|C|$). 163

6.5. Regiones de estabilidad del método BDF- α (exterior a curva). 164

6.6. Radios espectrales del método BDF- α 166

6.7. Radios espectrales del método BDF- α (detalle de la Figura 6.6). 166

6.8. Amortiguamiento algrítmico del método BDF- α 167

6.9. Error relativo en el periodo del método BDF- α 167

6.10. Error relativo en el periodo del método BDF- α (detalle de la Figura 6.9). 168

7.1. Arquitectura del objeto principal por agregación de objetos subordinados. 170

7.2. Organización del proceso de cálculo, problema lineal. 174

7.3. Arquitectura de objetos problema lineales por herencia. 176

7.4. Arquitectura de objetos elemento finito por herencia.	177
7.5. Transformación del dominio patrón al dominio real del elemento.	178
7.6. Arquitectura de objetos elemento patrón por herencia.	179
7.7. Arquitectura de objetos punto de integración por herencia.	179
7.8. Arquitectura de objetos material por herencia.	180
7.9. Dos posibles casos de la arquitectura elemento finito (caso lineal).	181
7.10. Organización del proceso de cálculo, problema no lineal.	185
7.11. Arquitectura de objetos punto de integración por herencia.	186
7.12. Arquitectura de objetos deformación por herencia.	187
7.13. Arquitectura de objetos material por herencia.	188
7.14. Arquitectura de objetos problema lineales y no lineales por herencia.	189
7.15. Arquitectura de objetos Jacobiano por herencia.	190
7.16. Dos posibles casos de la arquitectura elemento finito (caso no lineal).	192
7.17. Arquitectura de objetos fty por herencia.	197
7.18. Arquitectura de objetos método ODE por herencia.	198
7.19. Arquitectura de objetos métodos multipaso no lineales por herencia.	203
7.20. Arquitectura de objetos métodos multipaso lineales por herencia.	204
7.21. Dos posibles casos de la arquitectura elemento finito (caso dinámico).	224
7.22. Solución de la ec. de onda CI pulso (HHT- $\alpha = 0,05$ 400 elem. 1400 pasos).	225
7.23. Radios espectrales del problema (1.23), método HHT- α	225
7.24. Solución de la ec. de onda CI pulso (HHT- $\alpha = 0,3$ 400 elem. 2000 pasos).	225
7.25. Solución ec. onda CI pulso (BDF- $\alpha = -0,475065$ 400 elem. 1400 pasos).	226
7.26. Solución ec. onda CI pulso (BDF- $\alpha = -0,35$ 400 elem. 1400 pasos).	226
7.27. Radios espectrales del problema (1.23), método BDF- α	227
7.28. Solución de la ec. de onda CI pulso (BDF- $\alpha = -0,5$ 400 elem. 1400 pasos).	227
7.29. Solución de la ec. de onda CI pulso (BDF- $\alpha = 0$ 400 elem. 1400 pasos).	228
7.30. Solución ec. onda CI pulso (BDF- $\alpha = 9,51$, $\alpha = 1,17$ 400 elem. 1400 pasos).	228
7.31. Solución de la ec. de onda CI pulso (EBDF 400 elem. 1400 pasos).	228
7.32. Radios espectrales del problema (1.23), método EBDF.	229
7.33. Solución de la ec. de onda CI pulso (ENDF y MENDF 400 elem.).	229
7.34. Solución ec. onda no lineal (método BDF- $\alpha = 0$).	231
7.35. Solución ec. onda no lineal (HHT- $\alpha = 0,05$ y BDF- $\alpha = -0,475065$).	231
7.36. Solución ec. onda no lineal (métodos BDF- $\alpha = 1,17$ y BDF- $\alpha = -0,35$).	232
7.37. Detalle de las Figuras correspondientes a HHT- $\alpha = 0,3$ y BDF- $\alpha = -0,35$	233
7.38. Detalle de las Figuras correspondientes a HHT- $\alpha = 0,3$ y BDF- $\alpha = -0,35$	233
7.39. Arquitectura del ejemplo 3.	234
7.40. Modos 1, 5, 8 y 13 del bloque del ejemplo 3.	235
7.41. Tensiones y la deformada del ejemplo 3 en el instante final ($t = 0,5$).	236
7.42. Evolución temporal de los desplazamientos nodales (mét. trap. 200 pasos).	236
7.43. Evolución temporal nodo superior (HHT- $\alpha = 0$ y <i>supmod</i>) y detalle.	237
7.44. Evolución temporal nodo superior (BDF- $\alpha = -0,5$ y ENDF3).	237
7.45. Modo 1 del bloque del ejemplo 3.	238
7.46. Evolución temporal de los desplazamientos nodales (mét. trap. 200 pasos).	238

7.47. Modo 60 del bloque del ejemplo 3.	238
7.48. Evolución temporal de los nodos (mét. trap. 1000 pasos).	238
7.49. Evolución temporal de los nodos BDF- $\alpha = -0,35$ (317 y 1000 pasos).	239
7.50. BDF- $\alpha = -0,35$ y 2000 pasos.	239
7.51. Evolución del primer modo BDF- $\alpha = -0,35$ y 317 pasos.	239
7.52. Evolución temporal de un nodo BDF- $\alpha = -0,35$, 317 pasos (dcha. detalle).	240
7.53. Evolución temp. de un nodo BDF- $\alpha = -0,35$, 317 y 1000 pasos.	240
7.54. Evolución temporal nodo superior con varios métodos, derecha detalle.	240
7.55. Arquitectura del ejemplo 4, caso dinámico.	241
7.56. Diagrama fuerzas-desplazamientos del ejemplo 4.	242
7.57. Discretización y deformada del ejemplo 4.	242
7.58. Isolíneas de deformación plástica del ejemplo 4.	242
7.59. Diagrama fuerzas-desplazamientos del punto central superior de la laja.	243
7.60. Evolución temporal del desplazamiento vertical (nodo central sup.).	244
7.61. Resultados con el mét. trap. 500 pasos.	244
7.62. Isolíneas de deformación plástica del ejemplo 4 (caso dinámico).	244
7.63. Arquitectura del ejemplo 5, caso estático.	245
7.64. Tensiones y deformada del ejemplo 5, caso estático.	245
7.65. Desplazamientos nodales del ejemplo 5, caso estático.	245
7.66. Tensiones y deformada del ejemplo 5, caso dinámico.	246
7.67. Desplazamientos nodales del ejemplo 5, caso dinámico.	246
7.68. Verticales del nodo central del extremo del voladizo, ejemplo 5.	246
7.69. Solución del ejemplo 5 en deformación plana y material neohookeano.	247
7.70. Solución del sistema no lineal masa-muelle, energía-momento $h = 1$	248
7.71. Solución del sistema no lineal masa-muelle, método EDMC-1 $h = 1$	248
7.72. Solución del sistema no lineal masa-muelle, método HHT- α $h = 1$	249
7.73. Solución del sistema no lineal masa-muelle, método BDF- α $h = 1$	249
7.74. Solución del sistema no lineal masa-muelle, energía-momento $h = 1,6775$	250
7.75. Solución del sistema no lineal masa-muelle, método EDMC-1 $h = 1,6775$	251
7.76. Solución del sistema no lineal masa-muelle, método HHT- α $h = 1,6775$	251
7.77. Solución del sistema no lineal masa-muelle, método BDF- α $h = 1,6775$	252
B.1. Mayor autov. en módulo en función del n° elem. con epoxi, epoxi alúmina.	262
B.2. Mayor autovalor de la EDP onda en función del n° elem, nota LA.	268

Índice de tablas

2.1. Coeficientes del método de Adams Basforth [38].	55
2.2. Coeficientes del método de Adams Moulton [38].	56
2.3. Coeficientes y constantes de error de los métodos BDF.	57
2.4. $A(\alpha)$ -estabilidad de los métodos BDF.	58
2.5. NDFs de Klopfenstein y Shampine: eficiencia y estabilidad respecto BDF.	61
2.6. Características de los métodos de Enright.	62
2.7. Características de los métodos SDBDF.	62
2.8. Características del método New Efficient SDMM con $\beta^* = -0,2, \gamma^* = 0,2$	63
2.9. Características del método New Efficient SDMM con $\beta^* = -0,05, \gamma^* = 0,9$	63
2.10. Características de los métodos EBDF.	63
2.11. Características de los métodos MEBDF.	64
2.12. Características del bloque avanzado implícito.	64
2.13. Características de los métodos A-EBDF.	66
2.14. Características de los métodos E2BD.	66
2.15. Características de los métodos New SDMM.	66
2.16. Tabla Butcher de un método Runge-Kutta.	67
2.17. Representación matricial de la Tabla Butcher.	67
2.18. Valores de $\sigma_{r,q}, \gamma_{r,q}, \phi_{r,q}$ y $D_{r,q}$ hasta orden 5.	70
2.19. N° de condiciones a cumplir en los métodos Runge-Kutta (órdenes 1-10).	71
2.20. Relación entre orden y etapas en los métodos Runge-Kutta explícitos [52].	71
2.21. Tabla Butcher de los métodos Runge-Kutta anidados.	73
2.22. Constantes del método Runge-Kutta DOPRI (5,4) (<i>ode45</i>).	75
4.1. Valores de las constantes A_k, A_{k-1} y A_{k+1}	117
4.2. Número de pasos dado por la <i>ode45</i> y la <i>ode15s</i> en el problema (4.92).	120
4.3. Número de pasos dado por la <i>ode45</i> y la <i>ode15s</i> en el problema (4.94).	125
5.1. $A(\alpha)$ -estabilidad de los métodos EBDF, EBPDF, ENPDF, ENDF.	150
5.2. $A(\alpha)$ -estabilidad de los métodos MEBDF, MEBPDF, MENPDF, MENDF.	150
5.3. Resultados de integración del ejemplo 1 con NDF4, ENDF3 y MENDF3.	151
5.4. Resultados de integración del ejemplo 1 con BDF4, EBDF3 y MEBDF3.	152
5.5. Resultados de integración del ejemplo 2 utilizando la familia EBDF.	152
5.6. Resultados de integración del ejemplo 2 utilizando la familia MEBDF.	153

5.7. Resultados de integración del ejemplo 3 utilizando la familia EBDF.	153
5.8. Resultados de integración del ejemplo 3 utilizando la familia MEBDF.	153
7.1. Tiempo computacional al resolver (1.23) 400 elem., 1400 pasos.	230
7.2. Tiempo computacional al resolver (1.23) 400 elem., 1400 pasos.	230
B.1. Autovalores EDP difusión epoxi y epoxi alúm. en función del n° de elem.	262
B.2. Autovalores EDP difusión para epoxi y epoxi alúm. en función de L.	263
B.3. Autovalores EDP difusión epoxi alúm. y cobre en función del n° de elem.	265
B.4. Autovalores EDP difusión y onda, con $\alpha^2 = 1$ y $L = 8$	267
B.5. Autovalores EDP onda para dar la nota LA.	268

Bibliografía

- [1] <http://www.mathworks.com>.
- [2] *The student edition of MATLAB: The Language of technical computing*. The Mathworks Inc., 1997.
- [3] Matlab 7. *Classes and Object Oriented Programming*. The Mathworks Inc., 2008.
- [4] F. Armero and I. Romero. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part I: low order methods for two model problems and nonlinear elastodynamics. *Comput. Meth. Appl. Mech. Eng.*, 190:2603–2649, 2000.
- [5] F. Armero and I. Romero. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part II: second order methods. *Comput. Meth. Appl. Mech. Eng.*, 190:6783–6824, 2001.
- [6] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, Philadelphia, 1998.
- [7] V. Avansini Botta. *Zeros de polinômios característicos e estabilidade de métodos numéricos*. PhD thesis, Instituto de Ciências matemáticas y de la Computación, USP-São Carlos, 2008.
- [8] F. Bashforth and J. C. Adams. *An attempt to test the theories of Capillary action by comparing the theoretical and measured forms of drops of fluid*. Cambridge University Press, Cambridge, 1883.
- [9] T. Belytschko, W. Kam Liu, and B. Moran. *Nonlinear finite elements for continua and structures*. John Wiley & Sons, Chichester, 2000.
- [10] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, Cambridge, 1997.
- [11] P. N. Brown and A. C. Hindmarsh. Matrix-free methods for stiff systems of ODEs. *SIAM J. Numer. Anal.*, 23 (3):610–638, 1986.
- [12] R. L. Burden and J. D. Faires. *Numerical Methods*. Brooks/Cole publishing Thomson Learning, Pacific Grove, California, 1998.

- [13] J. C. Butcher. Numerical methods for Ordinary Differential Equations in the 20th century. *J. Comput. Appl. Math.*, 125:1–29, 2000.
- [14] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Chichester, 2008.
- [15] S. L. Campbell. *An introduction to differential equations and their applications (second Edition)*. Wadsworth Pub. Co., California, 1990.
- [16] J. R. Cash. On the integration of stiff systems of ODEs using extended backward differentiation formula. *Numer. Math.*, 34 (3):235–246, 1980.
- [17] J. R. Cash. Second Derivative Extended Backward Differentiation formulas for the numerical integration of stiff systems. *SIAM J. Numer. Anal.*, 18 (1):21–36, 1981.
- [18] J. R. Cash. The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae. *Comput. Math. Appl.*, 9 (5):645–657, 1983.
- [19] J. R. Cash. Modified Extended Backward Differentiation Formulae for the numerical solution of stiff initial value problems in ODEs and DAEs. *J. Comput. Appl. Math.*, 125:117–130, 2000.
- [20] J. R. Cash and S. Considine. An MEBDF code for initial value problems. *ACM Trans. Math. Softw.*, 18 (2):142–155, 1992.
- [21] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- α method. *J. Appl. Mech.*, 60 (2):371–375, 1993.
- [22] G. G. Dahlquist. A special stability problem for linear multistep methods. *Bit*, 3 (1):27–43, 1963.
- [23] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, 6 (1):19–26, 1980.
- [24] W. H. Enright. Second derivative multistep methods for stiff ordinary differential equations. *SIAM J. Numer. Anal.*, 11 (2):321–331, 1974.
- [25] S. J. Farlow. *Partial differential equations for Scientists & Engineers*. Dover, New York, 1993.
- [26] E. Fehlberg. Low order classical Runge-kutta formulas with step-size control and their application to some heat transfer problems. *NASA TR R*, 315, 1969.
- [27] E. Fehlberg. Classical fifth, sixth, seventh and eighth order Runge-Kutta formulas with stepsize control. *NASA TR R*, 287, Oct. 1968.
- [28] C. Fredebeul. A-BDF: A generalization of the backward differentiation formulae. *SIAM J. Numer. Anal.*, 35 (5):1917–1938, 1998.

- [29] R. M. French. *Engineering the guitar. Theory and Practice*. Springer, New York, 2009.
- [30] C. W. Gear. *Numerical initial value problems in Ordinary Differential Equations*. Prentice Hall, New Jersey, 1971.
- [31] I. Gladwell and R. Thomas. Stability properties of the Newmark, Houbolt and Wilson θ methods. *Int. J. Numer. Anal. Methods Geomech.*, 4 (2):143–158, 1980.
- [32] M. Golubitsky and M. Dellnitz. *Álgebra lineal y ecuaciones diferenciales con uso de MATLAB*. International Thomson Editores, México, 2001.
- [33] O. Gonzalez. Exact energy-momentum conserving algorithms for general models in nonlinear elasticity. *Comput. Methods Appl. Mech. Eng.*, 190:1763–1783, 2000.
- [34] K. F. Graff. *Wave motion in elastic solids*. University Press, Belfast, 1975.
- [35] P. M. Gresho and R. L. Sani. *Incompressible flow and the finite element method*. John Wiley & Sons, Chichester, 1998.
- [36] E. Hairer and G. Wanner. On the instability of the BDF formulas. *SIAM J. Numer. Anal.*, 20 (6):1206–1209, 1983.
- [37] E. Hairer and G. Wanner. *Solving ordinary differential equations, II, Stiff and Differential Algebraic Problems*. Springer, Berlin, 1991.
- [38] E. Hairer, G. Wanner, and S. P. Nørsett. *Solving ordinary differential equations, I, Nonstiff problems*. Springer, Berlin, 1993.
- [39] E. Haug and H. P. Langtangen. *The Basic Equations in Eulerian Continuum Mechanics*. Birkhäuser, 1997.
- [40] M. T. Heath. *Scientific Computing. An introductory survey*. Mc Graw Hill, New York, 1997.
- [41] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations*. John Wiley and Sons, 1962.
- [42] H. M. Hilber and T. J. R. Hughes. Collocation, dissipation and overshoot for time integration schemes in structural dynamics. *Earthq. Eng. Struct. Dyn.*, 6 (1):99–117, 1978.
- [43] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthq. Eng. Struct. Dyn.*, 5 (3):283–292, 1977.
- [44] G. Hojjati, M. Y. Rahimi Ardabili, and S. M. Hosseini. A-EBDF: an adaptative method for numerical solution of stiff systems of ODEs. *Math. Comput. Simul.*, 66:33–41, 2004.

- [45] G. Hojjati, M. Y. Rahimi Ardabili, and S. M. Hosseini. New second derivative multistep methods for stiff systems. *Appl. Math. Model.*, 30 (5):466–476, 2006.
- [46] S. M. Hosseini and G. Hojjati. Matrix-free MEBDF method for numerical solution of systems of ODEs. *Math. Comput. Modell.*, 29:67–77, 1999.
- [47] T. J.R. Hughes. *The finite element method. Linear Static and dynamic finite element analysis*. Prentice-Hall International Editions, Englewood Cliffs, New Jersey, 1987.
- [48] Gamal A. F. Ismail and Iman H. Ibrahim. A new higher order effective P-C methods for stiff systems. *Math. Comput. Simul.*, 47 (6):541–552, 1998.
- [49] Gamal A. F. Ismail and Iman H. Ibrahim. New efficient second derivative multistep methods for stiff systems. *Appl. Math. Model.*, 23 (4):279–288, 1999.
- [50] E. J. Kostelich and D. Armbruster. *Introductory differential equations: From linearity to chaos*. Addison Wesley, Massachusetts, 1996.
- [51] J. D. Lambert. *Computational Methods in Ordinary Differential Equations*. John Wiley, Chichester, 1973.
- [52] J. D. Lambert. *Numerical methods for ordinary differential systems: the initial value problem*. John Wiley & Sons, New York, 1991.
- [53] H. P. Langtangen. *Computational Partial Differential Equations*. Springer Verlag, Berlin, 1999.
- [54] F. R. Moulton. *New methods in exterior Ballistics*. University of Chicago Press, Chicago, 1926.
- [55] N. M. Newmark. A method of computation for structural dynamics. *J. Eng. Mech. Div.*, 85 (3):67–94, 1959.
- [56] A. C. Peterson and W. G. Kelley. *Difference equations. An introduction with Applications*. Academic Press, New York, 2001.
- [57] G. Psihoyios. A block implicit advanced-step point (BIAS) algorithm for Stiff Differential Systems. *Computing Letters*, 2:51–58, 2006.
- [58] N. Rattenbury. *Almost Runge-Kutta methods for stiff and non-stiff problems*. PhD thesis, The University of Auckland, 2005.
- [59] L. F. Shampine. Error estimation and control for ODEs. *J. Sci. Comput.*, 15:3–16, 2005.
- [60] L. F. Shampine and R. M. Corless. Initial value problems for ODEs in problem solving environments. *J. Comput. Appl. Math.*, 125:31–40, 2000.
- [61] L. F. Shampine, I. Gladwell, and S. Thompson. *Solving ODEs with Matlab*. Cambridge University Press, New York, 2003.

- [62] L. F. Shampine and M. W. Reichelt. The MATLAB ODE Suite. *SIAM J. Sci. Comput.*, 18:1–22, 1997.
- [63] J. C. Simo and T. J. R. Hughes. *Computational inelasticity*. Springer-Verlag, New York, 1998.
- [64] J. C. Simo and N. Tarnow. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *ZAMP*, 43:757–793, 1992.
- [65] R. D. Skeel. Thirteen ways to estimate global error. *Numer. Math.*, 48 (1):1–20, 1986.
- [66] R. D. Skeel and A. K. Kong. Blended linear multistep methods. *ACM Trans. Math. Softw.*, 3 (4):326–345, 1977.
- [67] M. N. Spijker. Stiffness in numerical initial-value problems. *J. Comput. Appl. Math.*, 72:393–406, 1996.
- [68] E. L. Wilson. *A computer program for the dynamic stress analysis of underground structures*. SEL Report 68-1, 1968.
- [69] W. L. Wood. On the Zienkiewicz four-time-level scheme for the numerical integration of vibration problems. *Int. J. Numer. Methods Eng.*, 11 (10):1519–1528, 1977.
- [70] W. L. Wood, M. Bossak, and O. C. Zienkiewicz. An alpha modification of Newmark's method. *Int. J. Numer. Methods Eng.*, 15 (10):1562–1566, 1981.
- [71] O. C. Zienkiewicz. A new look at the Newmark, Houbolt and other time-stepping formulae. A weighted residual approach. *Earthq. Eng. Struct. Dyn.*, 5 (4):413–418, 1977.
- [72] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method. Its basis & fundamentals*. Elsevier Butterworth-Heinemann, Oxford, 2005.