Curso Básico de Métodos Numéricos

Natividad Calvo y Fernando Varas

10 de febrero de 2009

Índice general

Presentación

Probablemente sea conveniente comenzar por aclarar la intención de estos apuntes. Desde luego, estas páginas no aspiran a convertirse en un libro de texto. Son tan sólo unas notas que pretenden acercar a sus lectores a alguno de los numerosos buenos textos sobre métodos numéricos. En ese sentido, los apuntes abordan la presentación de los métodos más elementales refiriendo al lector a otros textos para ampliar sus contenidos. Por otro lado, estas notas se han elaborado pensando en alumnos de Ingeniería o Física. Esto no quiere decir que, eventualmente, alumnos de Matemáticas no puedan sacar algún provecho de su lectura, pero sí que el interés de las presentaciones que se hacen aquí se centra en la aplicación de los métodos numéricos y no tanto en el análisis pormenorizado de sus propiedades.

Ciertamente, estos apuntes se apartan algo de los textos *tradicionales* de métodos numéricos al no detallar la implementación de los algoritmos o los detalles de constructivos de algunos métodos (por ejemplo, el lector puede echar en falta la presentación de detalles sobre el cálculo de factorizaciones de matrices o los algoritmos de diferencias divididas). La razón está en que los autores consideran que (a diferencia de lo que ocurría cuando éstos estudiaron por primera vez métodos numéricos y fueron escritos muchos de los textos clásicos) hoy es poco frecuente que se programen los métodos numéricos más básicos (a los que se dedica este texto) sino que se acuda a alguno de los muy probados y efecientes códigos disponibles (sobre los que se han incluido oportunas menciones). Así, se ha preferido insistir sobre las propiedades de los métodos y lo que cabe (o no) esperar de ellos, antes que dedicar su atención a detalles constructivos que probablemente los lectores no fuesen a valorar.

Aunque algunos textos recientes (o incluso adaptaciones recientes de textos más antiguos) de métodos numéricos para Ingeniería o Física han optado por eliminar toda demostración de las propiedades de los métodos, aquí se ha optado por presentar las más elementales y citar alguna referencia bibliográfica para las más complejas. La razón está en el convencimiento de que esta presentación (junto con la ilustración a través de ejemplos adecuadamente elegidos) refuerza notablemente la comprensión de las capacidades de cada método numérico. Por el contrario, los autores creen que la reciente tendencia a elaborar textos con presentaciones de extensas listas de métodos acompañados eventualmente de alguna propiedad de convergencia pero desprovistos de toda demostración difícilmente puede hacer que el lector adquiera alguna seguridad en la utilización de los métodos numéricos. Ciertamente, con frecuencia no es fácil elegir un buen método numérico para la resolución de un determinado problema, pero probablemente sea mejor que dicha capacidad se adquiera progresivamente (partiendo de una buena comprensión de los métodos más básicos) con el estudio pausado de nuevos métodos y la experiencia ganada con la práctica. Pretender que un alumno enfrentado a

una presentación de decenas y decenas de métodos, junto con observaciones sobre el tipo de problemas para el que es adecuado cada uno de estos métodos, pueda adquirir esta capacidad no parece en absoluto realista.

Al final de cada tema, tras la presentación de los métodos más básicos, se han incluido dos secciones dedicadas a recoger, la primera, una bibliografía relevante para ampliar los contenidos de dicho tema y, la segunda, una referencia a programas o paquetes de programas libremente disponibles que incorporan los métodos presentados. La razón de incluir la primera sección es, como ya se ha dicho, que la aspiración de estas notas es simplemente la de servir de acercamiento a estos textos. La segunda sección se ha incluido porque los autores consideran importante resaltar la conveniencia, a la hora de emplear métodos numéricos, de emplear códigos eficientes y bien depurados. Así, salvo que se trate de un método muy específico (lo cual difícilmente ocurrirá en un primer acercamiento a los métodos numéricos), no debe preferirse la programación por parte del usuario de una función a la utilización de una buena función de las muchas disponibles libremente, pues éstas últimas emplearán con toda seguridad una programación más eficiente y habrán sido bien depuradas tras ser revisadas por una amplia comunidad de usuarios.

Como puede observarse, este documento se distribuye con una versión de la Licencia Creative Commons. Los autores quieren así contribuir (aunque sea muy modestamente, desde luego) a una comunidad a la que deben mucho. La labor de quienes elaboran, depuran o mantienen numerosos códigos de cálculo científico distribuidos bajo licencias libres hace que nuestro trabajo sea inconmensurablemente más fácil. Es desde luego fundamental reconocer y respaldar esa labor, y entendemos que quizás un modo de hacerlo pueda ser (al margen de la propia difusión de los códigos) mediante la tarea complementaria de la elaboración de textos relativos a los métodos numéricos distribuidos bajo licencias igualmente libres.

Invitamos finalmente a todos aquellos lectores que deseen hacer alguna sugerencia sobre este texto o llamar nuestra atención sobre algún error contenido en él, a que se dirijan a nosotros a través de la dirección de correo electrónico `curro@dma.uvigo.es`.

Vigo, Febrero de 2007

Historial de revisiones

- Versión 0.1, Febrero de 2007

  Primera versión del texto con su estructura definitiva. Al margen de las (numerosas) erratas, faltan aún algunos ejemplos y reelaborar las figuras.

- Versión 0.2, Febrero de 2008

  Se han corregido algunas de las (muy numerosas) erratas de la versión 0.1, aún no se han añadido los ejemplos pendientes ni se han reelaborado las figuras.

- Versión 0.3, Febrero de 2009

  Se han corregido unas pocas erratas adicionales.

Capítulo 1

Introducción y necesidad de los métodos numéricos

En este tema se consideran algunas cuestiones previas relativas a los métodos numéricos como son su relación con el modelado matemático, la definición (habitual) de análisis numérico y de métodos numéricos, las fuentes de error en el empleo de los métodos numéricos, así como algunas referencias sobre los códigos informáticos disponibles.

1.1. Modelado matemático

Uno de los aspectos de las Matemáticas con mayor relevancia en el campo científico y tecnológico es la modelización matemática, que trata de establecer modelos formados por un conjunto más o menos complejo de ecuaciones (en los casos más simples se tratará de ecuaciones algebraicas, pero en la mayor parte serán ecuaciones diferenciales) que representen un cierto sistema (comúnmente, se tratará de un sistema físico, pero desde hace tiempo también se extienden esas ideas a los sistemas económicos y, más recientemente, a los sistemas sociales).

La modelización matemática permite entonces predecir el comportamiento de los sistemas, a condición de disponer de un modelo adecuado (una etapa imprescindible será la contrastación del modelo con la realidad) y ser capaces de resolver el modelo matemático propuesto.

Este enfoque se aplica a sistemas con escalas (de complejidad) muy distintas. La descripción del comportamiento eléctrico de una resistencia mediante la ley de Ohm (que permite predecir la caída de tensión o la corriente eléctrica) es un ejemplo de un sistema sencillo. La predicción del clima global (véase, por ejemplo, el texto de J.L. Lions citado en las referencias) es un ejemplo de un sistema de enorme complejidad.

De una u otra forma, la resolución de los modelos matemáticos que representan los sistemas físicos está detrás de la mayor parte de las Matemáticas que se manejan en las enseñanzas técnicas (aunque los modelos terminan por confundirse con la realidad). Aún

más, casi todas las ramas de las Matemáticas nacieron para responder a la manipulación de modelos para el mundo físico y, ciertamente, el papel de las Matemáticas ha sido muy importante en el desarrollo tanto de otras ciencias (especialmente la Física) como en la Tecnología (consúltese, por ejemplo, el texto y la conferencia de J.L. Vázquez que aparecen en las referencias).

Considerando ahora la resolución de las ecuaciones que constituyen el modelo matemático, se observa que la mayor parte de los modelos considerados son, sin embargo, lo suficientemente complejos como para no poder ser resueltos analíticamente.

Una alternativa para poder tratar entonces analíticamente los modelos, empleada muchas veces, es admitir las simplificaciones necesarias para que lo sean (las ventajas de disponer de una solución analítica que permita ilustrar algunas propiedades del modelo puede pesar más, sobre todo desde un punto de vista académico, que los inconvenientes de que el modelo no sea realista). Un caso habitual es la hipótesis de linealidad de los modelos, justificada cuando se trata de analizar el comportamiento *local* de un sistema (se describe el funcionamiento de un sistema muy cerca de un estado de referencia y se aproxima su ley de comportamiento por un desarrollo de Taylor truncado en el primer orden) pero no en otros casos.

Sin embargo, en la mayor parte de las situaciones las hipótesis necesarias para obtener un modelo resoluble analíticamente desvirtúan por completo el modelo y lo hacen inservible con fines prácticos (como ocurre, por ejemplo, con el comportamiento de un diodo en un circuito eléctrico).

Así, en la práctica es absolutamente habitual encontrar problemas que no pueden ser resueltos de forma analítica. Además, en muchos otros casos, se encuentran problemas que son resolubles analíticamente pero para los cuales dicha resolución es inviable desde el punto de vista práctico debido a la complejidad o al volumen de los cálculos implicados. A continuación se muestran algunos ejemplos:

- En teoría de circuitos (recuérdense los contenidos de *Análisis de Redes* o *Dispositivos Electrónicos*) existen muchos componentes para los cuales se supone un comportamiento lineal (como se acaba de mencionar, esta hipótesis puede ser aceptable si consideramos el comportamiento del circuito en un entorno, quizás reducido, de un cierto régimen de funcionamiento pero no tanto para estudiar un régimen más general), de forma que resulta relativamente fácil analizar los circuitos en los cuales sólo interviene este tipo de elementos (siempre y cuando el circuito no contenga un número elevado de componentes). Sin embargo, si introducimos elementos con comportamiento no lineal (piénsese, por ejemplo, en el caso antes citado de un diodo) en general ya no será posible la resolución de dicho circuito. El tema 2 de la asignatura (*Resolución numérica de ecuaciones de una variable*) aborda el estudio de métodos numéricos para resolver de forma aproximada este tipo de problemas.

- Retomando la teoría de circuitos, cuando se consideran circuitos formados exclusivamente por elementos lineales, la resolución de éstos mediante la leyes de Kirchhoff conduce a sistemas de ecuaciones lineales. Si el circuito no es demasiado complejo, estos sistemas pueden ser resueltos *a mano* pero en cuanto el número de elementos

en el circuito crece, los sistemas comienzan a ser inabordables y es preciso contar con técnicas que permitan su resolución (exacta o aproximada) mediante un ordenador. Ni que decir tiene que la resolución de circuitos complejos con elementos no lineales obliga, por doble razón, a emplear métodos numéricos en su resolución. El tema 3 (*Resolución de sistemas de ecuaciones lineales y no lineales*) aborda precisamente dichos métodos.

- El cálculo del índice con el cual el buscador Google ordena las páginas encontradas (que el programa denomina *PageRank*) constituye un problema algebraico interesante. Como puede consultarse en el libro de C. Moler o el artículo divulgativo de P. Fernández mencionados al final de este tema (o en multitud de páginas en Internet), el índice *PageRank* es en realidad un autovector de una matriz asociada a la conectividad de las páginas de Internet. Es fácil imaginar que dada la magnitud del problema, es preciso contar con técnicas computacionales adecuadas para abordar su cálculo. De forma quizás más modesta, el análisis de la estabilidad de los sistemas (lineales o no lineales, en este último caso a través de su linealización) pasa por el cálculo de los autovalores de ciertas matrices asociadas a estos sistemas. Dicho problema, como se sabe, no puede ser resuelto de forma analítica salvo que el sistema sea muy sencillo. En el tema 4 (*Autovalores y autovectores*) se estudiarán algunos métodos numéricos para la aproximación de autovalores (y autovectores asociados) de matrices.

- En la materia *Señales y sistemas analógicos* se ha abordado la representación y manipulación de señales muestreadas. En dichas operaciones interviene a menudo la necesidad de aproximar una señal mediante una función polinómica (ya sea global o a trozos) y obtener aproximaciones de derivadas o integrales donde aparece dicha función (como es el caso del cálculo de convoluciones). Este tipo de problemas (con aplicaciones en muchísimos más ámbitos que el mencionado) serán abordados en los temas 5 (*Interpolación numérica*), 6 (*Derivación numérica*) y 7 (*Integración numérica*) de la asignatura.

Se plantea entonces la necesidad de completar las herramientas analíticas con otras técnicas capaces de resolver de forma aproximada (pero con la posibilidad de cometer errores arbitrariamente pequeños, aunque sea con un coste de cálculo muy elevado) esos problemas y ésa es justamente la misión de los métodos numéricos.

1.2. Análisis numérico y métodos numéricos

Es posible definir los métodos numéricos como aquellos algoritmos que permiten resolver, de forma exacta o aproximada, problemas matemáticos que involucran el cálculo de determinados valores y no pueden ser abordados (o lo son con un coste o complejidad muy elevados) mediante técnicas analíticas.

Ciertamente esta definición es vaga pero, a cambio, permite abarcar todos aquellos métodos que son habitualmente incluidos en el estudio de los métodos numéricos.

Aunque exista básicamente un acuerdo acerca de lo que son los métodos numéricos,

no existe una definición unánimemente aceptada de los contenidos, límites o enfoque del campo de conocimiento dedicado al estudio de los métodos numéricos (como ejemplo, pueden considerarse los muy diversos contenidos, pero sobre todo enfoques, de asignaturas que llevan el título de *Métodos Numéricos*, algo que no ocurre si se consultan materias que lleven el nombre de *Cálculo*). De hecho, existen diferentes nombres para describir el estudio de los métodos numéricos: análisis numérico, cálculo científico y cálculo numérico y, en cierto modo, el empleo de cada uno de ellos está asociado a un determinado enfoque.

La razón de estas diferencias se encuentra en que los métodos numéricos están relacionados con muchas disciplinas. Así, los métodos numéricos guardan, desde luego, relación con las Matemáticas, pero también con muchas otras áreas como la Algorítmica, la Programación, la Ingeniería del Software y, sobre todo, con las diferentes áreas tecnológicas en las cuales la necesidad de resolver problemas intratables mediantes técnicas analíticas ha motivado el desarrollo de los métodos numéricos.

Esta vinculación de los métodos numéricos con distintos ámbitos de conocimiento da lugar a varios posibles enfoques de su estudio, entre los cuales están:

- análisis de las propiedades de convergencia de los métodos numéricos
- utilización eficiente de los métodos numéricos
- estudio de la programación (eficiente) de los métodos numéricos
- aplicación de los métodos numéricos en un determinado campo

El presente curso abordará (parcialmente) los dos primeros enfoques (con especial énfasis en el segundo). El tercero de los aspectos será abordado (de forma introductoria y parcial) en la materia *Laboratorio de Análisis Numérico* y el último será considerado en diversas materias de la titulación.

1.3. Análisis de errores

Los métodos numéricos proporcionan una solución aproximada de los problemas que tratan de resolver (aunque es cierto que algunos de los métodos que se estudiarán en el curso proporcionarían *idealmente* una solución exacta, como se verá a continuación esto no ocurrirá prácticamente nunca una vez que se resuelva, como se hará siempre, mediante la programación de los algoritmos en un ordenador).

Denominaremos error a la diferencia entre la solución que los métodos, una vez programados, devuelven y la solución exacta del problema que se trata de resolver. Este error es debido a dos fuentes bien distintas:

- error de truncamiento (también llamado, a veces, de aproximación)
- error de redondeo

que se presentan brevemente a continuación.

Errores de truncamiento

La finalidad de la mayor parte de los métodos numéricos es proponer algoritmos para resolver de forma aproximada aquellos problemas que no se pueden resolver mediante métodos analíticos (sólo una parte de los métodos -fundamentalmente del álgebra numérica lineal- buscan *idealmente* resolver de forma exacta problemas que pueden ser resueltos exactamente mediante métodos analíticos pero que implican cálculos suficientemente tediosos como para que se prefiera no llevarlos a cabo). Así, lo que proponen estos métodos es una técnica para calcular un valor próximo al de la solución exacta del problema en cuestión y la diferencia entre estos dos valores es lo que (generalmente) denominaremos *error de truncamiento*.

A cualquier método numérico debemos pedirle, desde luego, que ese error de truncamiento pueda hacerse arbitrariamente pequeño (a costa habitualmente de la necesidad de hacer más cálculos). Una parte fundamental del estudio de métodos numéricos es, justamente, el estudio de la dependencia del error de truncamiento con respecto a los datos del problema (pues habrá problemas más o menos difíciles de resolver) y los parámetros del método numérico (como, por ejemplo, el número de iteraciones o el valor de partida en aquellos métodos que calculen un valor mediante aproximaciones sucesivas).

Errores de redondeo

Desde hace bastante tiempo (aunque no siempre ha sido así) resulta impensable resolver cierto problema mediante métodos numéricos llevando a cabo manualmente las operaciones, sino que el algoritmo propuesto por el método se programará en un ordenador. La programación en un ordenador de los métodos numéricos conlleva una fuente de error adicional, ligada a la representación mediante un número finito de bits de las variables numéricas. A este error se le denomina *error de redondeo* y está ligado fundamentalmente al tipo de precisión que se emplee (algo determinado por el procesador y el software usados). Sin embargo, como se verá posteriormente, el efecto final de los errores de redondeo depende también del algoritmo propuesto por el método numérico y por la forma de programarlo.

La representación en coma flotante con doble precisión fijada por la norma ANSI/IEEE (ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic) que es la que emplean, por ejemplo, tanto Octave como MATLAB por defecto, emplea 64 bits para la representación de los números (reales). De ellos, 1 bit corresponde al signo, 11 bits al exponente y los 52 restantes a la mantisa.

Así, la representación normalizada de un número almacenado en doble precisión toma la forma

$$\pm(1+f)\times 2^e$$

donde

$$1+f=(1.a_1a_2a_3...a_{52})_2$$

$$e = -1023 + (b_1 b_2 ... b_{11})_2$$

de forma que

$$1 \leq 1 + f \leq 2 - 2^{-52}$$

$$-1023 \leq e \leq -1023 + 2^{11} - 1 = 1024$$

aunque los valores extremos de e se reservan para representar dos números especiales (el valor inferior para 0 y el valor superior para Inf).

Para el almacenamiento con doble precisión descrito por la norma, se tendrá entonces que:

- el menor número real (positivo) representable (denominado `realmin` en Octave y MATLAB) viene dado por

  $$+(1.000...000)_2 \times 2^{-1022} \simeq 2.2251 \times 10^{-308}$$

  de modo que cualquier resultado más pequeño que esta cantidad originará un error de *underflow* (o será guardado como cero, según el entorno).

- el mayor número real representable (denominado `realmax` en Octave y MATLAB) viene dado por

  $$+(1.111...111)_2 \times 2^{1023} = (2 - 2^{-52}) \times 2^{1023} \simeq 1.7977 \times 10^{308}$$

  y cualquier resultado mayor que dicho número generará un error de *overflow* (o será almacenado como Inf, según el entorno).

- la distancia entre el número 1 (representable de forma exacta) y el número inmediatamente mayor representable de forma exacta (esta distancia se almacena en la constante `eps` en Octave y MATLAB, y habitualmente se denomina *epsilon de la máquina*) es

  $$2^{-52} \simeq 2.2204 \times 10^{-16}$$

De este modo, el simple almacenamiento del resultado de una operación origina un error de redondeo (que, en términos relativos, es inferior a la constante `eps` descrita). El efecto que este error tiene sobre el resultado final del conjunto de operaciones asociado a la programación de un determinado método numérico depende de dos cuestiones:

- de las operaciones que tengan lugar (existen operaciones que son especialmente sensibles a los errores de redondeo), y

- de cómo están organizados los cálculos (pues el algoritmo puede hacer que se amplifiquen o no los errores de redondeo en la evolución de los cálculos).

Así, una operación especialmente *delicada* en relación con los errores de redondeo es la resta de dos cantidades muy próximas, que da lugar a importantes pérdidas de precisión. Esto ocurre porque el resultado contendrá un número de cifras significativas sensiblemente menor que las variables que se operan.

Considérese, por ejemplo, la evaluación (directa) de $f(x) = \sqrt{x^2+1}-1$. Dicha evaluación será imposible si se emplea doble precisión cuando $|x| < \sqrt{\texttt{eps}} \approx 1.5 \times 10^{-8}$ (salvo que se empleen bits de protección) y producirá importantes errores de redondeo para $|x| > 1.5\times 10^{-8}$ pero $|x| << 1$.

A menudo, este tipo de problemas puede resolverse con una correcta reordenación de los cálculos. Así, para el ejemplo anterior, la función puede reescribirse como $f(x) = \frac{x^2}{\sqrt{x^2+1}+1}$ lo que permite que sea calculada sin problemas.

Por esa misma razón, la división por cantidades pequeñas (cuando el resultado final del grupo de operaciones que se van a realizar no es un número muy grande) origina también importantes pérdidas de precisión y debe también evitarse.

Ciertamente, los problemas de pérdida de precisión no son tan graves hoy en día como lo eran hace unos cuantos años, debido al empleo hoy habitual del almacenamiento con 64 bits (y, en algunos casos, al empleo además de bits de protección). Este cambio puede observarse fácilmente por la gran atención prestada a los errores de redondeo en textos de análisis numérico de hace dos o tres décadas. Sin embargo, estas cuestiones tampoco deben ser obviadas sin más pues pueden ser fuentes de errores que no son fácilmente detectables (de hecho son *transparentes* en la compilación). Así, una parte relevante de los errores graves en códigos de cálculo científico se ha debido históricamente a dificultades originadas por los errores de redondeo (y, sobre todo, su propagación a lo largo de los cálculos como se va a considerar a continuación). Se puede consultar una lista de *famosos* errores de programación recopilada por T. Huckle, bajo el título *Collection of Software Bugs*, en la dirección:

`http://www5.in.tum.de/~huckle/bugse.html`

Por otro lado, es muy importante conocer, para un algoritmo dado, cómo puede amplificar los errores de redondeo que sistemáticamente se irán produciendo a lo largo de los cálculos (incluso cuando se ha tenido un gran cuidado en evitar operaciones con importantes pérdidas de precisión).

Considérese (este ejemplo, como el anterior, está tomado del texto de D. Kincaid y W. Cheney incluido en la bibliografía básica de la asignatura) la sucesión definida mediante la fórmula de recurrencia

$$x_{n+1} = \frac{13}{3}x_n - \frac{4}{3}x_{n-1}$$

cuyos primeros términos son $x_0 = 1$ y $x_1 = \frac{1}{3}$. Es sencillo ver que dicha sucesión es

$$x_n = \left(\frac{1}{3}\right)^n \qquad \text{para } n = 1, 2, 3...$$

y se esperaría que la programación de su cálculo empleando la fórmula de recurrencia de acuerdo con el siguiente pseudocódigo:

```
x(0)=1

x(1)=1/3

para n=1,2,3...

    x(n+1)=13/3*x(n)-4/3*x(n-1)

terminar en n
```

devolviese los valores exactos salvo pequeños errores de redondeo. Sin embargo (¡pruébese!), es fácil comprobar que después de unos pocos términos los errores son enormes (y la sucesión numérica generada, inicialmente decreciente, es, a partir de un cierto término, creciente).

La explicación de este comportamiento está relacionada con la forma de la solución general de la ecuación en diferencias, que viene dada por

$$x_n = C_1 \left(\frac{1}{3}\right)^n + C_2 4^n$$

Para la sucesión que deseamos generar se tiene $C_1 = 1$ y $C_2 = 0$ (que vienen dados por la *condición inicial* impuesta por los dos primeros términos). Sin embargo, en la práctica los errores de redondeo son inevitables (x_1 no es, de hecho, representable de forma exacta) y la sucesión de valores generados por el programa se parecerá a una sucesión con un valor reducido de C_2 pero no nulo.

Este tipo de esquemas numéricos (capaces de amplificar enormemente los errores de redondeo) se denominan *esquemas inestables* y deben ser completamente evitados. Una de las preocupaciones del análisis numérico es justamente el diseño de esquemas con buenas propiedades de *estabilidad* ante la inevitable aparición de errores de redondeo. En esta asignatura no se insistirá en general sobre estas cuestiones (que no son fáciles en muchos casos), pero sí se deberá tener en cuenta que el comportamiento de los esquemas numéricos no siempre es obvio y no todos los esquemas (por muy razonables que parezcan en su deducción) se comportan tan bien como se esperaría de ellos en ausencia de errores de redondeo.

Otras fuentes de error

Además del error debido al empleo de los métodos numéricos (que es, como se ha comentado, la diferencia entre la solución que devuelve la programación de un método numérico y la solución exacta del problema que se resuelve) existen otras importantes fuentes del error final, considerado como la diferencia entre la solución que devuelven los métodos numéricos y la magnitud física (o económica o social, si es el caso) que deseamos calcular. Estas fuentes de error son:

- errores de modelado
- errores de medición de parámetros del modelo

donde los primeros errores están relacionados con la elección de un modelo (habitualmente dentro de una gama, que va desde modelos muy groseros que tienen en cuenta sólo los aspectos fundamentales de los procesos que se tratan de describir hasta modelos muy refinados que tienen en cuenta todos los posibles aspectos implicados) que represente adecuadamente el sistema en estudio, y los segundos con la medida (o estimación si no pueden ser directamente medidos) de todas las propiedades del sistema que aparecen en el modelo.

Así, considérese el cálculo de la corriente que circula (o cualquier otra variable de interés) por un determinado circuito eléctrico más o menos complejo. La primera etapa consistirá en la representación del sistema por un modelo que, en este caso, estará formado por un conjunto de elementos eléctricos con unas determinadas leyes de comportamiento. Las leyes de comportamiento pueden ser muy simples o muy complejas, dependiendo del detalle con el que queramos representar el comportamiento; por ejemplo, para una resistencia podemos elegir una ley lineal con una resistencia fija (como correspondería a la ley de Ohm) o una ley más general (podríamos representar la caída de potencial como un polinomio de un grado más o menos elevado de la corriente que circula a través de ella, con coeficientes dependientes de la temperatura, que a su vez deberá ser calculada en el modelo).

En una segunda etapa, se deberán medir o estimar los parámetros que aparecen en el modelo, como los coeficientes contenidos en las leyes de comportamiento o la tensión a la que se somete el circuito (si es éste el caso). Dependiendo de la precisión de los equipos con los que se trabaja y del número de medidas que se tomen (en el caso de leyes más complejas donde se deba ajustar una cierta función, como ocurre en la generalización de la ley de Ohm mencionada), aparecerán nuevos errores que serán más o menos importantes y que se añadirán al error de modelado.

Aunque estas dos fuentes de error (el error de modelado y el error de medida de los parámetros del modelo) sean por completo ajenas a los métodos numéricos, su conocimiento es muy importante a la hora del empleo de los métodos numéricos. En particular, no presenta ningún interés práctico la resolución con enorme precisión de un cierto problema asociado a un modelo grosero (el error final estará completamente condicionado por los errores de modelado y el coste de cálculo podría dispararse). De igual manera, resulta inadecuado (salvo que sea técnicamente inevitable) resolver con muy poca precisión un problema para el cual se ha asegurado una minuciosa elección del modelo adecuado y se han estimado sus parámetros con gran cuidado (pues el error final estará ahora determinado por los errores introducidos por el método numérico, lo que habrá convertido en inútil el esfuerzo previo en la etapa de modelado).

Volviendo al ejemplo anterior, es fácil comprender que no es sensato tratar de calcular la corriente que recorre cierta rama del circuito en cuestión con enorme precisión si se está resolviendo un modelo con leyes de comportamiento muy simples, aún cuando se sabe que -por ejemplo- las corrientes son muy elevadas y provocan un comportamiento altamente no lineal de los elementos, y los equipos empleados en la medida de las propiedades de los

elementos no son muy precisos. Asimismo, tampoco parece procedente calcular con escasa precisión esa misma corriente a partir de un modelo muy elaborado (que incluye multitud de dependencias de los parámetros de las leyes de comportamiento) que ha implicado además medidas muy precisas de los parámetros.

Medida del error

Existen (básicamente) dos formas de medir los errores que se cometen en la aproximación de la solución de un cierto problema mediante un método numérico (ya nos estemos refiriendo al error *ideal* que se cometería en ausencia de errores de redondeo o al error *total*, una vez considerados éstos):

- error absoluto
- error relativo

En el primer caso, el error se mide con las mismas unidades que la variable que se trata de aproximar (por ejemplo, en amperios si se quiere aproximar la corriente que circula por determinado circuito). Así, si denominamos x a la solución de un cierto problema que deseamos resolver numéricamente y x_{num} al valor obtenido empleando un método numérico, el error absoluto e_{abs} se construirá como

$$e_{abs} = |x - x_{num}|$$

Desde luego, dicha medida puede resultar útil en ciertas situaciones (por ejemplo en la resolución de un problema particular, donde ya conocemos el orden de magnitud de la solución) pero no en muchas otras (fundamentalmente, cuando queremos estudiar las propiedades del método para un rango amplio de problemas) donde el error será aceptable o no, dependiendo de su comparación con el orden de magnitud de la propia solución. De esta forma, se definirá un error relativo (existen otras posibilidades) como

$$e_{rel} = \frac{|x - x_{num}|}{|x|}$$

aunque la definición será modificada en la práctica para tratar aquellos casos donde x pueda tomar valores en un rango que incluya el valor cero.

En la formulación del error descrita se ha considerado que x representa una variable escalar. Sin embargo, en muchos problemas los métodos numéricos tratan de aproximar vectores (por ejemplo en la resolución de sistemas de ecuaciones, lineales o no lineales) o funciones (por ejemplo, en los problemas de interpolación). En estos casos, se deberá sustituir el valor absoluto (que constituye una norma para el espacio vectorial asociado a $\mathbf{R}$) por una norma adecuada.

Cuando se trata con vectores, existen tres medidas habituales en la construcción de los errores absolutos (los errores relativos se construirán de igual modo). Denotando mediante $\vec{x}$ el vector (de $\mathbf{R}^n$) solución exacta del problema y $\vec{x}_{num}$ el vector que devuelve la aproximación mediante un cierto método numérico, los errores correspondientes a cada una de estas tres medidas se expresan como

$$||\vec{x} - \vec{x}_{num}||_1 = \sum_{i=1}^{n} |x_i - x_{num,i}|$$

$$||\vec{x} - \vec{x}_{num}||_2 = \sqrt{\sum_{i=1}^{n} (x_i - x_{num,i})^2}$$

$$||\vec{x} - \vec{x}_{num}||_\infty = \max_{1 \leq i \leq n} |x_i - x_{num,i}|$$

Estas tres medidas devuelven, en general, valores numéricos distintos y, de hecho, consideran diferentes *pesos* para construir el error total cometido en la aproximación del vector a partir de los errores en cada componente de dicho vector. Sin embargo, ofrecen básicamente la misma información acerca de la precisión lograda por la aproximación numérica. Esto es así porque para cada pareja de normas $||\cdot||_a$ y $||\cdot||_b$ elegidas entre las anteriores (o, de hecho, para cualquier pareja de normas en $\mathbf{R}^n$) existen unas constantes positivas k_1 y k_2 tales que

$$k_1||x||_a \leq ||x||_b \leq k_2||x||_a \qquad \forall\, x \in \mathbf{R}^n$$

(cuando dos normas verifican esta propiedad se dice que son equivalentes).

Cuando se trata de aproximar funciones, la medida del error *global* (obviamente, también existe la posiblidad de medir el error en un solo punto o en un conjunto finito de puntos, reduciendo entonces el problema a la aproximación de un escalar o un vector) es algo más compleja pero existen, entre otras, las medidas que tratan de adaptar las normas vectoriales descritas anteriormente al caso de una función.

Así, sea $f : [a, b] \to \mathbf{R}$ una cierta función y consideremos la medida del error que se comete si se aproxima por la función $f_{num} : [a, b] \to \mathbf{R}$. Suponiendo la regularidad necesaria de ambas funciones, se podría medir el error absoluto mediante

$$||f - f_{num}||_1 = \int_a^b |f(x) - f_{num}(x)|\, dx$$

$$||f - f_{num}||_2 = \sqrt{\int_a^b (f(x) - f_{num}(x))^2\, dx}$$

$$||f - f_{num}||_\infty = \sup_{a \leq x \leq b} |f(x) - f_{num}(x)|$$

A diferencia de lo que ocurría en el caso vectorial, estas normas ya no son equivalentes y las diferentes medidas ya no tienen por qué devolver la misma información sobre la calidad de la aproximación.

Obsérvese, por último, que la definición presentada de los errores es interesante desde el punto de vista teórico (a fin de estudiar las propiedades de un determinado método numérico) pero no tanto desde el punto de vista práctico. De hecho, es obvio que en la práctica no dispondremos de la solución exacta de nuestro problema para poder medir el error comparándola con la aproximación numérica (si dispusiésemos de la solución exacta, ¿para qué plantearnos dicha aproximación?).

Así, en la práctica es preciso construir medidas *indirectas* del error. Una medida habitual es lo que se denomina en términos generales el *residuo* del problema que se desea resolver.

Con cierta vaguedad (bastante, en realidad) podemos definir el residuo como la cantidad no nula, en general, que queda en la formulación exacta de nuestro problema si sustituimos la solución calculada numérica en el lugar que ocupa la solución.

Retomando un ejemplo empleado varias veces a lo largo de la presentación, en el cálculo numérico de la corriente que circula por determinado circuito eléctrico se plantearán unas ciertas ecuaciones a partir de las leyes de Kirchhoff. Una de estas ecuaciones puede representar, por ejemplo, el sumatorio de las corrientes en un nodo del circuito. Es claro que la solución exacta del problema hace que dicha suma sea igual a cero, pero de la solución numérica sólo se espera que tome un valor muy reducido, al que denominaremos residuo. Este residuo reflejará, según sea menor o mayor, lo cerca o lejos que está la solución numérica de la solución exacta (aunque, como se verá posteriormente al definir el condicionamiento de un cierto problema, es preciso tomar con cuidado esta interpretación).

Para la construcción de una medida para los residuos cabe también distinguir entre medidas absolutas y relativas, como se ha hecho para los errores. Asimismo, se deben considerar medidas que tengan en cuenta si se desea estimar un escalar, un vector o una función.

1.4. Códigos de métodos numéricos

A continuación, se hace una presentación de algunos códigos informáticos disponibles que incorporan métodos numéricos (y, en algunos casos, sirven también de plataforma de desarrollo de otros). Esta presentación no tiene intención, en absoluto, de ser exhaustiva. Por el contrario, busca simplemente presentar algunas herramientas y hacer algunos comentarios generales sobre el tipo de herramientas disponibles.

Entornos de programación

Se describen aquí algunos entornos que incluyen un lenguaje de alto nivel interpretado junto con una biblioteca más o menos amplia de métodos ya programados:

- OCTAVE (`http://www.octave.org`), incorpora subrutinas de diversos paquetes de dominio público y existe además una colección de funciones escritas por usuarios en `http://octave.sourceforge.net/`

- SCILAB (`http://www.scilab.org/`), incorpora numerosas subrutinas tanto de paquetes de dominio público como de autores que colaboran con el proyecto. Existe un proyecto relacionado, Scicos (`http://www.scicos.org/`), que desarrolla un simulador de sistemas a partir de diagramas de bloque.

- MATLAB (`http://www.mathworks.com/`), entorno similar a OCTAVE y SCILAB (en gran medida los tres entornos son compatibles; aunque SCILAB se separa algo más de los otros dos, dispone de un conversor) que incorpora además numerosos paquetes con aplicaciones concretas en multitud de dominios.

Los dos primeros entornos son códigos de software libre (el primero distribuido bajo licencia GPL y el segundo con una licencia similar) en tanto que el tercero es un (caro) código comercial.

Un entorno hasta cierto punto similar (y que recibe una atención creciente en el dominio del cálculo científico) lo constituyen Python, NumPy (Numerical Python) y SciPy (Scientific Python). Python (`http://www.python.org/`) es un lenguaje de alto nivel interpretado, orientado a objeto y cuyo intérprete se distribuye como software libre. NumPy es una biblioteca de funciones de Python para la manipulación de grandes estructuras de datos (arrays) y SciPy (`http://www.scipy.org/`), por su parte, es una bibliotecas de subrutinas de Python para cálculo científico basada en NumPy.

Adicionalmente, existen herramientas como SWIG (`http://www.swig.org/`), Boost.Python (`http://www.boost.org/libs/python/`), F2PY (`http://cens.ioc.ee/projects/f2py2e/`) o Wave (`http://www.scipy.org/Weave`) que permiten hacer fácilmente accesibles desde Python programas escritos en C, C++ o fortran.

Bibliotecas de subrutinas

A continuación se ofrece una lista de bibliotecas que contienen programas que implementan diversos métodos numéricos.

- La biblioteca NetLib (`http://www.netlib.org`) constituye la colección más completa de códigos libremente disponibles (aunque algunos de ellos sí imponen algunas restricciones a su uso)
- El portal GAMS (`http://gams.nist.gov/`) proporciona información y enlaces relativos a códigos disponibles (ya sean libres o comerciales) en el campo de las matemáticas
- La biblioteca GNU Scientific Library (GSL) es una colección de subrutinas escritas en C y C++ distribuida bajo licencia GPL y disponible en:

  `http://www.gnu.org/software/gsl/`

- La biblioteca NAG Library of Numerical Algorithms (`http://www.nag.com/`), que constituye una colección comercial de subrutinas en Fortran, C y C++
- La biblioteca IMSL Numerical Library (`http://www.vni.com/products/imsl/`) es una colección comercial de subrutinas en Fortran, C y Java

Otras subrutinas disponibles

Por otro lado, muchos libros sobre métodos numéricos incluyen programas sencillos que muestran la implementación de diversos métodos. Así, el texto

- J.H. Mathews; K.D. Fink; *Métodos Numéricos con MATLAB*, 3a ed. Prentice-Hall, 2000.

desarrolla numerosos programas en MATLAB para los métodos descritos en el libro, que se pueden encontrar en la dirección:

`http://math.fullerton.edu/mathews/books/nbook.htm`

También el libro

- J.D. Faires; R. Burden; *Métodos Numéricos*, 3a ed. Thomson, 2004.

incluye un CD-ROM con archivos fuente en C, Fortran y Pascal, y archivos de Maple, Mathematica y MATLAB correspondientes a los ejemplos desarrollados en el libro. Estos archivos también están accesibles en la dirección

`http://www.as.ysu.edu/~faires/Numerical-Methods/DiskMaterial/index.html`

Asimismo, las ediciones originales del texto de D. Kincaid y W. Cheney *Numerical Analysis: Mathematics of Scientific Computing*, publicadas por Brooks/Cole, contienen programas (escritos en MATLAB, MATHEMATICA y FORTRAN77) ilustrativos de los métodos expuestos en el texto. Los correspondientes archivos pueden descargarse desde la página personal de Ward Cheney:

`http://www.ma.utexas.edu/users/cheney/`

siguiendo el enlace indicado para el texto *Numerical Analysis*.

En todo caso, es preciso aclarar que la finalidad de estos programas incluidos en libros de texto no es, en absoluto, su empleo para resolver problemas prácticos (a diferencia de los programas contenidos en las bibliotecas descritas en el apartado anterior) sino puramente académica. En particular resultan interesantes para aclarar las principales características del comportamiento de los distintos métodos numéricos al tratar ciertas clases de problemas.

Con un carácter intermedio entre los dos tipos de códigos descritos previamente se encuentra la colección de textos y programas *Numerical Recipes in ....* Se trata de una colección bastante extensa de programas (disponibles Fortran, C, C++ y Pascal) que implementan algoritmos numéricos más o menos básicos. Aunque se trata de programas algo más refinados que los ejemplos elementales que suelen acompañar a los libros de texto y pueden, en consecuencia, ser útiles en diversas ocasiones, es preciso aclarar que, en general, se encuentran muy lejos de la eficiencia, fiabilidad y robustez de los códigos contenidos en las bibliotecas descritas (desarrollados en muchos casos por los principales especialistas, y probados y discutidos largamente por la comunidad científica).

Más información sobre códigos

Puede encontrarse más información sobre los programas disponibles en:

- M.T. Heath; *Scientific Computing: An Introductory Survey*, 2nd ed. McGraw-Hill, 2002

disponible en:

`http://www.cse.uiuc.edu/heath/scicomp/software.html`

y para el caso específico de códigos libres disponibles para GNU/Linux en:

- M. Warrier; S. Deshpande; V.S.Ashoka; *Scientific Computing with Free software on GNU/Linux HOWTO*,

disponible, por ejemplo, en:

`http://www.tldp.org/HOWTO/Scientific-Computing-with-GNU-Linux/`

Finalmente, pueden encontrarse numerosos enlaces a recursos interesantes en la página mantenida por H. Greenside disponible en:

`http://www.phy.duke.edu/~hsg/sci-computing.html`

1.5. Referencias

- **J. Hoffman; C. Johnson; A. Logg; *Dreams of Calculus*. Springer, 2004.**

  Este ameno texto se encuentra disponible, junto con otros materiales sobre la enseñanza de las matemáticas, en la página

  `http://www.phi.chalmers.se/bodysoul/books/`

  y contiene interesantes reflexiones sobre la conveniencia de reformar la enseñanza del cálculo para dotarle de un enfoque más ligado a la computación.

- **J.L. Lions; *El planeta Tierra. El papel de las Matemáticas y de los superordenadores.* Espasa, 1990.**

  Este libro, escrito por una de las figuras más relevantes de la Matemática Aplicada de los últimos años, contiene reflexiones muy interesantes de carácter general sobre la modelización matemática, así como una presentación de los retos que plantea la modelización matemática de un sistema de enorme complejidad como es el clima global.

- **J.L. Vázquez; *The importance of Mathematics in the development of Science and Tehcnology*. Bol. Soc. Esp. Mat. Apl. 19 (2001), 69-112.**

  Tanto este texto como una versión posterior en castellano están disponibles en la página del autor

  `http://www.uam.es/personal_pdi/ciencias/jvazquez/jlvescr.html`

Además se puede consultar la grabación de la conferencia de este mismo autor en la E.T.S.I. de Telecomunicación, el día 2 de mayo de 2005, titulada *La comunidad científica y su incidencia en la sociedad. El papel de las Matemáticas* en la dirección

`http://tv.uvigo.es/VODpublic/ASX/ciencia/20050502.asx`

- **C. Moler; *Numerical Computing with MATLAB*. SIAM, 2004.**

  Este libro (que constituye una presentación muy amena y útil de las aplicaciones de MATLAB) puede consultarse en la dirección

  `http://www.mathworks.com/moler/`

  En el capítulo 2 (*Linear Equations*), presenta la aplicación del índice *PageRank* a un modelo sencillo de red.

- **P. Fernández; *El secreto de Google y el Álgebra lineal*. Bol. Soc. Esp. Mat. Apl. 30 (2004), 115-141.**

  Este artículo divulgativo analiza los elementos matemáticos que intervienen en la formulación de índice *PageRank*.

Capítulo 2

Resolución numérica de ecuaciones de una variable

2.1. Motivación

Como se ha visto en el tema anterior, existen multitud de ejemplos de problemas que conducen a la resolución de ecuaciones escalares y deben ser abordados mediante métodos numéricos. Una fuente de este tipo de problemas es el análisis de circuitos. Así, el estudio de elementos realistas lleva a considerar comportamientos no lineales que impiden resolver analíticamente dichos elementos. A continuación se muestran dos ejemplos.

Ejemplo 2.1 *Un primer ejemplo lo constituye el comportamiento de resistencias (resistores) para valores elevados de la corriente (donde la saturación de las bandas de conducción conduce a una desviación con respecto al comportamiento lineal).*

Un ajuste polinómico de la ley de comportamiento llevará a leyes del tipo siguiente (obsérvese que debido a la simetría del dispositivo sólo aparecerán términos impares):

$$u_R(i) = a\,i + b\,i^3 + c\,i^5 + \cdots$$

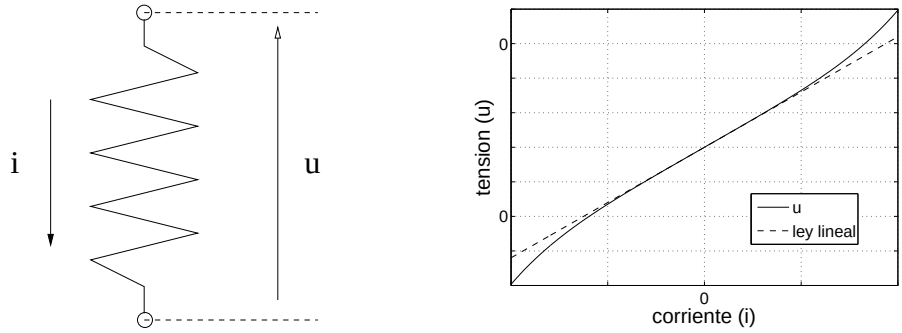

Figura 2.1: Esquema y ley de comportamiento de una resistencia

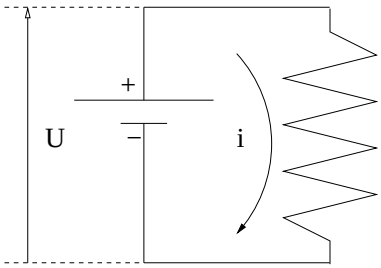

Figura 2.2: Circuito elemental asociado a una resistencia

La figura 2.1 muestra el esquema de una resistencia junto con una ley de comportamiento realista y su comparación con una ley linealizada.

Así, la resolución del circuito más simple (como conectar esta resistencia a una fuente de tensión, tal y como se muestra en la figura 2.2) ya obliga a resolver una ecuación no lineal. Por ejemplo, si retenemos un ajuste polinómico con 3 términos, la corriente i que circula por el circuito será solución de la ecuación:

$$U = a\,i + b\,i^3 + c\,i^5$$

□

Ejemplo 2.2 *Un segundo ejemplo resulta el comportamiento de diodos, que idealmente son descritos mediante una ley que no permite el paso de corriente cuando se polarizan de modo inverso y no origina ninguna caída de tensión cuando la corriente circula en sentido directo (véase la figura 2.3), pero que en la práctica deben ser tratados mediante modelos más realistas. Así, para tensiones de polarización elevadas es preciso tener en cuenta los fenómenos que ocurren tanto en polarización directa como inversa (véase la figura 2.4, observando que las tensiones de polarización directa e inversa no están a la misma escala)*

Uno de los circuitos equivalentes más simples para modelar un diodo (de tipo unión P-N) emplea una fuente de corriente no lineal de la forma

$$i = i_s(exp(\frac{u_D}{u_T}) - 1)$$

donde $u_T = kT/q$ es un parámetro denominado tensión térmica ($\simeq 26 \times 10^{-3}\,V$ a temperatura ambiente) e i_s representa la corriente de saturación. La figura 2.5 ilustra esta ley de comportamiento y su comparación con el esquema de una ley realista (en la figura 2.4) pone de manifiesto que este modelo es adecuado salvo para tensiones de polarización inversa muy elevadas.

Al igual que ocurría con la resistencia, incluso los circuitos más simples que incluyen este elemento conducen a ecuaciones que no podemos resolver analíticamente. De este modo, si

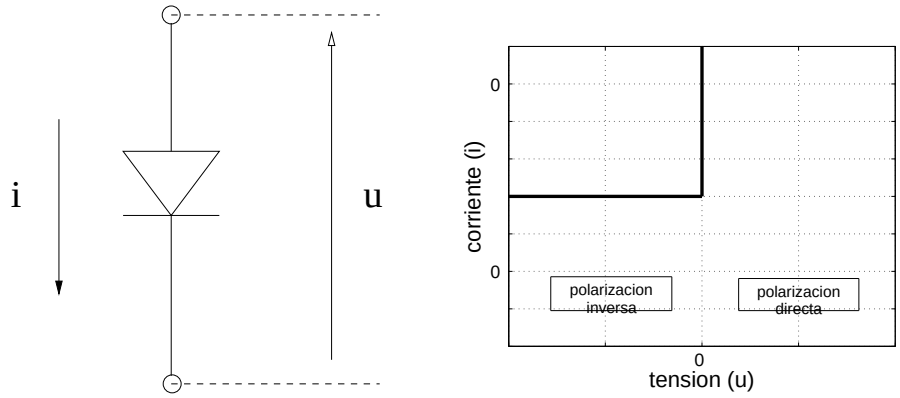

Figura 2.3: Esquema y ley de comportamiento ideal de un diodo

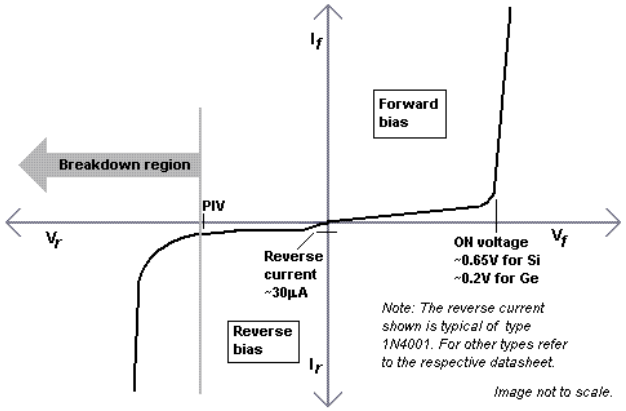

Figura 2.4: Ley de comportamiento real de un diodo

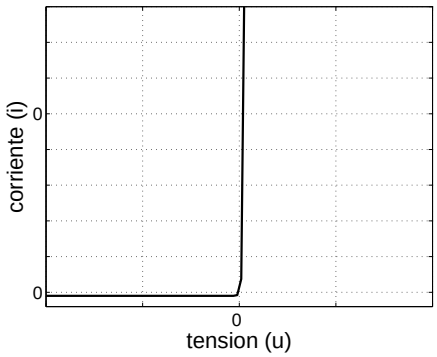

Figura 2.5: Ley de comportamiento simple de un diodo

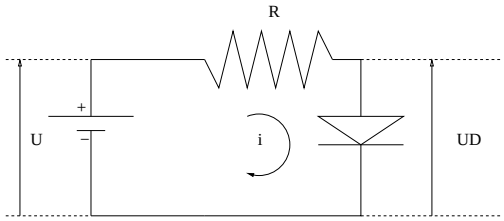

Figura 2.6: Circuito elemental asociado a un diodo

se considera la conexión (en serie) de un diodo y una resistencia a una fuente de tensión, tal y como aparece en la figura 2.6, deberá resolverse (si se emplea la ley de comportamiento que se acaba de describir) el sistema de ecuaciones

$$U = R\,i + u_D$$
$$i = i_s(e^{u_D/u_T} - 1)$$

y, por tanto, eliminando la corriente i se busca la tensión en bornas del diodo (u_D) solución de la ecuación

$$U = R\,i_s(e^{u_D/u_T} - 1) + u_D$$

□

Desde un punto de vista formal, los problemas anteriores pueden escribirse como:

Dada una cierta función $f : A \subset \mathbf{R} \to \mathbf{R}$: encontrar $x \in A$ tal que $f(x) = 0$

donde las funciones f tienen expresiones muy sencillas y pueden entenderse como definidas en todo $\mathbf{R}$ o, más comúnmente, sobre un cierto intervalo donde sabemos que se encuentra la solución (es ésta última una información con la que siempre debemos contar en la práctica cuando se emplean métodos numéricos, ya que a éstos se les debe confiar exclusivamente la tarea de calcular con cierta precisión un valor del que conocemos una estimación más o menos grosera).

Cabe, no obstante, observar que muchos otros problemas se plantearán en el campo complejo. Así, por ejemplo, la búsqueda de polos o ceros de una determinada función de transferencia conduce a la búsqueda de las raíces (posiblemente complejas) de unos ciertos polinomios. También el análisis de circuitos en corriente alterna, donde aparecen ahora coeficientes complejos (aunque se debe tener cuidado con el empleo del análisis armónico en el caso no lineal), llevará a operar con raíces complejas.

Por otro lado, no siempre f admite una escritura analítica *tratable* y, de hecho, esto condiciona el tipo de métodos que se pueden emplear en la práctica. De este modo, podemos

considerar el diseño de un determinado circuito lo bastante complicado como para confiar a un determinado código (SPICE, por ejemplo) su resolución. En dicho diseño se busca ajustar el valor de un único parámetro de diseño (por ejemplo, el valor de una resistencia) con la condición de que el circuito satisfaga una cierta especificación (por ejemplo, una cierta diferencia de tensión entre dos puntos del circuito). Así, denotando mediante x el valor de dicho parámetro (resistencia) y c el valor especificado para la correspondiente variable (diferencia de tensión), se puede escribir el problema como

$$\boxed{\textit{Encontrar } x \in A \textit{ tal que } f(x) = G(x) - c = 0}$$

donde $G(x)$ representa el valor que devuelve el código para la variable que se quiere controlar (en el ejemplo, la diferencia de tensión calculada por SPICE para una cierta configuración del circuito) cuando la variable de diseño (en el ejemplo, la resistencia) toma el valor x y, por su parte, c representa el valor deseado para la variable que se quiere controlar (la diferencia de tensión que se busca en el diseño). Obsérvese que aunque el problema puede ser llevado a una forma similar a la de los ejemplos antes mencionados (y, de este modo, se podrán emplear los mismos métodos, a condición de que éstos no necesiten calcular más de la función f) la evaluación de la función f requiere la ejecución de un código. De forma similar, pueden formularse problemas donde la evaluación de este función requiera, por ejemplo, la realización de un experimento y observación de su resultado.

Condicionamiento del problema

Una cuestión de enorme importancia en todos los problemas que se resuelven numéricamente es la sensibilidad de la solución a pequeñas perturbaciones de los datos. La razón está en que en la práctica se resuelve un problema ligeramente modificado debido tanto a los errores de truncamiento como a los errores de redondeo (adicionalmente, deberá también tenerse en cuenta que los errores de modelado y medición juegan un papel semejante).

Considérese entonces un cierto problema:

$$\boxed{\textit{Encontrar } x \textit{ solución de: } f(x) = 0}$$

y un problema perturbado

$$\boxed{\textit{Encontrar } x \textit{ solución de: } f(x) + \delta f = 0}$$

donde δf representa un número pequeño (en comparación con los valores característicos de f) que representa la *perturbación* de la función que describe el problema asociado a ciertos errores de redondeo, de modelado o de medición.

Obsérvese que, en la práctica, siempre se resolverá un problema perturbado. Así,volviendo a los dos ejemplos presentados anteriormente, será inevitable que aparezcan ciertas perturbaciones:

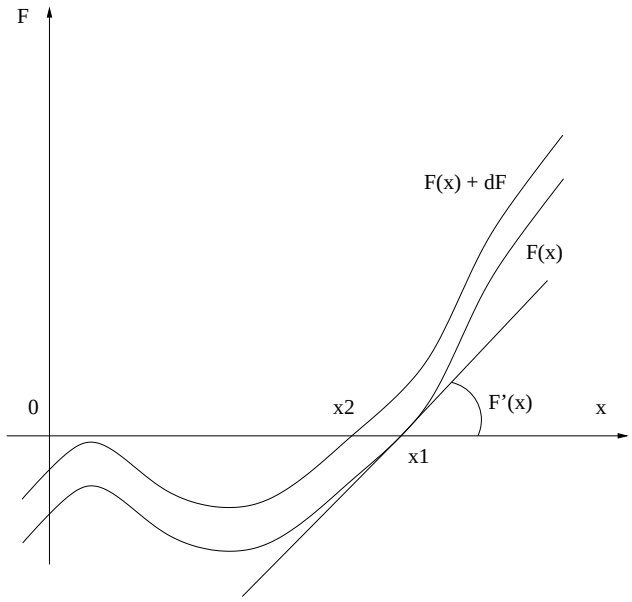

Figura 2.7: Condicionamiento del cálculo de raíces

- provenientes del truncamiento asociado a la precisión finita del ordenador sobre el que trate de resolverse el problema
- originadas por las simplificaciones empleadas en la elaboración de las leyes de comportamiento
- asociadas a los errores de medición de las propiedades de los elementos del circuito (suponiendo que las leyes de comportamiento fuesen rigurosamente exactas)

Generalmente, las soluciones de ambos problemas (el problema original y el problema perturbado) serán diferentes (tal y como se observa en la figura 2.7). Denotaremos mediante x_1 la solución del problema original y mediante x_2 la solución del problema perturbado, de modo que:

$$f(x_1) = 0 \qquad \text{y} \qquad f(x_2) + \delta f = 0$$

Escribiendo $x_2 = x_1 + \delta x$ y desarrollando f (suponiendo la regularidad suficiente) en el problema perturbado se obtiene

$$f(x_1) + f'(x_1)\,\delta x + \mathcal{O}((\delta x)^2) + \delta f = 0$$

de modo que, empleando $f(x_1) = 0$ y despreciando los términos de segundo orden:

$$\delta x \simeq -\frac{1}{f'(x_1)}\,\delta f$$

De este modo, la *amplificación* del error (relación entre la modificación de la solución del problema y el tamaño de la perturbación) viene caracterizada por el factor $-1/f'(x_1)$

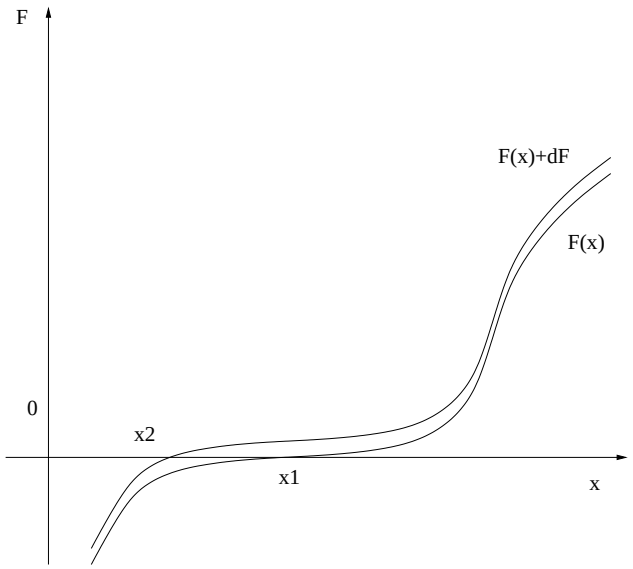

Figura 2.8: Problema muy mal condicionado

(véase la figura 2.7) y el problema estará muy mal condicionado si $|f'(x_1)| \simeq 0$ ya que los distintos errores que intervienen (de truncamiento, de redondeo, de modelado o de medición) originarán que la solución numérica obtenida esté muy lejos del valor buscado (obsérvese la enorme diferencia entre las raíces x_1 y x_2 en la figura 2.8 a pesar de que la perturbación δf no es muy grande).

En el límite (con $f'(x_1) = 0$) puede ocurrir incluso (basta con que $f''(x_1) \neq 0$, que corresponde al caso de una raíz doble) que las perturbaciones alteren el número de raíces (véase la figura 2.9, donde dependiendo del signo de la perturbación el problema puede pasar a tener dos raíces reales distintas o ninguna).

Métodos para ecuaciones algebraicas

En la presentación que se hace a continuación se considerarán métodos generales para la resolución de ecuaciones de una variable. Sin embargo, existen métodos específicos para determinados tipos de problemas y, en particular, métodos adaptados al cálculo de raíces de polinomios. Estos métodos aprovechan ciertas características específicas del cálculo de raíces de polinomios (como las técnicas de deflación), organizan de forma eficiente la evaluación de la función (como el empleo del algoritmo de Horner para minimizar el número de cálculos y el efecto de los errores de redondeo, así como para la evaluación sencilla de la derivada en la formulación del método de Newton-Raphson) o incluso se trata de métodos que sirven exclusivamente para el cálculo de raíces de polinomios. En este (breve) curso no se considerarán este tipo de métodos. No obstante, se verá posteriormente que es posible convertir el problema de cálculo de raíces de un polinomio en el cálculo de autovalores de una cierta matriz (con tal de que ésta tenga a aquel como su polinomio característico) de modo que los métodos que se presentarán para el cálculo de autovalores de matrices servirán para la

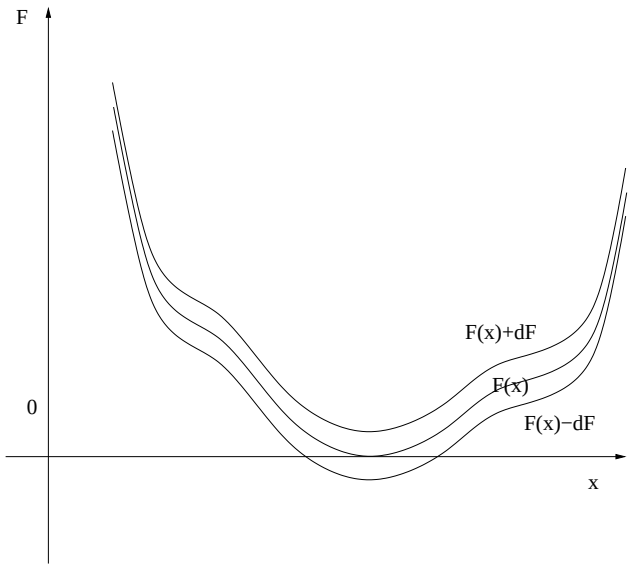

Figura 2.9: Problema infinitamente mal condicionado

obtención de raíces de polinomios.

2.2. Separación de raíces (reales)

Formulado de modo abstracto el problema, como la búsqueda de un número (real, aunque en algunos casos se buscarán también soluciones complejas) x_* tal que

$$f(x_*) = 0$$

siendo $f : A \subset \mathbf{R} \to \mathbf{R}$, se denominará raíz de f a dicho número y se dirá que es una raíz separada si es la única raíz en un cierto conjunto abierto B.

Una tarea importante previa al empleo de los métodos numéricos es justamente la *separación* (o localización) de raíces. Se trata de una tarea fundamental ya que de otro modo es difícil que los métodos numéricos puedan converger (en algún caso, como en el método de bisección, no podrían ni siquiera arrancar). Además, desde el punto de vista práctico, el empleo de los métodos numéricos cuando se desconoce el valor aproximado (aunque sea groseramente) de la solución es un ejercicio muy arriesgado (y completamente desaconsejado).

Un ingrediente fundamental en la separación o localización de raíces lo constituye el conocimiento del propio problema que se desea resolver. Si se retoman, por ejemplo, los casos presentados en la sección anterior (circuitos asociados a una resistencia no lineal y un diodo) es fácil estimar dónde buscar la solución:

- en el primer caso, el aumento de la resistencia debido a la saturación impone claramente

que i_*, solución del problema, verifica

$$0 < i_* < \frac{U}{a}$$

y, de hecho, estará cerca del límite superior si b y c son pequeños así como si $\frac{U}{a}$ es pequeño, ya que en ambos casos la corrección del modelo lineal será poco importante

- en el segundo ejemplo (formulado en tensiones) es inmediato deducir que u_{D*} cumple

$$0 < u_{D*} < U$$

donde u_{D*} además estará cerca del límite inferior si U/R es reducido

En algunas situaciones, sin embargo, puede ser necesario emplear además otros métodos para refinar la separación de raíces. Entre éstos están:

- métodos gráficos, que permiten –a partir de la representación de la gráfica de la función– detectar un intervalo con un cambio de signo de la función (que encerrará por lo tanto alguna raíz si la función es regular)

- métodos analíticos, que emplean las propiedades de la función f y sus derivadas para obtener alguna conclusión sobre el número de raíces en un cierto intervalo

Dentro de los métodos analíticos, resultan muy útiles los teoremas de Bolzano y Rolle (ya estudiados previamente en Cálculo):

Teorema 2.1 *(Bolzano)*
Sea $f : [a, b] \to \mathbf{R}$ *continua en* $[a, b]$. *Supongamos que* $f(a)f(b) < 0$. *Entonces existe al menos un punto* $c \in (a, b)$ *tal que* $f(c) = 0$.

Demostración: Véase, por ejemplo, el texto *Cálculo Infinitesimal de una variable* de J. de Burgos. McGraw-Hill, 1994 (páginas 148–149).

□

Teorema 2.2 *(Rolle)*
Sea $f : [a, b] \to \mathbf{R}$ *continua en* $[a, b]$ *y derivable en* (a, b). *Supongamos que* $f(a) = f(b)$. *Entonces existe al menos un punto* $c \in (a, b)$ *tal que* $f'(c) = 0$.

Demostración: Véase, por ejemplo, el texto *Cálculo Infinitesimal de una variable* de J. de Burgos. McGraw-Hill, 1994 (páginas 209–210).

□

Adicionalmente, las propiedades de monotonía de la función permitirán estudiar la unicidad de la raíz.

Proposición 2.1 *Sea $f : [a, b] \to \mathbf{R}$ continua en $[a, b]$ y derivable en (a, b). Supongamos que $f'(x) \neq 0,\ \forall x \in (a, b)$. Entonces existe a lo sumo una raíz de f en (a, b).*

Demostración: Por reducción al absurdo aplicando el teorema de Rolle.

□

Ejemplo 2.3 *Así, definiendo*

$$f(u_D) = U - R\, i_s\, (exp(u_D/u_T) - 1) - u_D$$

para el problema asociado al diodo, es fácil comprobar que.

$$f(0) = U > 0, \quad f(U) = U - R\, i_s (exp(u_D/u_T) - 1) - U < 0$$

de modo que la continuidad de f asegura la existencia de una raíz en $(0, U)$.

Además se tiene:

$$f'(u_D) = -R\frac{i_s}{u_T} exp(u_D/u_T) - 1 < 0$$

de modo que esta solución es única en $(0, U)$ (y también en todo **R).**

Nota 2.1 *En el caso particular de la búsqueda de raíces de polinomios existen numerosos resultados que permiten acotar sus raíces reales. En el texto de Kincaid-Cheney pueden consultarse algunos de ellos.*

2.3. Método de bisección

Se trata de una aplicación directa del teorema de Bolzano (y corresponde, además, en cierto modo, a una búsqueda *gráfica* elemental). El método propone dividir repetidamente un intervalo con cambio de signo de la función en dos partes iguales, reteniendo aquella que conserve el cambio de signo. El algoritmo proseguiría hasta que la longitud del intervalo sea tan reducida que se considere que ya se ha alcanzado una precisión suficiente.

Así, dados a y b tales que $f(a)f(b) < 0$ se inicia un proceso iterativo:

`mientras` $|a - b| > \varepsilon_x$:

- `calcular` $c \leftarrow (a + b)/2$ y $f(c)$
- `si` $f(a)f(c) < 0$:

  `tomar` $b \leftarrow c$

  `en otro caso`:

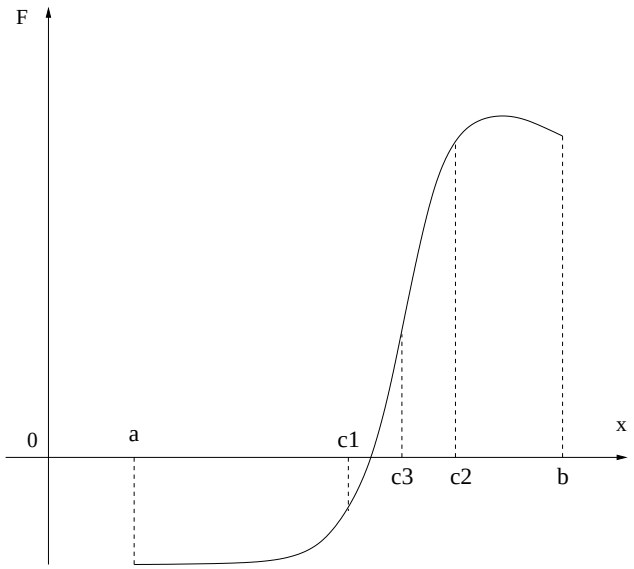

Figura 2.10: Interpretación gráfica del método de bisección

tomar $a \leftarrow c$

`terminar`

Normalmente se añade un segundo *test de parada* que compara $|f(c)|$ con ε_f (si el valor de $|f(c)|$ es muy pequeño querrá decir que c está próximo a la raíz, salvo que el problema esté mal condicionado), de modo que el algoritmo queda:

`para` $i = 1, 2, \ldots$

- `calcular` $c_i \leftarrow (a_i + b_i)/2$ `y` $f(c_i)$
- `si` $|f(c_i)| < \varepsilon_f$:

  `tomar` $x \leftarrow c_i$ `y terminar`
- `si` $f(a_i)f(c_i) < 0$:

  `tomar` $a_{i+1} \leftarrow a_i$, $b_{i+1} \leftarrow c_i$

  `en otro caso`:

  `tomar` $a_{i+1} \leftarrow c_i$, $b_{i+1} \leftarrow b_i$
- `si` $|b_{i+1} - a_{i+1}| < \varepsilon_x$:

  `tomar` $x \leftarrow \dfrac{a_{i+1} + b_{i+1}}{2}$ `y terminar`

`terminar en` n

Nota 2.2 *Además, comúnmente en la práctica:*

- *se añade un* test *inicial sobre signos de* $f(a)$ *y* $f(b)$
- *se prefiere comparar signo*$(f(a_i))$ *y signo*$(f(c_i))$ *(que sólo necesita comparar dos bits) y no evaluar signo*$(f(a_i)f(b_i))$ *(que requiere, además, un producto que no es necesario).*

Es fácil ver que si f es continua los intervalos $[a_i, b_i]$ seguirán conteniendo (al menos) una raíz de f, de modo que tomando $c_i = (a_i + b_i)/2$ como su aproximación se tendrá:

$$|x - c_n| \leq \frac{1}{2}|b_n - a_n| = \frac{1}{4}|b_{n-1} - a_{n-1}| = \cdots = \frac{1}{2^n}|b_1 - a_1|$$

que permite establecer una cota de error (absoluto) y calcular el número de iteraciones necesarias para alcanzar una cierta precisión ε_x, pues para asegurar que

$$\frac{1}{2^n}|b_1 - a_1| \leq \varepsilon_x$$

basta con tomar el (primer) valor de n tal que

$$2^n \geq \frac{|b_1 - a_1|}{\varepsilon_x}$$

esto es

$$n \geq \frac{\log(|b_1 - a_1|) - \log(\varepsilon_x)}{\log(2)}$$

A continuación se recoge un resultado formal de convergencia (cuya información ya estaba contenida, en todo caso, en los cálculos que se acaban de efectuar):

Teorema 2.3 *Sea* f *una función continua sobre el intervalo* $[a_0, b_0]$ *y tal que* $f(a_0)f(b_0) < 0$. *Sean* $\{a_n\}_{n=0}^{\infty}$, $\{b_n\}_{n=0}^{\infty}$ *y* $\{c_n\}_{n=0}^{\infty}$ *las sucesiones generadas por el método de bisección, de acuerdo con las notaciones anteriores.*

Entonces, denotando mediante x_* *una raíz de* f *en* $[a_0, b_0]$, *se tiene*

i) $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = \lim_{n\to\infty} c_n = x_*$

ii) $|x_* - c_n| \leq 2^{-(n+1)}(b_0 - a_0)$

Demostración: En primer lugar, $\{a_n\}$ y $\{b_n\}$ son sucesiones monótonas y acotadas, que deben tener el mismo límite pues

$$\lim_{n\to\infty} b_n - \lim_{n\to\infty} a_n = \lim_{n\to\infty} \frac{b_0 - a_0}{2^n} = 0$$

Por otro lado, denotando x_* al límite común y pasando al límite en

$$0 \geq f(a_n)f(b_n)$$

(f continua) se obtiene $(f(x_*))^2 \leq 0$ y, por tanto, $f(x_*) = 0$.

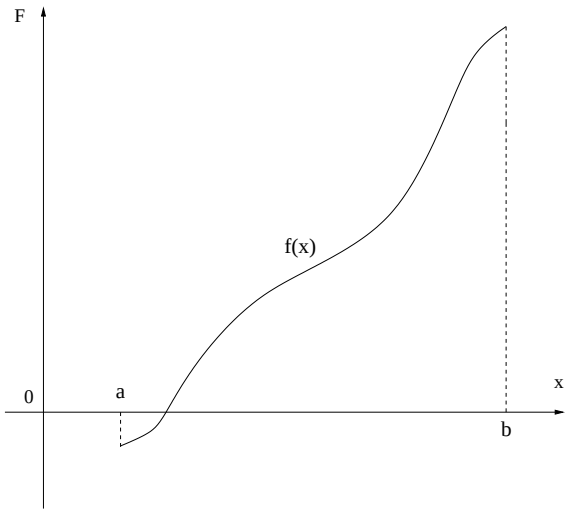

Figura 2.11: División *óptima* del subintervalo

□

Algunas observaciones

- Se trata de un método robusto (adecuado para una búsqueda global, y como tal empleado en muchos programas), pero con una convergencia relativamente lenta (por lo que, en la práctica, se combina con otros métodos). Una simple mejora sería dividir de otro modo el intervalo cuando es claro que la raíz está más cerca de un extremo (algo esperable cuando los valores absolutos de la función en los dos extremos son muy distintos, tal y como se ilustra en la figura 2.11). Esto es lo que hace, por ejemplo, el algoritmo de Brent implementado en la función `fzero` de MATLAB.

- Es importante recordar las hipótesis del teorema de Bolzano para emplear correctamente el método. Se debe haber separado una raíz (el método sólo puede calcular una) con $f(a)f(b) < 0$ (en caso contrario, no podrá arrancar o tendrá que iniciar una lenta búsqueda del intervalo, como hace la función 'fzero' de MATLAB). Además la función f ha de ser continua (en otro caso, el programa podría confundir una singularidad con una raíz en el criterio sobre ε_x).

- La combinación de los criterios sobre ε_x y ε_f puede permitir detectar los problemas mal condicionados (puesto que en ellos el criterio sobre ε_f se satisface fácilmente, pero no el criterio sobre ε_x).

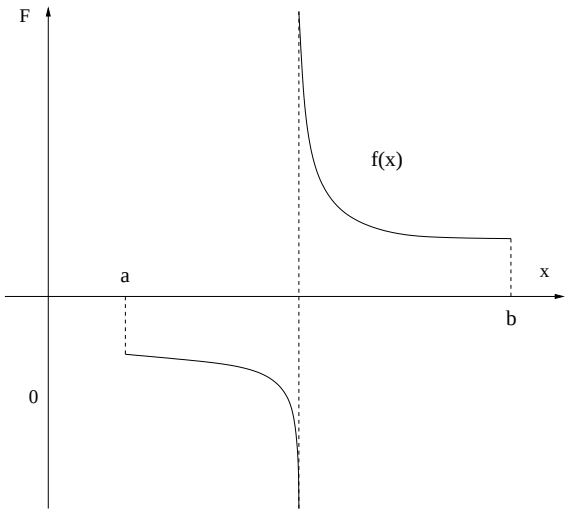

Figura 2.12: Inexistencia de raíz en el intervalo

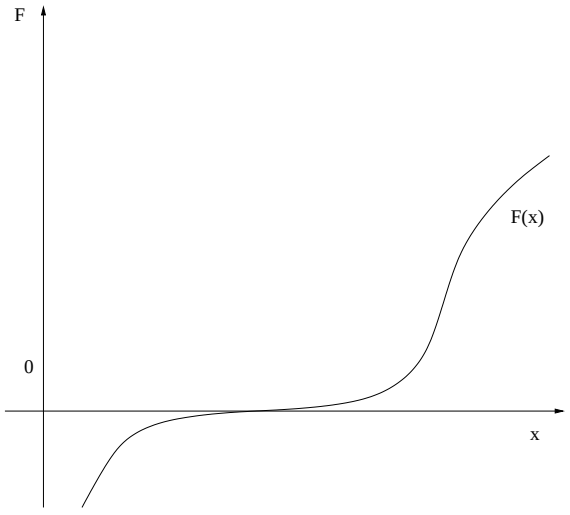

Figura 2.13: Problema mal condicionado

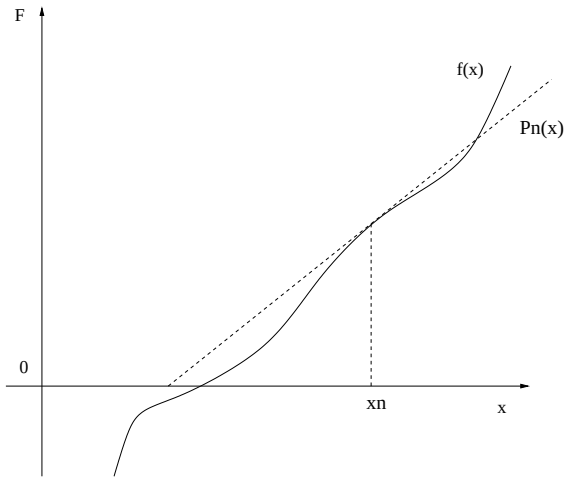

Figura 2.14: Interpretación del método de Newton-Raphson

2.4. Método de Newton-Raphson

Como se verá, el método de Newton-Raphson presenta unas características complementarias a las del método de bisección. La motivación del método es la siguiente (entre otras muchas): no disponemos de técnicas analíticas para resolver de forma exacta la ecuación

$$f(x) = 0$$

cuando f es una función cualquiera, pero sí cuando f es muy simple: un polinomio de grado uno o dos. Así, podríamos considerar el caso más sencillo (también existe un método que emplea el otro caso) y se haría necesario sustituir la función f por un polinomio de grado uno que aproxime a f. La elección parece fácil: el polinomio de Taylor de grado uno (en el entorno de una aproximación de la raíz x_*).

Sea entonces x_n una aproximación de x_*. Se calculará x_{n+1} (con la esperanza de generar una sucesión que converja a x_*) resolviendo

$$P_n(x_{n+1}) = 0$$

donde

$$P_n(x) = f(x_n) + f'(x_n)(x - x_n)$$

Así, es fácil ver que se está generando una sucesión:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

La figura 2.14 muestra una función $f(x)$, el polinomio de Taylor de grado uno para una cierta aproximación x_n de la raíz y la interpretación gráfica del iterante x_{n+1}.

Algunas observaciones

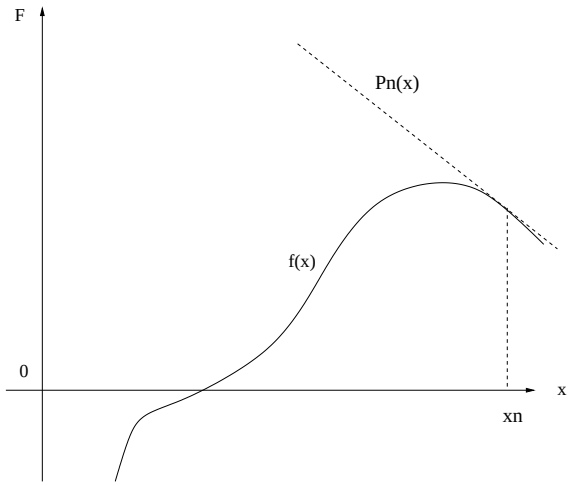

Figura 2.15: Dificultades de convergencia de Newton-Raphson

- En la programación del método se incluirán *test* de parada sobre ε_x (operarán con $|x_{n+1}-x_n|$) y sobre ε_f (operarán con $f(x_n)$). De nuevo, la relación entre ambos está ligada al condicionamiento del problema, aunque ahora se tiene acceso directo a él pues debe calcularse explícitamente $f'(x_n)$ (y $\lim_{n\to\infty} f'(x_n) = f'(x_*)$ si f es regular y $\lim_{n\to\infty} x_n = x_*$). Obsérvese que, en efecto, en los problemas mal condicionados los errores de redondeo se amplificarán notablemnente (al dividir por números pequeños). En el caso límite, con $f'(x_*) = 0$, también se deteriora el error de truncamiento (con una pérdida de un orden en la convergencia).

- Gráficamente, ya se puede ver que el método sólo funcionará bien si se arranca cerca de x_* pues en otro caso el resultado de una iteración del método puede llevarnos aún más lejos de la raíz (considérese, por ejemplo, una iteración desde x_n en la figura 2.15).

Convergencia (local) del método de Newton-Raphson

Si definimos el error (absoluto):

$$e_n = x_n - x_*$$

se tiene

$$e_{n+1} = x_{n+1} - x_* = x_n - \frac{f(x_n)}{f'(x_n)} - x_* = e_n - \frac{f(x_n)}{f'(x_n)} = \frac{e_n f'(x_n) - f(x_n)}{f'(x_n)}$$

Empleando ahora el desarrollo de Taylor:

$$0 = f(x_*) = f(x_n - e_n) = f(x_n) - e_n f'(x_n) + \frac{1}{2}\, e_n^2\, f''(\xi_n)$$

se obtiene

$$e_{n+1} = \frac{\frac{1}{2}\, e_n^2\, f''(\xi_n)}{f'(x_n)}$$

y esperamos que conforme se acerque x_n a x_* se tenga:

$$e_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)}\, e_n^2 \simeq \frac{f''(x_*)}{2f'(x_n)}\, e_n^2$$

Es posible entonces probar:

Teorema 2.4 *(convergencia local del método de Newton-Raphson)*
Supongamos que f es de clase C^2 en un entorno de x_* y que x_* es una raíz simple de f (esto es $f'(x_*) \neq 0$). Entonces existe un entorno $(x_*-\delta, x_*+\delta)$ tal que si se toma $x_0 \in (x_*-\delta, x_*+\delta)$ se tiene

i) $\lim_{n\to\infty} x_n = x_*$

ii) $\exists C > 0$ *tal que* $|x_{n+1} - x_*| \leq C\,|x_n - x_*|^2,\ \forall n \geq 0$

Demostración: El apartado *ii)* es inmediato (a partir de los cálculos anteriores) si x_{n+1} no abandona $(x_* - \delta, x_* + \delta)$ empleando una acotación de $\dfrac{f''(\xi_n)}{2f'(x_n)}$. Basta elegir δ de modo que $|e_{n+1}| < \delta$ si $|e_n| < \delta$ (esto es, si $x_n \in (x_* - \delta, x_* + \delta)$ entonces x_{n+1} también deberá estar en $(x_* - \delta, x_* + \delta)$); como

$$e_{n+1} = \frac{f''(\xi_n)}{2f'(x_n)}\, e_n^2$$

si definimos

$$c(\delta) = \frac{1}{2} \frac{\max\limits_{|x-x_*|\leq\delta} |f''(x)|}{\min\limits_{|x-x_*|\leq\delta} |f'(x)|}$$

es suficiente imponer $\delta\, c(\delta) < 1$ pues

$$|e_{n+1}| = \frac{|f''(\xi_n)|}{2|f'(x_n)|}\, e_n^2 \leq c(\delta)\, \delta^2 < \delta$$

Obsérvese que siempre puede elegirse δ con esta condición ya que

$$\lim_{z\to 0} z\, c(z) = 0$$

pues

$$\lim_{z\to 0} c(z) = \frac{1}{2} \frac{f''(x_*)}{f'(x_*)}$$

El apartado $i)$ es ahora fácil de obtener pues

$$|e_{n+1}| \leq c(\delta)\, |e_n|^2 \leq c(\delta)\, \delta\, |e_n|$$

y llamando $\rho = c(\delta)\, \delta$ se obtiene:

$$|e_{n+1}| \leq \rho\, |e_n| \leq \rho^2\, |e_{n-1}| \leq \cdots \leq \rho^{n+1}\, |e_0|$$

de forma que al pasar al límite (con $\rho < 1$):

$$\lim_{n\to\infty} |e_{n+1}| = 0$$

□

Cabe hacer ahora algunas observaciones:

- Aunque la demostración anterior es *constructiva* y devuelve una estimación del parámetro δ, en la práctica es de nula ayuda (pues obligaría a resolver problemas más complicados que la búsqueda de la raíz para encontrar el parámetro δ). Sólo en algunas situaciones muy concretas es posible obtener resultados precisos sobre el *dominio de atracción* del método hacia una raíz particular (véase resultado de convergencia *global* a continuación, que explota los signos de algunas derivadas). Además, los dominios de atracción pueden ser bastante complejos (véase comentario sobre extensión al cálculo de raíces complejas y ejemplo que ilustra la portada del texto de Kincaid-Cheney). Esta situación es todavía más complicada cuando se considera la extensión a la resolución de sistemas y conduce a la necesidad de ajustar el paso (mediante el denominado método de Newton amortiguado: *damped Newton*).

- El resultado de convergencia cuadrática es de gran interés desde el punto de vista computacional e impone en la práctica, *grosso modo*, que el número de decimales exactos de la aproximación se dobla con cada iteración del método. Así, si se arranca de una aproximación tal que $|e_0| \leq 10^{-2}$ se tendrá:

$$|e_1| \leq C\,|e_0|^2 \leq C\,10^{-4}$$

$$|e_2| \leq C\,|e_1|^2 \leq C^2\,10^{-8}$$

$$|e_3| \leq C\,|e_2|^2 \leq C^3\,10^{-16}$$

  Si para un cierto problema (fácil) se tiene $C \sim 1$ entonces arrancando con un error del orden de 10^{-2} en tres iteraciones ya se alcanza una precisión equivalente a la del almacenamiento en coma flotante con doble precisión.

Ejemplo 2.4 *(tomado del texto de Kincaid y Cheney)*
Para un número real positivo r se considera el cálculo de su raíz cuadrada como solución de la ecuación

$$x^2 - r = 0$$

mediante el método de Newton-Raphson (donde se tomará $f(x) = x^2 - r$):

$$x_{n+1} = x_n - \frac{x_n^2 - r}{2x_n} = \frac{1}{2}(x_n + \frac{r}{x_n})$$

Si se toma $r = 17$ y se arranca con $x_0 = 4$ ($e_0 \sim 10^{-1}$) se obtiene para los 4 primeros iterantes:

$$\begin{aligned} x_1 &= 4.12 \\ x_2 &= 4.123106 \\ x_3 &= 4.1231056256177 \\ x_4 &= 4.12310562561766054982140985 6 \end{aligned}$$

donde se han representado, en cada iterante, sólo los dígitos calculados exactamente (obsérvese que para retener toda la precisión obtenida en x_4 ya es necesario un almacenamiento más preciso que el aportado por la norma ANSI/IEEE de doble precisión; cabe mencionar aquí que algunos lenguajes de programación –como fortran 90 y 95– permiten manejar variables con precisión arbitrario de un modo fácilmente portable*).*

De hecho, la rápida convergencia de este esquema (que sólo emplea sumas y productos) hace que se utilice con frecuencia para implementar (en software de muy bajo nivel) la función raíz cuadrada en los procesadores. Aunque, como se verá posteriormente, para este caso las iteraciones del método convergen para $x_0 > 0$ cualquiera, el hecho de que la convergencia cuadrática no sea especialmente rápida lejos de la raíz hace que en la práctica se combine con técnicas de localización de la raíz para arrancar el cálculo con un error inicial relativamente reducido (obsérvese que estas técnicas pueden ser muy rápidas ya que pueden devolver una aproximación de la raíz a partir de la lectura de los primeros bits en la representación en coma flotante).

En relación con la convergencia local del método se tiene para este ejemplo:

$$C = \frac{1}{2} \max_{|x-x_*|\leq\delta} \frac{|f''(x)|}{|f'(x)|} \simeq \frac{1}{2}\frac{|f''(x_*)|}{|f'(x_*)|} = \frac{1}{2}\frac{2}{2x_*} = \frac{1}{2\sqrt{17}} \simeq 0.1213$$

lo que justifica la convergencia especialmente rápida (no obstante, obsérvese que en el cálculo de la raíz cuadrada esa situación no será rara ya que de hecho $C \to 0$ con $x_* \to \infty$), ganándose con cada paso un decimal adicional a los previstos.

Convergencia global del método de Newton-Raphson

Como se ha comentado anteriormente, en ciertas situaciones (aunque raras) es posible asegurar un cierto conjunto donde *arrancar* el método de Newton-Raphson con la seguridad de que la sucesión generada converja. A continuación se expone una de estas situaciones:

Teorema 2.5 *(convergencia (global) del método de Newton-Raphson)*
Sea $f : [a, b] \to \mathbf{R}$ de clase $C^2([a, b])$ y tal que :

- $f''(x) > 0, \ \forall x \in (a, b)$
- $f'(x) > 0, \ \forall x \in (a, b)$
- $f(a)f(b) < 0$

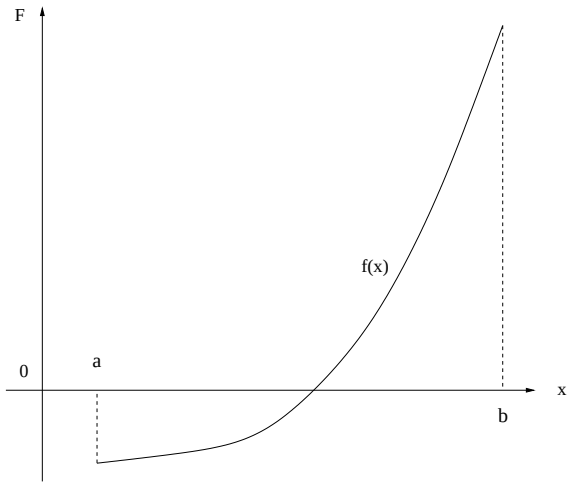

Figura 2.16: Convergencia global del método de Newton-Raphson

Entonces f tiene exactamente una raíz en (a,b) y el método de Newton-Raphson converge a ella para cualquier iterante inicial x_0 en (a,b), siempre que el primer iterante x_1 se encuentre en (a,b).

Demostración:

La idea de la demostración es fácil de comprender si se acude a la representación gráfica del método (véase la figura 2.16)

En primer lugar, si se considera el cálculo del primer iterante x_1 a partir de x_0 cabe distinguir dos situaciones:

i) $x_0 > x_*$; en cuyo caso, como se verá posteriormente, quedará garantizado que $x_1 \in (x_*, x_0)$ y por lo tanto, el primer iterante siempre está definido en (a,b).

ii) $x_0 < x_*$; ahora veremos que es suficiente imponer

$$\frac{|f(a)|}{f'(a)} < b - a$$

para asegurar

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \leq a - \frac{f(a)}{f'(a)} < a + b - a = b$$

Obsérvese, en el apartado ii), que para verificar la primera desigualdad, basta con acudir a las propiedades de $F(x) = x - \dfrac{f(x)}{f'(x)}$ (véase la figura 2.17) pues

$$F'(x) = 1 - \frac{f'(x)f'(x) - f''(x)f(x)}{(f'(x))^2} = \frac{f''(x)f(x)}{(f'(x))^2}$$

con $F'(x) < 0$ si $x < x_*$ y $F'(x) > 0$ si $x > x_*$. Así, junto con $F(x) > x$ si $x < x_*$ y $F(x) < x$ si $x > x_*$, se concluye la acotación buscada.

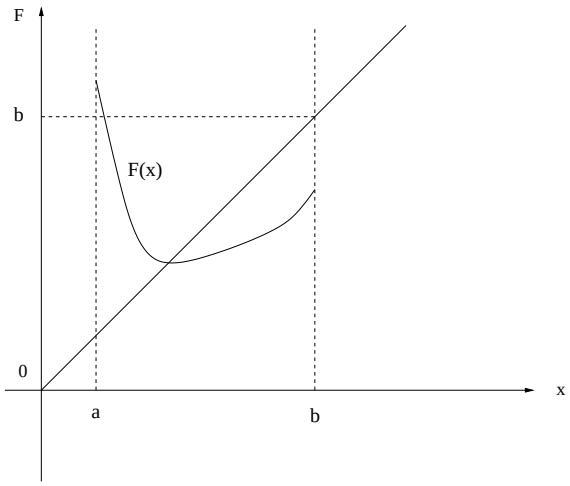

Figura 2.17: Convergencia global del método de Newton-Raphson

De este modo, para que el primer iterante esté bien definido basta con asegurar que lo está si arrancamos con $x_0 = a$ (o, en la prácica, si elegimos a de modo que lo esté).

Una vez verificada esta hipótesis obsérvese que

$$e_1 = e_0^2 \frac{1}{2} \frac{f''(\xi_0)}{f'(x_0)} > 0$$

de modo que, independientemente de donde se arranque, tras una iteración se obtiene:

$$x_* < x_1 < b$$

(tal y como se comprueba gráficamente de modo inmediato).

Así, teniendo en cuenta que todos los siguientes términos de la sucesión están en (x_*, b) puesto que

$$e_{n+1} = e_n^2 \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} > 0$$

y que, por otro lado (f es creciente y $f(x_*) = 0$):

$$e_{n+1} = e_n - \frac{f(x_n)}{f'(x_n)} < e_n$$

la sucesión es acotada y monótona decreciente. Por lo tanto, existe límite, tanto para la sucesión $\{x_n\}$ como para la sucesión $\{e_n\}$ verificando sus límites

$$e_\infty = e_\infty - \frac{f(x_\infty)}{f'(x_\infty)}$$

y en consecuencia

$$\lim_{n\to\infty} x_n = x_*$$

□

Ejemplo 2.5 *Retomando el ejemplo anterior (cálculo de raíces cuadradas) es fácil ver que para $f(x) = x^2 - r$ basta tomar $[a, b] = [\varepsilon, K]$ con K suficientemente grande y ε lo suficientemente pequeño (en la práctica podemos adaptar el teorema anterior para considerar el caso $[a, +\infty)$; en Kincaid-Cheney, el resultado se demuestra de hecho para $f : \mathbf{R} \to \mathbf{R}$, lo que simplifica las hipótesis) para comprobar que se tienen las hipótesis del teorema anterior. Recuérdese que, en todo caso, no se suele emplear el método de Newton para arrancar los cálculos (pues no es rápido lejos de la raíz).*

Observaciones sobre el cálculo de la derivada

- Puesto que el método necesita del cálculo de la derivada no resulta aplicable si ésta no se encuentra disponible (por ejemplo cuando el valor se obtiene de algún código cuya fuente no es accesible o incluso de medidas sobre un sistema físico) o su cálculo es demasiado costoso o complejo (por ejemplo, existen herramientas que permiten el cálculo automático de las derivadas en códigos de los cuales tenemos acceso a las fuentes –como ADIFOR para códigos en Fortran– pero es preciso generar el nuevo código y compilarlo). En estos casos aún resulta posible emplear derivación numérica para estimar las derivadas, pero como se verá posteriormente, resultará preferible acudir a otros métodos (como los métodos de tipo secante).

- La evaluación de la derivada resulta, en general, bastante más costosa que la evaluación de la función (piénsese en la derivación de una expresión no demasiado simple). Así, en la práctica, a veces no se actualiza el valor de la derivada con cada etapa sino que se mantiene fijo durante un cierto número de etapas. Se busca así una aceleración de los cálculos a costa de un deterioro de las propiedades de convergencia (de modo que se deberá garantizar que para el problema en cuestión dicha modificación mejora las propiedades globales). Esta técnica será muy habitual en la extensión del método de Newton para sistemas de ecuaciones.

Observación sobre cálculo de raíces complejas

El método de Newton-Raphson puede extenderse al cálculo de raíces complejas (de funciones consideradas ahora como funciones de variable compleja), para lo cual basta operar en aritmética compleja (tomando la derivada como derivada en el campo complejo). La propiedad de convergencia local se extiende además a la versión *compleja* del método (no así la propiedad de convergencia global que emplea resultados del análisis real).

Por otro lado, la versión compleja del método de Newton-Raphson permite mostrar de forma gráfica (a diferencia del caso real donde es menos ilustrativa) las dificultades en el comportamiento global del método.

Así, si se considera el cálculo de las raíces del polinomio $z^3 - 1$, las iteraciones convergen

a cada raíz:

$$z_1 = 1, \;\; z_2 = \frac{-1+\sqrt{3}\,i}{2}, \;\; z_3 = \frac{-1-\sqrt{3}\,i}{2}$$

siempre que se arranque cerca de ella, pero converge a otra raíz o no converge si lo hace no tan cerca (el conjunto de puntos para los cuales el esquema no converge se denomina el conjunto de Julia y constituye la frontera de cada *cuenca de atracción*). La portada del texto de Kincaid y Cheney ilustra con 3 colores las cuencas de atracción (conjuntos abiertos) de cada una de las 3 raíces e ilustra perfectamente las dificultades de convergencia global del método de Newton.

2.5. Método de la secante y variantes

Los métodos que se van a presentar a continuación (y otros muchos no mencionados aquí) buscan unas propiedades de convergencia lo más parecidas posibles al método de Newton-Raphson, pero sin necesidad de evaluar la derivada (porque no está disponible o es costosa de evaluar) al tratar con aproximaciones numéricas de ésta.

2.5.1. Método de la secante

Suponiendo que ya han sido calculados al menos dos términos de una sucesión (que se espera converja a x_*) x_{n-1} y x_n, el método de la secante propone sustituir el iterante del método de Newton

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

por este otro (que emplea una aproximación de la derivada)

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

tal y como se ilustra en la figura 2.18

- Obsérvese que, conforme x_n se acerque a x_* (si es que esto ocurre) la aproximación de la derivada es cada vez mejor pero, al mismo tiempo, se debe tener cuidado con la pérdida de precisión debido a la división por un número pequeño.

- Por otro lado, el método de la secante presenta en principio *parecidas* dificultades de convergencia global a las del método de Newton-Raphson. Posteriormente se verá (método de regula falsi) cómo resolver estas dificultades a costa, eso sí, de una nueva pérdida de velocidad de convergencia.

Nota 2.3 *A pesar del comentario anterior, en la práctica el método de la secante presenta menos dificultades de convergencia global que el método de Newton-Raphson, especialmente*

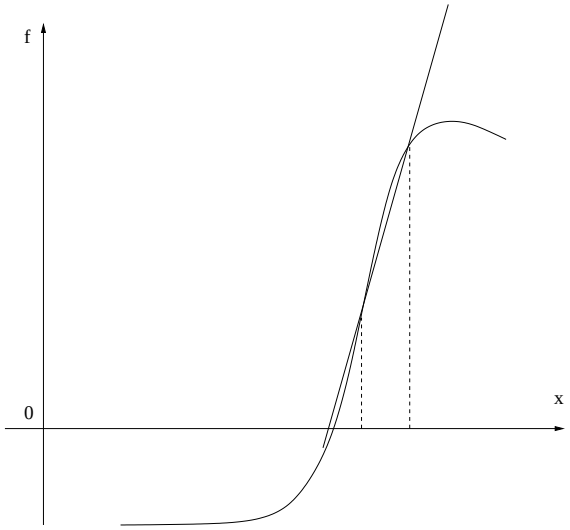

Figura 2.18: Esquema del método de la secante

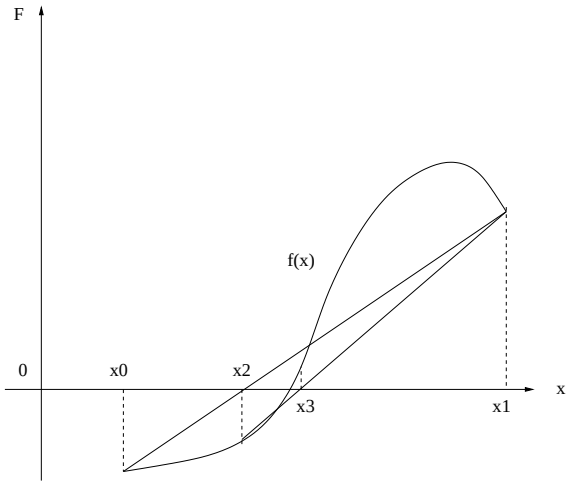

Figura 2.19: Ejemplo de convergencia global del método de la secante

si se arranca (como es habitual) desde los extremos de un intervalo que encierra una raíz separada. Así, en el ejemplo de la figura 2.19, el método de Newton-Raphson arrancando desde x_0 o x_1 no converge a la raíz, pero sí el método de la secante.

A continuación se presenta un teorema que asegura la convergencia local del método y estima el deterioro de la convergencia causado por el empleo de una aproximación de la derivada (por comparación con el método de Newton).

Teorema 2.6 *(convergencia local del método de la secante)*
Supongamos que f es de clase C^2 en un entorno de x_* y que x_* es una raíz simple de f. Entonces, existe un entorno $(x_* - \delta, x_* + \delta)$ tal que si se toman $x_0, x_1 \in (x_* - \delta, x_* + \delta)$ se tiene:

i) $\lim_{n\to\infty} x_n = x_*$

ii) $\exists C' > 0$ *tal que* $|x_{n+1} - x_*| \leq C' |x_n - x_*|^\alpha,\ \forall n \geq 0$ *con* $\alpha = (1+\sqrt{5})/2 \simeq 1.62$

Demostración: La demostración es semejante a la correspondiente al método de Newton-Raphson de modo que sólo se incluye la obtención del orden. Así, se toma

$$e_{n+1} = x_{n+1} - x_* = \frac{f(x_n)\, x_{n-1} - f(x_{n-1})\, x_n}{f(x_n) - f(x_{n-1})} - x_* = \frac{f(x_n)\, e_{n-1} - f(x_{n-1})\, e_n}{f(x_n) - f(x_{n-1})}$$

Sacando factor común $e_{n-1}e_n$ y multiplicando y dividiendo por $x_n - x_{n-1}$ se obtiene

$$e_{n+1} = \underbrace{\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}}_{a_n} \underbrace{\frac{\frac{1}{e_n} f(x_n) - \frac{1}{e_{n-1}} f(x_{n-1})}{x_n - x_{n-1}}}_{b_n} e_n\, e_{n-1}$$

donde es fácil ver que a_n puede escribirse como$\dfrac{1}{f'(\xi_n)}$, en tanto que para b_n empleando:

$$\begin{aligned} \frac{1}{e_n} f(x_n) &= \frac{1}{e_n} f(x_* + e_n) = \frac{1}{e_n}\{0 + f'(x_*)e_n + \frac{1}{2} f''(\eta_n^1) e_n^2\} \\ \frac{1}{e_{n-1}} f(x_{n-1}) &= \frac{1}{e_{n-1}} f(x_* + e_{n-1}) = \frac{1}{e_{n-1}}\{0 + f'(x_*)e_{n-1} + \frac{1}{2} f''(\eta_n^2) e_{n-1}^2\} \end{aligned}$$

junto con $x_n - x_{n-1} = e_n - e_{n-1}$ se obtiene (pueden consultarse los detalles, por ejemplo, en el texto de Kincaid y Cheney):

$$b_n = \frac{1}{2} \frac{f''(\eta_n^1) e_n - f''(\eta_n^2) e_{n-1}}{e_n - e_{n-1}} = \frac{1}{2} f''(\xi_n)$$

para algún ξ_n intermedio (entre η_n^1 y η_n^2), de modo que si x_{n_1} y x_n están próximos a x_*

$$b_n \simeq \frac{1}{2} f''(x_*)$$

Se tiene, en suma:

$$e_{n+1} = \frac{1}{2}\frac{f''(x_n)}{f'(\xi_n)}\, e_n\, e_{n-1}$$

Por lo tanto, en el límite, $e_{n+1} \simeq C\, e_n\, e_{n-1}$ con $C = \frac{1}{2}\frac{f''(x_*)}{f'(x_*)}$.

Supongamos ahora (a fin de comparar la velocidad de convergencia de este método con el método de Newton) que

$$e_{n+1} \simeq C'\, e_n^{\alpha}$$

(que corresponderá, de acuerdo con la definición que se verá posteriormente, a una convergencia con orden α), de modo que a su vez $e_n \simeq C'\, e_{n-1}^{\alpha}$.

Se tiene entonces por un lado que $e_{n+1} \simeq C\, e_n\, e_{n-1}$ al tiempo que (de acuerdo con la hipótesis hecha)

$$e_{n-1} \simeq (\frac{1}{C'}\, e_n)^{1/\alpha}$$

y así se concluye que, si esta hipótesis fuese cierta, entonces

$$e_{n+1} \simeq \frac{C}{(C')^{(1/\alpha)}}\, e_n^{(1+1/\alpha)}$$

Comparando ahora las dos expresiones de e_{n+1} se tiene

$$C'\, e_n^{\alpha} \simeq \frac{C}{(C')^{(1/\alpha)}}\, e_n^{(1+1/\alpha)}$$

y se comprueba que la hipótesis hecha será cierta si y sólo si:

$$(C')^{(1+1/\alpha)} = C \qquad \text{y} \qquad \alpha = 1 + \frac{1}{\alpha}$$

devolviendo esta segunda ecuación: $\alpha = \dfrac{1+\sqrt{5}}{2}$.

En suma, se ha obtenido entonces:

$$e_{n+1} \simeq \left(\frac{1}{2}\frac{f''(x_*)}{f'(x_*)}\right)^{\alpha-1}\, e_n^{\alpha}$$

donde se ha empleado: $\dfrac{1}{1+\frac{1}{\alpha}} = \dfrac{1}{\alpha} = \alpha - 1 \simeq 0.62$

□

Nota 2.4 *Recuérdese que la constante C' también tiene su relevancia en la velocidad de convergencia. Si comparamos los resultados para el método de la secante y el método de Newton se ve que:*

a) $C < C'$ *en los problemas* fáciles *(donde* $C' < 1$*)*

b) $C > C'$ *en los problemas* difíciles *(donde* $C > 1$*)*

lo que confirma la mayor robustez *del método de la secante.*

Se introduce a continuación la definición de orden de convergencia que resultará de gran utilidad a la hora de comparar dos métodos.

Definición 2.1 *(orden de convergencia)*
Sea $\{x_n\}$ *una sucesión generada por un cierto método numérico, que converge a un valor* x_**. Se denomina orden de convergencia al mayor número real* q *tal que el límite*

$$\lim_{n\to\infty} \frac{|x_{n+1} - x_*|}{|x_n - x_*|^q}$$

existe y es diferente de cero.

Obsérvese que puede ocurrir que tal número no exista. Así, por ejemplo, no existe en general para las sucesiones de puntos medios generados por el método de bisección. Sin embargo, sí existe para las sucesiones generadas por un gran número de métodos (a condición de que el problema tratado sea suficientemente regular).

El orden de convergencia es una propiedad muy importante de los métodos. Así, junto con la *robustez* (y, en menor medida, el coste de cálculo) constituye la propiedad más importante. Permite, por ejemplo, predecir el número de decimales exactos ganados con cada iteración del método una vez que se está cerca de la raíz.

Recordando ahora los resultados de convergencia (local) del método de Newton-Raphson y el método de la secante, se tiene:

$$\begin{aligned} \text{Método de Newton-Raphson} &: |x_{n+1} - x_*| \simeq C\,|x_n - x_*|^2 \\ \text{Método de la secante} &: |x_{n+1} - x_*| \simeq C'\,|x_n - x_*|^\alpha \end{aligned}$$

con $\alpha \simeq 1.62$ y $C' = C^{\alpha-1}$. Así, de acuerdo con la definición de orden, el método de Newton-Raphson es un método de orden dos pero el método de la secante es sólo de orden α. De este modo, aunque el método de la secante es algo más robusto que el método de Newton-Raphson, una vez cerca de la raíz este último método convergerá más rápidamente.

Nota 2.5 *Como se ha visto, la sustitución de la derivada (en el método de Newton-Raphson) por una aproximación de ésta lleva a una reducción del orden de convergencia. Sin embargo, también debe tenerse en cuenta que el coste de cálculo involucrado es distinto. Con cada nuevo paso el método de Newton-Raphson necesita calcular* $f(x_n)$ *y* $f'(x_n)$ *en tanto que el método de la secante sólo requiere calcular* $f(x_n)$*. En aquellas situaciones donde el cálculo de* f *y* f' *sea tan complejo que prácticamente* consuma *todos los cálculos, sería razonable*

comparar un paso del método de Newton-Raphson con dos pasos del método de la secante. Así

$$\begin{aligned} \textit{Método de Newton-Raphson} &: |x_{n+1} - x_*| \simeq C\,|x_n - x_*|^2 \\ \textit{Método de la secante} &: |x_{n+1} - x_*| \simeq C'\,(C'\,|x_{n-1} - x_*|^\alpha)^\alpha \end{aligned}$$

de modo que

$$|x_{n+1} - x_*| \simeq (C')^{1+\alpha}\,|x_{n-1} - x_*|^{\alpha^2}$$

con $\alpha^2 \simeq 2.62$. *Resultaría entonces preferible por razones de convergencia (local) el método de la secante.*

2.5.2. Método de regula falsi

Como se ha mencionado, el método de la secante presenta básicamente las mismas dificultades de convergencia global que el método de Newton-Raphson (véase, no obstante, la nota 2.6 más adelante). Sin embargo, el hecho de que base la aproximación de la derivada en una pareja de puntos permite su modificación para asegurar que la pareja de puntos *encaje* la raíz. Ésta es la idea del método de regula falsi, que genera una sucesión $\{x_n\}$ a partir de dos valores iniciales a y b tales que $f(a)f(b) < 0$ de la forma siguiente:

- `tomar` $x_0 \leftarrow a,\ \ x_1 \leftarrow b$
- `para` $n = 1, 2, \ldots$

  `tomar` $\quad x_{n+1} \leftarrow x_n - f(x_n)\dfrac{x_n - x_m}{f(x_n) - f(x_m)}$

  `donde` $\quad m = m(n)$ `mayor índice (menor que` n`) tal que` $f(x_n)f(x_m) < 0$

`terminar en` $\ n$

Desde luego, a esta versión básica del algoritmo se le añadirán los correspondientes *test de parada*.

Es posible interpretar este método como un método de bisección donde, en vez de dividir el intervalo por la mitad se intenta *acelerar* los cálculos dividiendo por el punto de intersección (con el eje de abscisas) de la recta que une los valores en los extremos (véase nota 2.7). Así, el intervalo definido por x_n y x_m encaja siempre a la raíz (evitando los problemas de convergencia del método de la secante).

Nota 2.6 *Como también se ha comentado previamente, no es estrictamente cierto que el método de la secante presente las mismas dificultades de convergencia global que el método de Newton. Si se arranca el método* encajando *una raíz, el método resulta bastante más robusto*

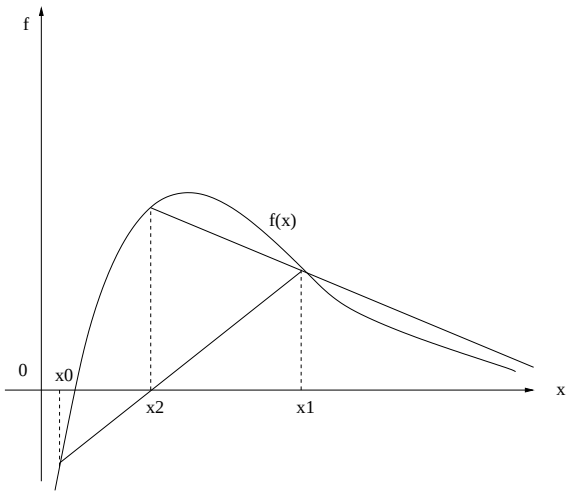

Figura 2.20: Dificultades del método de la secante

que el método de Newton. De hecho, en esta situación, la primera iteración (y quizás alguna más) coincide con el método de regula falsi. En cualquier caso, sí es fácil engañar *al método de la secante (como al de Newton), por ejemplo con* $f(x) = \dfrac{ln(x)}{x}$ *como se oberva en la figura 2.20.*

Nota 2.7 *Obsérvese que cabe esperar que el método de regula falsi proponga una mejor división del intervalo* $[a, b]$ *para encajar la raíz que el método de bisección cuando los valores* $|f(a)|$ *y* $|f(b)|$ *son muy distintos. Sin embargo, debe al mismo tiempo observarse que la división propuesta por el método de regula falsi sólo será adecuada si* f'' *es reducida; en otro caso sería preferible emplear una interpolación de mayor orden, como hace el método de Muler que utiliza una interpolación cuadrática. Así, es de esperar que si se arranca cerca de la raíz (con las contribuciones de* f'' *poco importantes) el método de regula falsi mejore el comportamiento del método de bisección (obsérvese la figura 2.11). Sin embargo, lejos de la raíz o con* f *complicada no tiene por qué mejorar.*

A continuación se enuncia un resultado de convergencia del método de regula falsi.

Teorema 2.7 *(convergencia del método de regula falsi)*
Sea $f : [a, b] \to \mathbf{R}$ *de clase* $C([a, b])$*. Supongamos que* $f(a)f(b) < 0$ *y que* $[a, b]$ *contiene una única raíz* x_**. Sea* $\{x_n\}$ *la sucesión generada por el método de regula falsi (prolongada por* x_l *si* $f(x_l) = 0$*).*

Entonces se tiene:

i) $\lim_{n\to\infty} x_n = x_*$

ii) $\exists C > 0$ *tal que* $|x_{n+1} - x_*| \leq C\,|x_n - x_*|\,, \quad \forall n \geq 1$

iii) El método no tiene orden mayor que uno.

Demostración: Véase el texto de Ralston y Rabinowitz *A First Course in Numerical Analysis.*

□

Obsérvese que, al igual que en el caso del método de bisección, se trata de un resultado tanto local como global de convergencia.

Como se ve, el coste de la *robustez* ganada con el método de regula falsi implica volver a una velocidad de convergencia similar a la del método de bisección (en cuanto a su orden, pues como se ha comentado, se espera que el método de regula falsi necesite menos iteraciones que el método de bisección al menos cerca de la raíz), a pesar de que su complejidad (y por lo tanto, su coste de cálculo) es similar a la del método de la secante.

A continuación se ilustra, mediante una pareja de ejemplos, el comportamiento típico de los métodos presentados.

Ejemplo 2.6 *Considérese el siguiente ejemplo (tomado del texto* Numerical Mathematics *citado al final del tema): Encontrar la única raíz positiva de (véase la figura 2.21)*

$$cos^2(2x) - x^2 = 0$$

empleando los métodos:

- *bisección*
- *regula falsi*
- *secante*
- *Newton-Raphson*

para los siguiente casos (donde se especifica el intervalo inicial para los métodos de bisección y regula falsi, que sirve también como pareja inicial de iterantes para el método de la secante, o el iterante inicial para el método de Newton):

a) $a_0 = 0$ *y* $b_0 = 3/2$ *ó* $x_0 = 3/4$

b) $a_0 = 0$ *y* $b_0 = 10$ *ó* $x_0 = 5$

y $\varepsilon_x = 10^{-12}$ *como* test de parada.

Obsérvese que es fácil obtener la existencia y unicidad de solución en $(0, \pi/4)$ *pues*

$$\begin{aligned} f(0) &= 1 > 0 \\ f(\pi/4) &= -(\pi/4)^2 < 0 \end{aligned}$$

y además

$$f'(x) = -2(sen(4x) + x) < 0$$

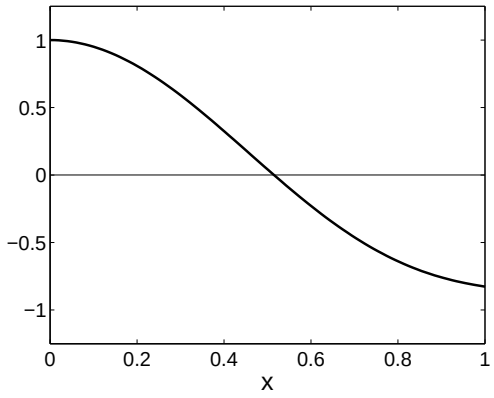

Figura 2.21: Función $f(x) = \cos^2(2x) - x^2$

en ese intervalo ya que $sen(4x) > 0$ *para* $x \in (0, \pi/4)$.

Por otro lado, es fácil ver que $f(1) < 0$ *y además* $f'(x) < 0$ *para* $x > 1$ *de modo que no pueden existir raíces con* $x > 1$ *ya que al ser* f *de clase* C^1*:*

$$f(z) = f(1) + f'(\eta)(z-1) < 0 \quad para\, z > 1.$$

Por último, en $(\pi/4, 1)$ f' *sólo tiene un extremo que es además un mínimo de modo que también* $f'(x) < 0$ *para* $x \in (\pi/4, 1)$ *y por lo tanto, este intervalo tampoco puede contener una raíz.*

En relación ahora con el comportamiento de los métodos, el caso a) (que ilustra el comportamiento local *de los métodos para un problema* fácil*, donde se arranca cerca de la raíz y la función no tiene derivadas de orden superior* grandes*) devuelve un comportamiento típico (véase la figura 2.22) donde:*

- *el método de Newton converge con la velocidad más alta, seguido por el de la secante y después por el método de regula falsi y el de bisección*

- *aunque para los métodos de regula falsi y bisección la convergencia es* lineal *(en el sentido en que el método de regula falsi es de orden uno y el método de bisección divide a cada paso la longitud del intervalo –y, por lo tanto, una cota de error– por un factor un medio con cada paso), el primero tiene una convergencia algo más rápida (como corresponde a una* mejor *división del intervalo que encaja la raíz).*

Para el caso b), que arranca más lejos de la raíz, es posible comprobar (véase la figura 2.23) que:

- *el método de regula falsi no devuelve ahora una mejora de la convergencia (si lo comparamos con el método de bisección)*

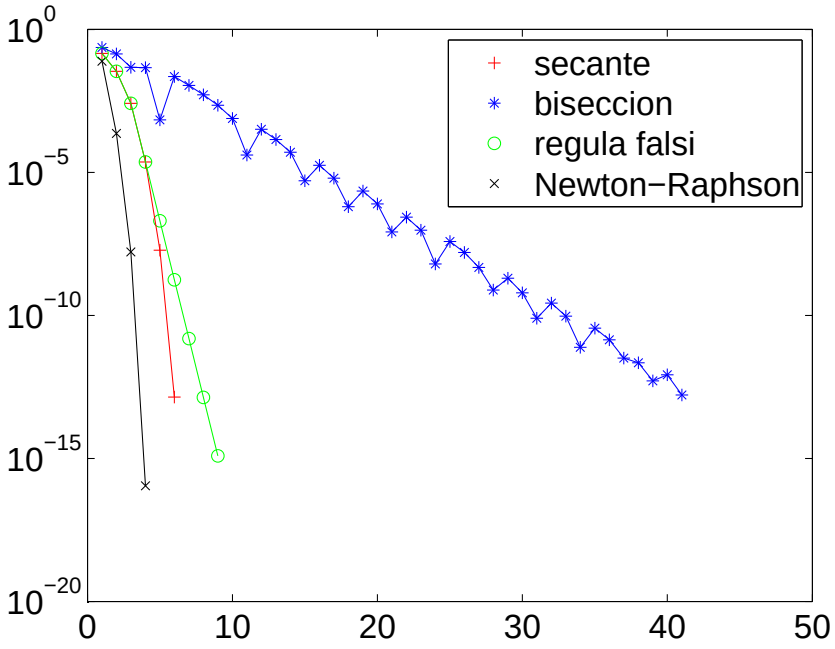

Figura 2.22: Comparación de la convergencia de los distintos esquemas en el ejemplo 2.6 para el caso a)

- *tanto el método de la secante como el método de Newton convergen a pesar de no arrancar cerca (lo que tiene que ver con que la función f se comporta lejos del origen como $-x^2$ con las propiedades de convergencia global que eso implica) y el primero se muestra como algo más robusto que el segundo.*

Ejemplo 2.7 *Se considera ahora este problema más difícil que el anterior: Encontrar la única raíz positiva de la ecuación (véase la figura 2.24)*

$$\frac{1}{2} + cos^2(6x) + \frac{3}{2}x^2$$

empleando los mismos métodos para $a_0 = 0$ y $b_0 = 5$ ó $x_0 = 2.5$ y con $\varepsilon_x = 10^{-12}$ como test de parada.

Este ejemplo permite comprobar que:

- *para un problema que no sea muy elemental (este problema es, de nuevo,* globalmente fácil *pues lejos del origen $f(x) \simeq \frac{3}{2}x^2$) no es posible garantizar la convergencia del método de Newton si no se arranca (muy) cerca (de hecho, tampoco hay convergencia con $x_0 = 0.5$)*
- *el método de la secante es robusto (si el problema es* globalmente fácil*)*

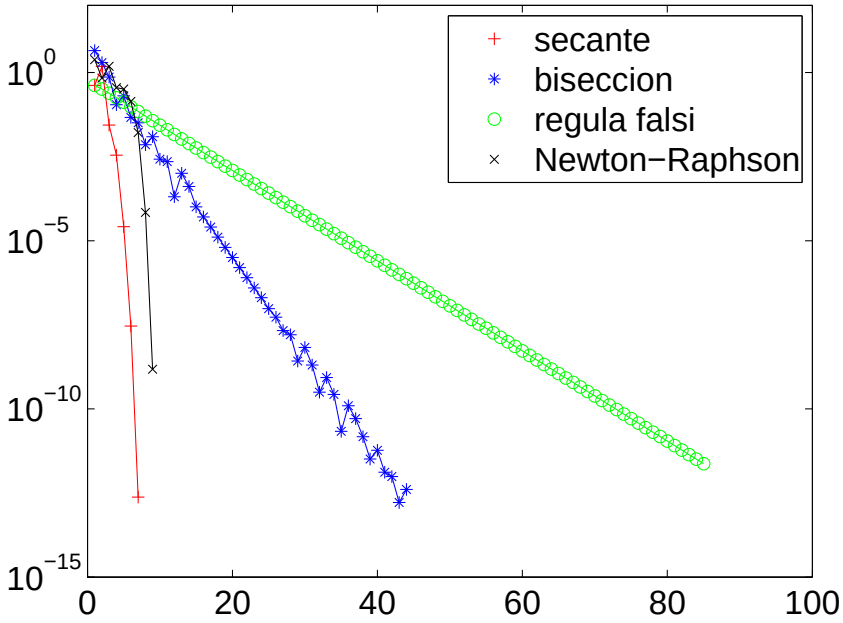

Figura 2.23: Comparación de la convergencia de los distintos esquemas en el ejemplo 2.6 para el caso b)

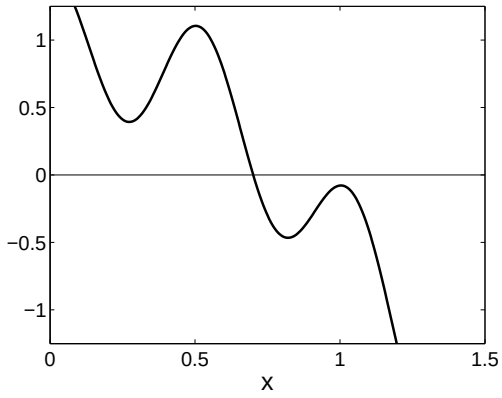

Figura 2.24: Función $f(x) = \frac{1}{2} + \cos^6(6x) + \frac{3}{2}x^2$

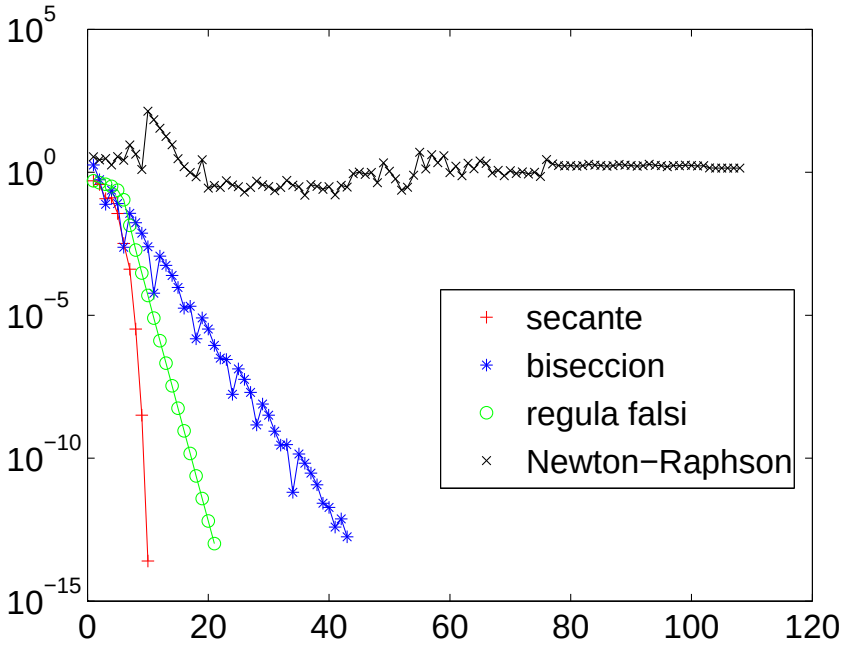

Figura 2.25: Comparación de la convergencia de los distintos esquemas en el ejemplo 2.7

2.6. Aceleración de la convergencia

En general, la búsqueda de la robustez de los métodos origina una lenta convergencia de éstos. Así, por ejemplo, el método de regula falsi permite ganar robustez en relación con el método de la secante, pero tiene a cambio sólo un orden uno.

Existen técnicas generales, sin embargo, que pueden mejorar el orden de los métodos generando una nueva sucesión con una convergencia más rápida. A este tipo de técnicas se les denomina métodos de aceleración de la convergencia.

A continuación se presenta una de ellas, conocida como algoritmo Δ^2 de Aitken que permite recuperar una convergencia superlineal a partir de la sucesión generada por un método cualquiera con convergencia lineal.

Así, sea un método con convergencia lineal y una sucesión (convergente) generada por dicho método de modo que, cerca ya de la raíz, se tenga

$$e_{n+1} \simeq C\, e_n$$

y para el paso siguiente

$$e_{n+2} \simeq C\, e_{n+1}$$

De las dos ecuaciones anteriores (tomando la igualdad, despreciando así el término adicional)

se puede eliminar C al tomar el cociente:

$$\frac{x_{n+2} - x_*}{x_{n+1} - x_*} = \frac{x_{n+1} - x_*}{x_n - x_*}$$

y de la ecuación resultante obtener x_*:

$$(x_{n+2} - x_*)(x_n - x_*) = (x_{n+1} - x_*)^2$$

$$x_{n+2}\, x_n - x_{n+2}\, x_* - x_n\, x_* + x_*^2 = x_{n+1}^2 - 2x_{n+1}\, x_* + x_*^2$$

$$x_* = \frac{x_n\, x_{n+2} - x_{n+1}^2}{x_n + x_{n+2} - 2x_{n+1}}$$

Nota 2.8 *La sucesión generada mediante el algoritmo puede también reescribirse de modo sencillo si se emplea el operador*

$$\Delta x_k = x_{k+1} - x_k$$

pues

$$x_{n+2} - 2x_{n+1} + x_n = (x_{n+2} - x_{n+1}) - (x_{n+1} - x_n) = \Delta x_{n+1} - \Delta x_n$$

y se escribe como $\Delta^2 x_n$.

Así, separando x_n *de la expresión anterior*

$$S_n = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$$

La idea del algoritmo Δ^2 de Aitken es entonces generar una segunda sucesión cuyo término general es

$$S_n = \frac{x_n\, x_{n+2} - x_{n+1}^2}{x_n + x_{n+2} - 2x_{n+1}}$$

que se espera que devuelva mejores aproximaciones (con una convergencia más rápida) que las suministradas por $\{x_n\}$. El siguiente resultado muestra que así ocurre:

Proposición 2.2 *Sea* $\{x_n\}$ *una sucesión convergente a la raíz* x_**, generada por un método de orden uno. Sea* $\{S_n\}$ *la sucesión generada por el algoritmo* Δ^2 *de Aitken. Entonces se tiene:*

$$\lim_{n\to\infty} \frac{|S_n - x_*|}{|x_n - x_*|} = 0$$

(que se denomina convergencia superlineal).

Demostración:

$$S_n - x_* = \frac{(x_* + e_n)(x_* + e_{n+2}) - (x_* + e_{n+1})^2}{(x_* + e_n) + (x_* + e_{n+2}) - 2(x_* + e_{n+1})} - x_* = \frac{e_n\, e_{n+2} - e_{n+1}^2}{e_n + e_{n+2} - 2e_{n+1}}$$

Empleando ahora

$$\begin{aligned} e_{n+1} &= (C+\delta_{n+1})e_n \\ e_{n+2} &= (C+\delta_{n+2})e_{n+1} \end{aligned}$$

con $\delta_n \to 0$ para $n \to +\infty$ (y $|C| < 1$ para que el método converja) se obtiene:

$$S_n - x_* = \frac{e_n(C+\delta_{n+2})(C+\delta_{n+1})e_n - (C+\delta_{n+1})^2 e_n}{e_n + (C+\delta_{n+2})(C+\delta_{n+1})e_n - 2(C+\delta_{n+1})e_n}$$

y así,

$$\frac{S_n - x_*}{x_n - x_*} = \frac{S_n - x_*}{e_n} = \frac{(C+\delta_{n+2})(C+\delta_{n+1}) - (C+\delta_{n+1})^2}{1 + (C+\delta_{n+2})(C+\delta_{n+1}) - 2(C+\delta_{n+1})} = \frac{N_n}{D_n}$$

de forma que

$$\begin{aligned} N_n &\to C^2 - C = 0 \text{ con } n \to \infty \\ D_n &\to 1 + C^2 - 2C = (1-C)^2 > 0 \text{ con } n \to \infty \end{aligned}$$

y en consecuencia

$$\lim_{n\to\infty} \frac{|S_n - x_*|}{|x_n - x_*|} = 0$$

□

Esta idea del algoritmo de Aitken es extendible a todos aquellos métodos para los cuales se dispone de una expresión asintótica del error ya que en tal caso será posible estimar las constantes implicadas y calcular así una corrección con lo que mejorar los resultados numéricos obtenidos con ese método. De hecho (ver Isaacson-Keller, pag. 102) si se aplica la expresión de la aceleración de Aitken a un método de iteración funcional simple (por ejemplo, el método de Newton-Raphson) con un orden $p \geq 2$, entonces la sucesión generada converge con orden $2p - 1$

2.7. Códigos disponibles

Puesto que los métodos generales para la aproximación de raíces de funciones (escalares) son relativamente sencillos de programar, es habitual que, cuando se necesita incorporar uno de estos métodos, se programen en vez de buscar un código ya escrito. Sin embargo, sí se pueden encuentrar códigos programados (y depurados!) para la resolución de ecuaciones escalares.

Los códigos más extendidos se basan el denominado método de Brent, que consiste en una combinación del método de bisección con métodos de interpolación (que incluyen el método

de la secante y variantes de orden superior) que buscan hacer una división lo más eficiente posible del intervalo que encierra la raíz (esto es, una reducción lo más rápida posible de dicho intervalo). Pueden consultarse los detalles de este método, por ejemplo, en el texto *Numerical Mathematics* citado al final de este tema.

Así, por ejemplo, la biblioteca GNU Scientific Library (que ya fue mencionada en el primer tema) incluye la función `gsl_root_fsolver_brent` y MATLAB incorpora la función `fzero`, ambas basadas en el método de Brent (Octave y Scilab, como veremos, contienen funciones más generales, que permiten la resolución de sistemas de ecuaciones no lineales, algo de lo que carece MATLAB).

En cualquier caso, pueden encontrarse más funciones para la resolución de ecuaciones escalares en la guía *Guide to Available Mathematical Software*, alojada en la dirección `http://gams.nist.gov/`, en el apartado *Solution of single general nonlinear equations* en `http://gams.nist.gov/serve.cgi/Class/F1b/`.

2.8. Referencias

- **D. Kincaid; W. Cheney; *Análisis Numérico. Las Matemáticas del Cálculo Científico.* Addison-Wesley Iberoamericana, 1994.**

  Existe una edición actualizada de este texto titulada *Numerical Analysis: Mathematics of Scientific Computing. 3rd ed.* publicada por Brooks/Cole en 2002, de la que no existe aún traducción.

- **A. Quarteroni; R. Sacco; F. Saleri; *Numerical Mathematics*: Springer, 2000.**

- **A. Ralston; P. Rabinowitz *First Course in Numerical Analysis.* McGraw-Hill, 1965.**

  Existe una traducción al castellano de este texto (editada por Limusa en 1970 bajo el título *Introducción al análisis numérico*) y una reimpresión (del año 2001) del texto original en la editorial Dover.

Capítulo 3

Resolución de sistemas de ecuaciones lineales y no lineales

3.1. Motivación

Existen multitud de problemas que conducen directamente a la formulación de sistemas de ecuaciones lineales o no lineales (en este último caso, la resolución pasa habitualmente por resolver una sucesión de sistemas de ecuaciones lineales) de tamaños muy diversos:

- en análisis de circuitos (con sistemas que en los casos sencillos irán de 10 a 100 ecuaciones pero cuyo tamaño será mucho mayor en bastantes ocasiones, como en el caso de circuitos integrados)
- en análisis de redes telemáticas o redes eléctricas de potencia (con sistemas de tamaño muy variados, desde unas pocas decenas hasta millones de ecuaciones)
- en discretización de modelos distribuidos regidos por ecuaciones en derivadas parciales en dominios como la acústica o el electromagnestismo (con sistemas en problemas 2D entre 10^3 y 10^5 y sistemas con 10^5 o 10^6 ecuaciones para problemas 3D)

Además, la resolución de sistemas de ecuaciones lineales o no lineales aparece en casi todos los métodos numéricos en alguna parte de los cálculos (por ejemplo, en los métodos numéricos para la integración de problemas de valor inicial asociados a ecuaciones diferenciales ordinarias, existe un grupo amplio de métodos, que denominamos métodos implícitos, que necesitan resolver un sistema de ecuaciones no lineales en cada paso de tiempo).

Por otro lado, los grandes sistemas (muy frecuentes, como se ha mencionado) en la práctica plantean desafíos importantes desde el punto de vista computacional. Es además importante aprovechar el carácter hueco habitual de las matrices (del sistema original en el caso lineal o de la linealización en cada etapa de resolución del caso no lineal):

- en redes (telemáticas o de potencia), aunque el sistema contenga un número muy elevado de nodos, las entradas no nulas en cada fila de la matriz no lo son pues están relacionadas con el número de nodos conectados directamente a uno dado

- en discretización de EDP's el número de entradas no nulas en cada fila está relacionada con los nodos que intervienen en la aproximación de las derivadas y coincide también con los nodos que mantienen una cierta conectividad

Esta propiedad debe ser forzosamente tenida en cuenta tanto para el almacenamiento como para la resolución:

- en almacenamiento, sólo si se guardan exclusivamente las posiciones no nulas es posible almacenar la matriz. Así, para un sistema con un millón de nodos se tiene una memoria de almacenamiento en coma flotante con doble precisión:

$$10^6 \times 10^6 \times 8 \text{ bytes } = 8 \times 10^3 \text{ G bytes}$$

- los métodos de resolución deben tenerlo en cuenta para minimizar los cálculos y adaptarlos a la estructura de la matriz. En los métodos basados en factorizaciones (métodos directos) es preciso minimizar el denominado "fill-in" y adaptar el cálculo de la factorización. En los métodos iterativos de tipo gradiente es preciso construir de forma eficiente los productos matriz por vector (sin almacenar la matriz completa) para acelerar los cálculos.

3.2. Condicionamiento del problema y clasificación de los métodos

En esta sección se considera, en primer lugar, el estudio del condicionamiento del problema de resolución de un sistema de ecuaciones lineales para, a continuación, clasificar los métodos numéricos para la resolución de estos sistemas.

Condicionamiento de sistemas de ecuaciones lineales

Como en todo problema que se resuelve numéricamente (y, en realidad, en todo modelo que se resuelve) es preciso identificar cuál será el efecto que tendrán los errores de redondeo (y, si es posible, también los errores de modelado) en la solución numérica obtenida en la práctica (como resultado de un programa ejecutado en un ordenador). Un modo elemental de estudiar este efecto es analizando cómo influye la perturbación de los elementos de la matriz o el vector a la solución obtenida. Una alternativa es analizar la relación existente entre el residuo (que en los denominados métodos directos constituye un modo de considerar los redondeos y en los métodos iterativos constituirá un indicador de error) y el error.

Puesto que en cualquiera de estas medidas aparece la necesidad de cuantificar el *tamaño* de una matriz, es preciso comenzar por la construcción de normas matriciales. Recuérdese que, dado un espacio vectorial V se define una norma $\|\cdot\|$, como una aplicación con las siguientes propiedades:

(P1) $\|x\| > 0,\ \forall x \in V,\ x \neq 0$

(P2) $\|\lambda x\| = |\lambda|\,\|x\|\ \ \forall \lambda \in \mathbf{R},\ \forall \mathbf{x} \in \mathbf{V}$

(P3) $\|x + y\| \leq \|x\| + \|y\|\ \ \forall x, y \in V$

y las habituales normas $\|\cdot\|_1$, $\|\cdot\|_2$, y $\|\cdot\|_\infty$ definidas para el espacio vectorial $\mathbf{R^N}$. En el caso de las matrices cuadradas es posible "inducir " normas (sobre el correspondiente espacio vectorial) a partir de las normas vectoriales gracias a la siguiente propiedad

Proposición 3.1 *Si la aplicación $\|\cdot\|$ constituye una norma sobre $\mathbf{R^N}$, entonces la aplicación*

$$\|A\|_M = \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} \|A\vec{x}\|$$

constituye una norma sobre el espacio vectorial de las matrices $N \times N$ (que denominaremos norma subordinada)

Demostración:

(P1) puesto que A debe tener al menos una columna distinta de cero, si se toma $\vec{x} = \vec{e}_j$ (j-ésimo vector de la base, con j igual al número de columna $A^{(j)}$ no nula):

$$\|A\|_M \geq \|A\,\frac{\vec{e}_j}{\|\vec{e}_j\|}\| = \frac{1}{\|\vec{e}_j\|}\,\|A^{(j)}\| > 0$$

(P2)

$$\|\lambda A\|_M = \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} \|\lambda A\vec{x}\| = |\lambda| \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} \|A\vec{x}\| = |\lambda|\,\|A\|_M$$

(P3)

$$\begin{aligned}\|A + B\|_M &= \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} \|(A+B)\vec{x}\| \leq \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} (\|A\vec{x}\| + \|B\vec{x}\|) \\ &\leq \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} \|A\vec{x}\| + \sup_{\vec{x}\in\mathbf{R^N}\,,\,\|\vec{\mathbf{x}}\|=\mathbf{1}} \|B\vec{x}\| = \|A\|_M + \|B\|_M\end{aligned}$$

□

En relación con las normas matriciales es preciso hacer algunas obervaciones:

- De la definición de norma subordinada se desprende inmediatamente:

$$\|A\vec{x}\| \leq \|A\|_M\,\|\vec{x}\|\,,\quad \forall \vec{x} \in \mathbf{R^N}$$

que constituye una propiedad importante.

- Es posible caracterizar la norma $\| \|_\infty$ como

$$\|A\|_\infty = \max_{1\le i\le N} \sum_{j=1}^{N} |a_{ij}|$$

Se tiene que

$$\begin{aligned} \|A\|_\infty &= \sup_{\vec{x}\in\mathbf{R^N},\, \|\vec{\mathbf{x}}\|_\infty=\mathbf{1}} \|A\vec{x}\|_\infty = \sup_{\vec{x}\in\mathbf{R^N},\, \|\vec{\mathbf{x}}\|_\infty=\mathbf{1}} \max_{1\le i\le N} |\sum_{j=1}^{N} a_{ij}x_j| \\ &= \max_{1\le i\le N} \sup_{\vec{x}\in\mathbf{R^N},\, \|\vec{\mathbf{x}}\|_\infty=\mathbf{1}} |\sum_{j=1}^{N} a_{ij}x_j| \\ &= \max_{1\le i\le N} \sum_{j=1}^{N} |a_{ij}| \end{aligned}$$

- Para las normas $\| \|_1$ y $\| \|_2$ se tiene

$$\begin{aligned} \|A\|_1 &= \max_{1\le j\le N} \sum_{i=1}^{N} |a_{ij}| \\ \|A\|_2 &= \sqrt{\rho(AA^t)} = \max_{1\le i\le n} |\lambda_i(AA^t)|^{1/2} \end{aligned}$$

lo que la convierte en difícil de calcular, aunque se sabe que

$$\|A\|_2 \le \sqrt{\|A\|_1 \, \|A\|_\infty}$$

Consideremos ahora una perturbación del término del segundo miembro y examinemos el efecto sobre la solución. Se tiene:

$$\begin{aligned} \text{Problema “exacto”:} \qquad & A\vec{x} = \vec{b} \\ \text{Problema perturbado:} \qquad & A\vec{x}^* = \vec{b}^* \end{aligned}$$

y para su diferencia

$$A(\vec{x} - \vec{x}^*) = \vec{b} - \vec{b}^*$$

o bien, si A es regular

$$\vec{x} - \vec{x}^* = A^{-1}\,(\vec{b} - \vec{b}^*)$$

Si queremos ahora una medida relativa debemos comparar $\|\vec{x} - \vec{x}^*\|$ con $\|\vec{x}\|$ de modo que empleando

$$\|\vec{b}\| = \|A\vec{x}\| \le \|A\|_M \, \|\vec{x}\|$$

obtenemos finalmente

$$\frac{\|\vec{x} - \vec{x}^*\|}{\|\vec{x}\|} = \frac{1}{\|\vec{x}\|}\|A^{-1}\,(\vec{b} - \vec{b}^*)\| \le \frac{1}{\|\vec{x}\|}\|A^{-1}\|_M \, \|\vec{b} - \vec{b}^*\| \le \frac{\|A\|_M}{\|\vec{b}\|}\|A^{-1}\|_M \, \|\vec{b} - \vec{b}^*\|$$

esto es

$$\frac{\|\vec{x}-\vec{x}^*\|}{\|\vec{x}\|} \leq (\,\|A\|_M\,\|A^{-1}\|_M\,)\,\frac{\|\vec{b}-\vec{b}^*\|}{\|\vec{b}\|}$$

Se tiene así que $\|A\|_M\,\|A^{-1}\|_M$ ofrece una cota superior para la amplificación de las perturbaciones sobre el término de segundo miembro. A dicho número se le denomina *número de condicionamiento de la matriz* A y se suele representar por $\kappa(A)$ (o simplemente κ si no hay ambigüedad posible).

Alternativamente, podemos considerar la relación entre el error cometido en la resolución del sistema ($\|\vec{x}-\vec{x}^*\|$) y la norma del residuo ($\|A\vec{x}^*-\vec{b}\|$), que es además, un elemento útil en la elaboración de criterios de parada para los métodos iterativos (no estacionarios).

Sean entonces

$$\begin{aligned} A\vec{x} &= \vec{b} \\ A\vec{x}^* &= \vec{b}+\vec{r} \end{aligned}$$

de donde

$$A(\vec{x}-\vec{x}^*) = -\vec{r}$$

y por lo tanto,

$$\begin{aligned} \vec{x}-\vec{x}^* &= -A^{-1}\,\vec{r} \\ \|\vec{x}-\vec{x}^*\| &= \|-A^{-1}\,\vec{r}\| \leq \|A^{-1}\|_M\,\|\vec{r}\| \end{aligned}$$

y en términos relativos

$$\frac{\|\vec{x}-\vec{x}^*\|}{\|\vec{x}\|} \leq (\,\|A\|_M\,\|A^{-1}\|_M\,)\frac{\|\vec{r}\|}{\|\vec{b}\|}$$

de forma que el número de condición de la matriz también sirve para medir esta amplificación.

Como en la resolución de ecuaciones escalares, el condicionamiento juega un papel importante en la resolución numérica de los correspondientes problemas. En la presentación de los métodos para la resolución de sistemas de ecuaciones lineales (tanto directos como iterativos) se insistirá sobre el modo de evitar el deterioro de la solución numérica por efecto del mal condicionamiento de las matrices (pues una gran parte de los problemas asociados a grandes sistemas de ecuaciones –provenientes fundamentalmente de la discretización de ecuaciones en derivadas parciales– están muy mal condicionados). Como se verá además en los métodos iterativos el mal condicionamiento de las matrices lleva no sólo a la amplificación de los errores de redondeo sino a impedir la convergencia práctica de los métodos (y de ahí la necesidad de emplear precondicionadores), así como a la dificultad de estimar el error a partir de la medida del residuo. En realidad ambas cuestiones están relacionadas pues los métodos iterativos habituales (como gradiente conjugado o GMRES) están basados en la minimización de los residuos.

Sin embargo, a diferencia de lo que ocurre en la resolución de ecuaciones escalares, en este caso no se tiene acceso sencillo al número de condicionamiento de la matriz.

Así, en las ecuaciones escalares bastaba con obtener una estimación de $F'(x_*)$ pero aquí es necesario obtener una estimación de $\|A\|_M \, \|A^{-1}\|_M$. En principio no es demasiado costoso obtener $\|A\|_M$ para algunas normas, pero no puede contarse con calcular A^{-1} para conocer $\|A^{-1}\|_M$.

Existen algunas técnicas para tratar de estimar $\|A^{-1}\|_M$. En el texto de Dahlquist y Bjork recogido al final de este tema (capítulo 7, página 90) puede consultarse un modo de acotar inferiormente $\|A^{-1}\|_M$ (aunque sólo se trate de una cota inferior, en la práctica devuelve el orden de magnitud correcto de $\|A^{-1}\|_M$). Se basa en que si $\vec{x}$ es solución de

$$A\vec{x} = \vec{w}$$

entonces se tiene

$$\|\vec{x}\| = \|A^{-1}\,\vec{w}\| \leq \|A^{-1}\|_M \, \|\vec{w}\|$$

y así

$$\|A^{-1}\|_M \geq \frac{\|\vec{x}\|}{\|\vec{w}\|}$$

donde se debe elegir w adecuadamente (y resolver el sistema de ecuaciones lineales aprovechando la factorización si se emplea un método directo).

En muchos casos, sin embargo, se dispone de estimaciones del número de condicionamiento vinculadas a la naturaleza del problema que se resuelve. Este es el caso (bien estudiado) de la discretización de ecuaciones en derivadas parciales en 2D o 3D mediante métodos de diferencias finitas o métodos de elementos finitos. De hecho, todos estos problemas (estacionarios) están muy mal condicionados y su condicionamiento se deteriora rápidamente con el refinamiento de la malla (lo que hace interesante el empleo de métodos directos a pesar de otras posibles desventajas).

Precondicionamiento de sistemas de ecuaciones lineales

En el caso de sistemas (muy) mal condicionados, se hace preciso reescribir previamente el sistema de ecuaciones para tratar con matrices con números de condicionamiento más bajos y evitar así que los inevitables errores de redondeo asociados a una aritmética finita deterioren por completo la solución. Así, dado un sistema de ecuaciones lineales

$$A\vec{x} = \vec{b}$$

(muy) mal condicionado, se propone resolver un sistema modificado, obtenido al multiplicar a la izquierda por una cierta matriz regular B el sistema original

$$BA\vec{x} = B\vec{b}$$

con la intención de que el condicionamiento del nuevo sistema $\kappa(BA)$ sea mucho más reducido.

Obsérvese que, idealmente, desearíamos tener $\kappa(BA) = 1$ (menor número de condición posible, asociado a la matriz identidad) lo cual se lograría si se toma $B = A^{-1}$. Obviamente, ésta no es una elección válida (recuérdese que encontrar la inversa de la matriz A obliga a resolver n sistemas de ecuaciones lineales asociados a la matriz A) pero sí sirve como guía para la elección de la matriz B, que debe parecerse en la medida de lo posible a A^{-1}.

En la práctica, se suele evitar la construcción explícita de la matriz BA y simplemente se calcula la *acción* de la matriz B sobre un vector (el producto de la matriz por el vector). Como por otro lado, B se debe parecer a A^{-1} se suele notar como P^{-1} (donde la resolución de un sistema de ecuaciones lineales asociado a la matriz P deberá parecerse a la solución con la matriz A, ya que eso es lo que implica que se parezcan P^{-1} y A^{-1}). Además, el producto de P^{-1} por un vector equivale a resolver un sistema de ecuaciones con matriz P y término de segundo miembro el vector dado. Este hecho impone que P debe ser elegida de modo que sea sencillo resolver un sistema de ecuaciones lineales asociado a dicha matriz.

En la práctica, dadas las dos condiciones impuestas sobre P (P debe parecerse a A y los sistemas asociados a P deben resultar fácilmente resolubles) y el hecho de que lo que realmente debe hacerse es obtener el producto de P^{-1} por un vector, se suelen tomar como P^{-1} una resolución aproximada del sistema de ecuaciones lineales asociado a la matriz A (véase en la siguiente sección la clasificación de los métodos y en las secciones posteriores la descripción de cada método):

- mediante los métodos directos a través de factorizaciones incompletas (por ejemplo, la factorización LU incompleta)
- mediante los métodos iterativos clásicos a través de un número prefijado de iteraciones (por ejemplo, con el método de Jacobi o el método de relajación)

Clasificación de los métodos

Existe una clasificación de los métodos en dos grandes grupos:

a) métodos directos, que tratan de calcular la solución exacta en un número prefijado de cálculos (en la práctica los errores de redondeo harán imposible conseguir la solución exacta y en grandes sistemas, especialmente si están mal condicionados es necesario garantizar que la programación y el esquema sean robustos ante estos redondeos)

b) métodos iterativos, que generan una sucesión de aproximaciones de la solución del sistema de ecuaciones (a su vez dentro de los métodos iterativos se distinguirá entre los denominados métodos clásicos - o estacionarios - y los métodos de tipo gradiente)

La elección entre un tipo u otro de métodos está condicionado por el tipo de problema que se resuelve, pero también por la eficiencia del software que se emplee o la máquina sobre la que se trabaje. Como norma general, es habitual considerar la resolución mediante métodos

directos (siempre y cuando estén bien programados) para sistemas no demasiado grandes (es éste desde luego un concepto variable en el tiempo, pero en 2007 podemos fijar el tamaño límite en $N \sim 10^5$ y con métodos iterativos (de tipo gradiente) para sistemas mayores. La razón es que, con una formulación adecuada (en particular con pivotes y reescalado), los métodos directos son relativamente *inmunes* a la amplificación de los errores de redondeo y su principal defecto es sólo el número de operaciones ($\simeq 1/3N^3$, que los hace inadecuados, junto con los problemas de almacenamiento, para sistemas grandes).

No obstante, además del tamaño hay numerosas cuestiones que deben ser tenidas en cuenta a la hora de seleccionar el método de resolución:

- Si es preciso resolver varias veces un sistema de ecuaciones cambiando exclusivamente el vector de segundo miembro (algo muy habitual en la integración de ecuaciones en derivadas parciales evolutivas e incluso en la resolución de ecuaciones en derivadas parciales elípticas no lineales si no se actualiza el Jacobiano con cada paso; también en cálculo de estructuras ante distintas hipótesis de carga o en la resolución de redes de potencia con distintas hipótesis de consumo) el coste de los métodos directos para los sucesivos sistemas (una vez calculada y almacenada la factorización) es mínimo. (De forma grosera, el coste es de un orden N^2 y por lo tanto, el coste relativo es del orden de $1/N$.

- Si se tiene un sistema muy mal condicionado y no se sabe construir un buen precondicionador (y esto no es obvio en general) la convergencia de los métodos iterativos (de tipo gradiente, pues los clásicos rara vez se emplearán) puede ser muy lenta y hacer que los tiempos de cálculo totales sean menores para los métodos directos.

- En sistemas muy grandes con matrices huecas (que constituye el caso habitual, como se ha mencionado, cuando se encuentran grandes sistemas) los métodos directos originan un *rellenado* en el cálculo de las factorizaciones (denominado *fill-in*) que puede originar graves problemas de almacenamiento (que se sumarán al elevado número de operaciones).

- En muchos casos, el sistema de ecuaciones que se resuelve proviene de un modelo que ya ha incorporado diversos errores (de modelado, de medida o de truncamiento asociado a otros métodos numéricos que conducen a la formulación del sistema); en esta situación podría estar justificada la aproximación de la solución con precisión poco estricta y merecer la pena acudir a los métodos iterativos (aunque la ventaja sólo será clara si el sistema es grande).

- Adicionalmente, existen métodos especialmente adaptados a problemas específicos. Este es el caso de los métodos multimalla (geométricos) para la resolución de los sistemas originados en la discretización de ecuaciones en derivadas parciales elípticas, o los métodos para resolver sistemas asociados a matrices de Vandermonde que aparecen en los problemas de interpolación.

3.3. Métodos directos

Como ya se ha comentado, los métodos directos tratan de calcular la solución exacta en un número prefijado de cálculos. A continuación recordaremos el método de Gauss, que constituye la base de los métodos directos (aunque el lector haya quizás estudiado en algún momento la regla Cramer como método directo para la resolución de sistemas de ecuaciones lineales, es fácil convencerse de la absoluta ineficiencia de dicho método para resolver sistemas de más de tres ecuaciones y, en consencuencia, del nulo interés práctico de la regla de Cramer).

Método de Gauss

Como se sabe, el método de Gauss (o método de eliminación, como se denomina con frecuencia) propone una sucesión de operaciones elementales que elimine ordenadamente determinadas incógnitas de las ecuaciones hasta lograr un sistema triangular (superior) de modo que las incógnitas puedan irse despejando sucesivamente a partir de la última ecuación. A continuación se recuerda con un ejemplo elemental el algoritmo de eliminación.

Ejemplo 3.1 *Se considera el sistema de ecuaciones*

$$\left\{\begin{array}{rrrcl} 4x_1 & -x_2 & +x_3 & = & 8 \\ 2x_1 & +5x_2 & +2x_3 & = & 3 \\ x_1 & +2x_2 & +4x_3 & = & 11 \end{array}\right.$$

que se escribe matricialmente

$$\left(\begin{array}{rrr} 4 & -1 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{array}\right)\left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right)=\left(\begin{array}{c} 8 \\ 3 \\ 11 \end{array}\right)$$

Se trata de hacer en cada etapa k-ésima ceros en la columna k por debajo de la diagonal principal. Para ello ha de tenerse que el elemento a_{kk} de cada etapa k sea distinto de cero.

Etapa 1 del proceso de eliminación gaussiana:

Comencemos haciendo ceros en la primera columna debajo del elemento a_{11}; para ello hemos de sumarle a la segunda fila la primera multiplicada por $-1/2$ y a la tercera la primera fila multiplicada por $-1/4$; esto es equivalente a

$$\left(\begin{array}{rrr} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/4 & 0 & 1 \end{array}\right)\left(\begin{array}{rrr} 4 & -1 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{array}\right)\left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right)=\left(\begin{array}{rrr} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/4 & 0 & 1 \end{array}\right)\left(\begin{array}{c} 8 \\ 3 \\ 11 \end{array}\right)$$

$$\left(\begin{array}{rrr} 4 & -1 & 1 \\ 0 & 11/2 & 3/2 \\ 0 & 9/4 & 15/4 \end{array}\right)\left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right)=\left(\begin{array}{r} 8 \\ -1 \\ 9 \end{array}\right)$$

Etapa 2 del proceso de eliminación gaussiana:

Debemos hacer ahora un cero debajo del elemento $11/2$, así que le sumamos a la tercera fila la segunda multiplicada por $-9/22$ o lo que es lo mismo

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -9/22 & 1 \end{pmatrix} \begin{pmatrix} 4 & -1 & 1 \\ 0 & 11/2 & 3/2 \\ 0 & 9/4 & 15/4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -9/22 & 1 \end{pmatrix} \begin{pmatrix} 8 \\ -1 \\ 9 \end{pmatrix}$$

$$\begin{pmatrix} 4 & -1 & 1 \\ 0 & 11/2 & 3/2 \\ 0 & 0 & 69/22 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 8 \\ -1 \\ 207/22 \end{pmatrix}$$

Finalmente, resolvemos el sistema cuya matriz es triangular superior.

Remonte tras eliminación gaussiana:

$$\begin{aligned} x_3 &= \frac{22}{69}\,\frac{207}{22} = 3 \\ x_2 &= \frac{2}{11}(-1 - \frac{3}{2}x_3) = -1 \\ x_1 &= \frac{1}{4}(8 + x_2 - x_3) = 1 \end{aligned}$$

Desde el punto de vista matricial, el método de Gauss para resolver un sistema de la forma $A\,\vec{x} = \vec{b}$ trata de determinar una matriz regular M de forma que MA sea una matriz triangular superior (proceso de eliminación) y resolver a continuación el sistema equivalente $MA\,\vec{x} = M\vec{b}$ (proceso de remonte). Hay que tener en cuenta que en los cálculos efectivos no se calcula explícitamente la matriz M sino solamente la matriz MA y el vector $M\vec{b}$, conduciendo al sistema de ecuaciones equivalente

$$MA\vec{x} = M\vec{b}.$$

Si denotamos mediante U la matriz (triangular superior) MA y tenemos en cuenta que al ser la matriz M regular podemos multiplicar a la izquierda por M^{-1} para recuperar una reescritura del sistema de ecuaciones original

$$M^{-1}U\vec{x} = \vec{b}$$

Finalmente, basta con observar cómo se ha procedido a la eliminación de los elementos de A por debajo de la diagonal para convencerse de que la matriz M^{-1} debe ser triangular inferior. Llamando entonces L a la matriz (triangular inferior) M^{-1} se ha reescrito A como el producto de las matrices L y U. A dicha reescritura de A se le denomina factorización LU de la matriz A.

De este modo, dado un sistema de ecuaciones

$$A\vec{x} = \vec{b}$$

si descomponemos $A = LU$ (con L triangular inferior y U triangular superior) es posible resolver el sistema mediante una etapa de descenso

$$L\vec{y} = \vec{b}$$

y otra de remonte

$$U\vec{x} = \vec{y}$$

Ejemplo 3.2 *Retomemos el ejemplo anterior y calculemos la factorización LU de la matriz del sistema*

$$\begin{pmatrix} 4 & -1 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 3 \\ 11 \end{pmatrix}$$

Empleando las matrices E_1 y E_2 asociadas a las operaciones elementales

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1/4 & 0 & 1 \end{pmatrix} \qquad E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -9/22 & 1 \end{pmatrix}$$

y llamando U a la matriz triangular E_2E_1A

$$U = \begin{pmatrix} 4 & -1 & 1 \\ 0 & 11/2 & 3/2 \\ 0 & 0 & 69/22 \end{pmatrix}$$

puede escribirse (E_1 y E_2 regulares)

$$E_1^{-1}E_2^{-1}U = A$$

Definiendo ahora

$$L = E_1^{-1}E_2^{-1}$$

se tiene

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 9/22 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/4 & 9/22 & 1 \end{pmatrix}$$

En suma, se ha obtenido

$$A = \begin{pmatrix} 4 & -1 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/4 & 9/22 & 1 \end{pmatrix} \begin{pmatrix} 4 & -1 & 1 \\ 0 & 11/2 & 3/2 \\ 0 & 0 & 69/22 \end{pmatrix}$$

y la resolución del sistema de ecuaciones puede hacerse mediante una etapa de descenso (que determina los valores auxiliares y_1, y_2 e y_3)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/4 & 9/22 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 3 \\ 11 \end{pmatrix}$$

y otra de remonte (para calcular x_1, x_2 y x_3)

$$\begin{pmatrix} 4 & -1 & 1 \\ 0 & 11/2 & 3/2 \\ 0 & 0 & 69/22 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

□

Desde luego, la principal aplicación de la factorización LU es la resolución de sistemas de ecuaciones lineales (como se acaba de ver, la factorización LU es, en realidad, una reescritura del método de eliminación de Gauss). Sin embargo, esta factorización tiene algunas otras aplicaciones. Así, supongamos que una cierta matriz A admite factorización LU. Entonces, podemos calcular a partir de ella:

- El determinante de la matriz A, puesto que:

$$det(A) = det(L)det(U) = \Pi_{i=1}^{N} u_{ii}$$

- La inversa de la matriz A resolviendo N sistemas de ecuaciones de la forma $A\vec{x}^j = e^j$, de modo que A^{-1} se construye a través de los N vectores $\vec{x}^j$. Obsérvese que el coste de la resolución de cada sistema no es muy elevado puesto que la factorización se calcula una vez y se almacena.

Se plantea ahora la cuestión acerca de si todo sistema de ecuaciones puede ser resuelto mediante el método de Gauss o, equivalentemente, si toda matriz admite una factorización LU. La respuesta (insatisfactoria, como se verá) viene dada por el siguiente resultado.

Proposición 3.2 *(condición suficiente para la existencia de factorización LU)*
Sea A una matriz cuadrada $N \times N$ tal que sus N menores principales son distintos de cero. Entonces la matriz A tiene una factorización LU donde L es una matriz triangular inferior con unos en la diagonal principal y U es una matriz triangular superior.

Demostración: Puede consultarse en el texto de Kincaid y Cheney (página 136).

□

Cabe hacer aquí dos observaciones sobre el resultado que se acaba de exponer:

- La condición suficiente de existencia de factorización LU conlleva un coste muy elevado pues requiere calcular numerosos determinantes. De hecho, como se acaba de ver, un modo eficiente de calcular el determinante de una matriz es precisamente a través de su factorización LU. Así, no resulta una condición adecuada para examinar previamente si una matriz admitirá o no factorización pues exigiría más cálculos que la propia factorización; sería entonces preferible intentar calcular directamente la factorización.

- El método de Gauss aparece así como un método poco satisfactorio para la resolución de sistemas de ecuaciones lineales, pues existen matrices regulares que no satisfacen la condición anterior y el resultado no permite asegurar entonces que el método de Gauss pueda resolver un sistema de ecuaciones asociado a dicha matriz.

Veamos ahora que, en efecto, existen sistemas de ecuaciones lineales que no pueden ser resueltos mediante el método de Gauss en la formulación elemental que hemos presentado

Ejemplo 3.3 *Se considera el siguiente sistema de ecuaciones lineales*

$$\begin{array}{rcl} x_1 - x_2 + 3x_3 & = & 3 \\ 3x_1 - 3x_2 + x_3 & = & 1 \\ x_1 + x_2 + x_3 & = & 3 \end{array}$$

El proceso de eliminación comienza haciendo cero los coeficientes de la matriz por debajo de la diagonal en la primera columna. Para ello se deberá multiplicar a la izquierda a la matriz del sistema A por una matriz

$$\begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

que corresponde (pensando en términos del método de Gauss) a sustituir la segunda ecuación del sistema por la ecuación resultante de restarle 3 *veces la primera ecuación a la segunda, y sustituir la tercera ecuación por aquella que se obtiene tras restar la primera ecuación de la última.*

Sin embargo, es fácil ver que el resultado no permite continuar el proceso de eliminación para hacer cero los coeficientes por debajo de la diagonal en la segunda columna ya que

$$\begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 3 \\ 3 & -3 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 3 \\ 0 & 0 & -8 \\ 0 & 2 & -2 \end{pmatrix}$$

Ahora bien, pensando de nuevo en términos del método de Gauss, el resultado no podía ser mejor ya que el sistema de ecuaciones resultante de las combinaciones mencionadas es:

$$\begin{array}{rrrcr} x_1 & -x_2 & +3x_3 & = & 3 \\ & & -8x_3 & = & -8 \\ & 2x_2 & -2x_3 & = & 0 \end{array}$$

y por lo tanto no necesita ningún proceso adicional de eliminación ya que la segunda ecuación permite obtener x_3 y, a partir de ésta, la tercera ecuación devuelve x_2 y, finalmente, de la primera ecuación se despeja x_1. Es cierto que éste no es estrictamente un proceso de remonte (pues primero se resuelve la segunda ecuación, después la tercera y finalmente la primera), pero parece obvio cómo lograr que esta última etapa de resolución sea un remonte: basta con reordenar las dos últimas ecuaciones y escribir el sistema como

$$\begin{array}{rrrcr} x_1 & -x_2 & +3x_3 & = & 3 \\ & 2x_2 & -2x_3 & = & 0 \\ & & -8x_3 & = & -8 \end{array}$$

Esta simple observación permite resolver entonces la dificultad con el cálculo de la factorización LU: bastaba con intercambiar la segunda y tercera filas de la matriz A (que equivale a intercambiar desde el principio el orden de las ecuaciones).

Al proceso descrito en el ejemplo anterior para calcular la factorización LU de una matriz (que intercambia las filas de la matriz) se le denomina *factorización indirecta*. De forma más precisa, se dice que una matriz A admite una factorización LU indirecta si existe una cierta matriz P de permutaciones (matriz con ceros y unos como elementos, organizados de modo que cada fila y cada columna tienen un y sólo un elemento igual a 1) tal que la matriz PA (que corresponde a la matriz A con sus filas reordenadas de acuerdo con la matriz de permutación) admite factorización LU.

A continuación se verá un resultado que asegura que, para toda matriz regular siempre es posible, a través quizás de la reordenación de las ecuaciones, la aplicación del método de Gauss para resolver el sistema de ecuaciones lineales. En términos matriciales, se va a asegurar que toda matriz regular admite factorización LU indirecta.

Proposición 3.3 *Sea A una matriz regular. Entonces existe una matriz de permutación P tal que $PA = LU$ donde L es una matriz triangular inferior con unos en la diagonal principal y U es una matriz triangular superior.*

Demostración: Este resultado se basa en que, en la etapa k-ésima del proceso de eliminación, puede suceder que $a_{kk} = 0$ y, por tanto, no pueda usarse la fila k-ésima para eliminar los elementos de la columna k-ésima que están por debajo de la diagonal principal. Lo que puede hacerse en este caso es intercambiar la fila k-ésima con alguna fila posterior para conseguir un elemento pivote que no sea cero; si esto no puede hacerse, entonces la matriz de coeficientes del sistema es singular y el sistema no tiene solución única.

□

En relación con la factorización LU indirecta debe observarse que:

- El intercambio de filas de la matriz A debe acompañarse del intercambio de elementos de b cuando se trata de resolver un sistema $A\vec{x} = \vec{b}$. Así, supongamos que la matriz A

admite una factorización LU indirecta, es decir, $P\,A = L\,U$. Para resolver un sistema de la forma $A\vec{x} = b$ se resolverá el sistema $L\,\vec{y} = P\,\vec{b}$ por sustitución progresiva y a continuación $U\,\vec{x} = \vec{y}$ por sustitución regresiva.

- En la práctica, dada una cierta matriz no es inmediato saber si es o no una matriz regular (salvo que el problema que ha originado el sistema de ecuaciones lineales posea alguna propiedad que permita concluir la regularidad o singularidad de la matriz). De hecho, el criterio más inmediato para saber si la matriz es o no regular (el cálculo de su determinante), pasaría en la práctica por obtener su factorización LU. Así, el (intento de) cálculo de la factorización LU indirecta de una matriz A sirve precisamente para conocer si dicha matriz es o no regular.

Se ha obtenido entonces un algoritmo (el método de eliminación de Gauss con posible intercambio de ecuaciones) para la resolución (exacta) de cualquier sistema de ecuaciones lineales con matriz regular.

Una estimación del orden de magnitud del coste computacional de este algoritmo (donde se evaluará exclusivamente el número de productos que se deben efectuar, puesto que el resto de las operaciones son sumas que se ejecutan en un tiempo mucho más corto o divisiones cuyo número es muy inferior) devuelve un orden de N^3 (donde N representa el número de ecuaciones). Obsérvese que dicho orden se obtiene del hecho de que es preciso repetir el proceso de eliminación de términos por debajo de la diagonal $(N-1)$ veces y cada proceso de eliminación opera sobre k filas (con k el número de filas, que irá reduciéndose en una unidad desde la primera columna, donde será $k = N-1$, de modo que en la j-ésima etapa se tendrán $N-j+1$ filas) de k elementos cada una, realizando por lo tanto un número de productos del orden de N^2 (una estimación fina devuelve $\frac{1}{3}N^2$). Así, el coste de cálculo del método de eliminación de Gauss crece rápidamente con el tamaño del sistema de ecuaciones y es precisamente este hecho lo que hará necesario encontrar métodos aleternativos cuando se trate de resolver sistemas de gran tamaño.

Por otro lado, es preciso hacer una observación sobre este método (y, en general, sobre los métodos directos). El método de Gauss devolvería la solución exacta de un sistema de ecuaciones tras un número finito de operaciones (que puede, como acabamos de observar, ser grande) siempre y cuando éstas se realicen con aritmética exacta. Lógicamente esto no ocurre prácticamente nunca cuando se trabaja con un algoritmo programado sobre un ordenador (tendría que ocurrir que durante la aplicación del algoritmo se obtuviesen exclusivamente números racionales representables de forma exacta en el ordenador sobre el que se trabaja). En consecuencia, en sentido estricto los métodos directos también devuelven (como los métodos iterativos) una solución aproximada del sistema de ecuaciones, aunque aquí la aproximación es debida tan sólo a los errores de redondeo y su propagación a lo largo de los cálculos. Como se va a ver, en algunas ocasiones esto conducirá a errores muy reducidos (manteniéndose el orden de magnitud del error de redondeo) mientras que en otros el redondeo puede originar (debido a la propagación y amplificación de los errores de redondeo) errores enormes.

Ejemplo 3.4 *Se considera de nuevo el sistema de ecuaciones*

$$\left\{\begin{array}{llll} 4x_1 & -x_2 & +x_3 & = & 8 \\ 2x_1 & +5x_2 & +2x_3 & = & 3 \\ x_1 & +2x_2 & +4x_3 & = & 11 \end{array}\right.$$

y su resolución mediante el método de eliminación de Gauss empleando una aritmética pseudo-decimal finita. En particular, se emplea una hipotética aritmética finita con coma flotante que reserva cuatro dígitos (decimales) para la mantisa, junto con otro dígito para el exponente y un bit para el signo:

$$f = \pm a_1.a_2a_3a_4 \times 10^{b_1}$$

Etapa 1 del proceso de eliminación gaussiana:

$$\left\{\begin{array}{llll} +4.000 \times 10^0\, x_1 & -1.000 \times 10^0\, x_2 & +1.000 \times 10^0\, x_3 & = & +8.000 \times 10^0 \\ & +5.500 \times 10^0\, x_2 & +1.500 \times 10^0\, x_3 & = & -1.000 \times 10^0 \\ & +2.250 \times 10^0\, x_2 & +3.750 \times 10^0\, x_3 & = & +9.000 \times 10^0 \end{array}\right.$$

Etapa 2 del proceso de eliminación gaussiana:

$$\left\{\begin{array}{llll} +4.000 \times 10^0\, x_1 & -1.000 \times 10^0\, x_2 & +1.000 \times 10^0\, x_3 & = & +8.000 \times 10^0 \\ & +5.500 \times 10^0\, x_2 & +1.500 \times 10^0\, x_3 & = & -1.000 \times 10^0 \\ & & +3.136 \times 10^0\, x_3 & = & +9.409 \times 10^0 \end{array}\right.$$

Remonte tras eliminación gaussiana:

$$\begin{array}{lll} x_3 & = +3.000 \times 10^0 & \\ x_2 & = -1.818 \times 10^0 - 8.182 \times 10^0 & = -1.000 \times 10^0 \\ x_1 & = +2.000 \times 10^0 - 7.500 \times 10^{-1} + 2.500 \times 10^{-1} & = +1.000 \times 10^0 \end{array}$$

□

En el ejemplo que se acaba de mostrar, los errores de redondeo en cada etapa (asociados al truncamiento de los resultados para poder ser representados en la aritmética finita) originan en la solución del sistema de ecuaciones errores del mismo orden de magnitud, de modo que la solución obtenida es, para los dígitos retenidos en esa aritmética finita, exacta. Sin embargo, ésta no será siempre la situación. A continuación se verá un segundo ejemplo donde los errores de redondeo sí pueden amplificarse a lo largo de los cálculos y se mostrará un modo de evitar que dicha amplificación se produzca.

Ejemplo 3.5 *Se considera el sistema de ecuaciones*

$$\left\{\begin{array}{llll} 0.003\, x_1 & +59.14\, x_2 & = & 59.17 \\ 5.291x_1 & -6.130x_2 & = & 46.78 \end{array}\right.$$

con solución exacta $x_1 = 10$, $x_2 = 1$ y su resolución mediante el método de Gauss empleando la misma hipotética aritmética finita que en el ejemplo anterior.
Etapa de eliminación gaussiana:

$$\left\{\begin{array}{lll} +3.000\times 10^{-3}\, x_1 & +5.914\times 10^{1}\, x_2 & = +5.917\times 10^{1} \\ & -1.043\times 10^{5}\, x_2 & = +1.044\times 10^{5} \end{array}\right.$$

Remonte tras eliminación gaussiana:

$$\begin{array}{ll} x_2 & = +1.001\times 10^{0} \\ x_1 & = +1.972\times 10^{4} - 1.971\times 10^{4} \quad = -1.000\times 10^{1} \end{array}$$

Obsérvese que aunque el valor obtenido para la variable x_2 contiene un error de solamente el $0.1\,\%$*, la aproximación obtenida para la variable x_1 es desastrosa: se ha cometido un error del* $200\,\%$*, lo que supone una amplificación del error de redondeo con un factor de* 10^6*.*

Puede, desde luego, objetarse que se está trabajando con una aritmética finita muy pobre *(lo cual es absolutamente cierto). Sin embargo, obsérvese también que se ha considerado un sistema del orden más pequeño posible. De forma grosera, la propagación del efecto de los redondeos a lo largo de la resolución del sistema de ecuaciones lineales tiene que ver con ambos parámetros y lo que aquí ha ocurrido con una precisión muy grosera en un sistema muy pequeño ilustra lo que ocurrirá en sistemas de tamaños realistas con precisiones también realistas.*

Podría quizás atribuirse el enorme error al mal condicionamiento del problema. Sin embargo, para convencerse de que esto no es así basta con examinar el condicionamiento de la matriz, que empleando la norma 2 *es $\kappa(A) \simeq 11.298$. Así, el condicionamiento del problema sólo justificaría una amplificación de los errores de redondeo para devolver errores finales de un* $0.1\,\%$ *(que son los que aparecen en la variable x_2).*

La verdadera razón del enorme error cometido en la resolución del sistema tiene que ver con el proceso de eliminación. En particular, obsérvese que (en la única etapa de eliminación existente dado que el sistema es de dos ecuaciones) el término en la diagonal (0.003*) es muy reducido lo que lleva, en las siguientes operaciones, a dividir por un número muy pequeño. Esta operación es siempre delicada cuando se calcula en aritmética finita ya que si, como en este caso, se opera sobre cantidades muy grandes para obtener finalmente mediante restas una cantidad que no lo es, se perderá mucha precisión en los cálculos (de forma grosera, el error de redondeo se multiplicará por el inverso de esa cantidad pequeña, lo que en este caso supone una amplificación de los errores generados por la aritmética finita y el condicionamiento desde un* $0.1\,\%$ *hasta órdenes de* $100\,\%$ *como, de hecho, ocurre en la práctica).*

Obsérvese que, desde la perspectiva de la aritmética finita, la situación es en cierto modo parecida a la que ocurría en el ejemplo 3.3 donde el proceso de eliminación no podía continuar debido a que aparecía un elemento nulo en la diagonal (en el presente ejemplo, se tiene que $\frac{a_{11}}{a_{12}} \sim 10^{-4}$, que es la precisión de la hipotética aritmética finita empleada). En suma, trabajando con aritmética finita, un valor muy reducido puede considerarse como un elemento nulo que hace imposible en la práctica proceder (con garantías) en la eliminación.

Esta última analogía sirve también para proponer el modo de resolver esta dificultad. Como se hace en la eliminación de Gauss con aritmética exacta al encontrar un cero en la diagonal, al trabajar con aritmética finita si aparece un término muy pequeño en la diagonal se

deberán intercambiar las ecuaciones para utilizar en la etapa de eliminación correspondiente divisiones por números que no sean pequeños.

En la práctica, en cada etapa de eliminación se busca la fila que contiene (en la columna que se está tratando de eliminar y por debajo de la diagonal) el elemento con valor absoluto más grande y se intercambia con la fila correspondiente. De este modo, se asegura que se divide por el número más grande posible y se reduce la amplificación de los errores de redondeo. Al método de eliminación de Gauss con esta estrategia se le denomina método de Gauss con pivote total.

Aplicado al presente ejemplo, esta estrategia reordenaría (en la única etapa de eliminación) las ecuaciones. Así, si se considera ahora el sistema con las ecuaciones reordenadas:

$$\left\{\begin{array}{lll} 5.291x_1 & -6.130x_2 & = & 46.78 \\ 0.003\,x_1 & +59.14\,x_2 & = & 59.17 \end{array}\right.$$

Etapa de eliminación gaussiana:

$$\left\{\begin{array}{lll} +5.291\times 10^0\,x_1 & -6.130\times 10^0\,x_2 & = & +4.678\times 10^1 \\ & +5.914\times 10^1\,x_2 & = & +5.914\times 10^1 \end{array}\right.$$

Remonte tras eliminación gaussiana:

$$\begin{array}{ll} x_2 & = +1.000\times 10^0 \\ x_1 & = +8.841\times 10^0 + 1.159\times 10^0 \;\; = +1.000\times 10^1 \end{array}$$

De este modo, la simple reordenación de las ecuaciones ha llevado a que la solución que devuelve el método sea, de nuevo, exacta en los dígitos que retiene la aritmética finita empleada.

□

Se acaba de ilustrar como, en determinadas ocasiones, la reordenación de las ecuaciones (durante el proceso de eliminación) atendiendo a la magnitud de los coeficientes encontrados en la correspondiente columna basta para evitar la amplificación de los errores de redondeo. Sin embargo, esta técnica (conocida como método de Gauss con pivote parcial) no siempre es suficiente para garantizar que no se produce la amplificación, tal y como se muestra en el siguiente ejemplo, que servirá al mismo tiempo para considerar una estrategia alternativa (al método de Gauss con pivote parcial) que resulte más robusta.

Ejemplo 3.6 *Se considera el sistema de ecuaciones*

$$\left\{\begin{array}{lll} 30\,x_1 & +591400\,x_2 & = & 591700 \\ 5.291x_1 & -6.130x_2 & = & 46.78 \end{array}\right.$$

con solución exacta $x_1 = 10$ y $x_2 = 1$ (se trata del mismo problema con un escalamiento diferente), mediante el método de Gauss con pivote parcial y la misma aritmética finita.

Etapa de eliminación gaussiana:

$$\left\{\begin{array}{lll} +3.000\times 10^1\, x_1 & +5.914\times 10^5\, x_2 & = +5.917\times 10^5 \\ & -1.043\times 10^5\, x_2 & = -1.044\times 10^5 \end{array}\right.$$

Remonte tras eliminación gaussiana:

$$\begin{array}{ll} x_2 & = +1.001\times 10^0 \\ x_1 & = +1.972\times 10^4 - 1.973\times 10^4 \quad = -1.000\times 10^1 \end{array}$$

Así, se ha originado de nuevo un error del 0.1 % *en la variable* x_2 *y un error del* 200 % *en la variable* x_1*. Obsérvese que no resulta raro que se haya obtenido de nuevo este resultado teniendo en cuenta que, con respecto al ejemplo original, sólo se ha reescalado la primera ecuación evitando así el intercambio de filas (o* pivote *en la terminología habitual) y manteniendo entonces las mismas operaciones.*

De acuerdo con la observación anterior, se comprueba que la estrategia de intercambio de filas no debería atender exclusivamente a comparar los valores absolutos de los coeficientes sobre la columna donde se está llevando a cabo la eliminación sino el cociente entre este valor absoluto y el mayor valor absoluto sobre esa misma fila (lo que se denomina el método de Gauss con pivote parcial y reescalado de filas).

Retomando el mismo ejemplo, para hacer el escalado de filas se elige dentro de cada fila el elemento con mayor valor absoluto; en este caso 591400 *para la primera fila y* 6.130 *en la segunda. A continuación se elige el mayor entre los cocientes* 30/591400 *y* 5.91/6.130 *y se toma esa fila como fila pivote:*

$$\left\{\begin{array}{lll} 5.291x_1 & -6.130x_2 & = 46.78 \\ 30\, x_1 & +591400\, x_2 & = 591700 \end{array}\right.$$

lo que lleva a un proceso de eliminación:

$$\left\{\begin{array}{lll} +5.291\times 10^0\, x_1 & -6.130\times 10^0\, x_2 & = +4.678\times 10^1 \\ & +5.914\times 10^5\, x_2 & = +5.914\times 10^5 \end{array}\right.$$

y de remonte:

$$\begin{array}{ll} x_2 & = +1.000\times 10^0 \\ x_1 & = +8.841\times 10^0 + 1.159\times 10^0 \quad = +1.000\times 10^1 \end{array}$$

□

Se comprueba entonces que la estrategia propuesta (método de Gauss con pivote parcial y reescalado de filas) proporciona una técnica más robusta para evitar la amplificación de los errores de redondeo.

Por último, se considera sobre el mismo ejemplo una alternativa al método de reescalado de filas (que simplemente facilita las comparaciones de los pivotes) y que resulta en un algoritmo denominado método de Gauss con equilibrado de filas.

Ejemplo 3.7 *Se considera el sistema de ecuaciones*

$$\left\{\begin{array}{lll} 30\,x_1 & +591400\,x_2 & = \ 591700 \\ 5.291x_1 & -6.130x_2 & = \ 46.78 \end{array}\right.$$

con solución exacta $x_1 = 10$ *y* $x_2 = 1$*, resuelto con la misma aritmética finita.*

Equilibrado de filas del sistema: se divide cada fila entre el elemento de mayor valor absoluto dentro de cada fila. En este caso se divide la primera fila entre 591400 *y la segunda entre* 6.130*:*

$$\left\{\begin{array}{lll} +5.073\times 10^{-5}\,x_1 & +1.000\times 10^{0}\,x_2 & = \ +1.001\times 10^{0} \\ +8.631\times 10^{-1}\,x_1 & -1.000\times 10^{0}\,x_2 & = \ +7.631\times 10^{0} \end{array}\right.$$

Reordenación y etapa de eliminación gaussiana:

$$\left\{\begin{array}{lll} +8.631\times 10^{-1}\,x_1 & -1.000\times 10^{0}\,x_2 & = \ +7.631\times 10^{0} \\ & +1.000\times 10^{0}\,x_2 & = \ +1.001\times 10^{0} \end{array}\right.$$

Remonte tras eliminación gaussiana:

$$\begin{array}{ll} x_2 & = +1.001\times 10^{0} \\ x_1 & = +8.841\times 10^{0} + 1.160\times 10^{0} \ = +1.000\times 10^{1} \end{array}$$

En la práctica, para evitar redondeos adicionales (obsérvese la aparición de un error del 0.1 *% en la variable* x_2*), el equilibrado de filas se hace dividiendo por una potencia de* 2 *adecuada. En la aritmética finita (decimal) considerada en este ejemplo se haría con potencias de* 10*, y al sistema*

$$\left\{\begin{array}{lll} 3.000\times 10^{1}\,x_1 & +5.914\times 10^{5}\,x_2 & = \ 5.917\times 10^{5} \\ 5.291\times 10^{0}\,x_1 & -6.130\times 10^{0}\,x_2 & = \ 4.678\times 10^{1} \end{array}\right.$$

le correspondería un sistema equilibrado

$$\left\{\begin{array}{lll} 3.000\times 10^{-4}\,x_1 & +5.914\times 10^{0}\,x_2 & = \ 5.917\times 10^{0} \\ 5.291\times 10^{0}\,x_1 & -6.130\times 10^{0}\,x_2 & = \ 4.678\times 10^{1} \end{array}\right.$$

que lleva (en la mencionada aritmética) a

$$x_1 = +1.000\times 10^{1} \qquad\qquad x_2 = +1.000\times 10^{0}$$

□

Factorización de Cholesky

Para terminar el estudio de los métodos directos se considera el caso particular de matrices simétricas y definidas positivas para las cuales el siguiente resultado devuelve la existencia de una factorización que preserva la simetría de la matriz (de modo que la matrices triangular inferior L y superior U son simétricas entre sí). Adicionalmente, como se mencionará más

tarde, esta factorización presenta unas buenas propiedades relativas a la no amplificación de los errores de redondeo en el correspondiente algoritmo de resolución de sistemas de ecuaciones lineales.

Proposición 3.4 *(Factorización de Cholesky)*
Sea A una matriz real, simétrica y definida positiva. Entonces A admite una factorización única $A = U^t U$ en donde U es una matriz triangular superior con diagonal positiva.

Demostración: Véase, por ejemplo, el texto de Kincaid y Cheney (páginas 136–138).

□

Cabe hacer aquí algunas observaciones relativas a la factorización de Cholesky:

- Resulta difícil probar que la matriz es definida positiva. De hecho, en la práctica se acude precisamente a la factorización de Cholesky para examinar si una matriz simétrica es o no definida positiva, puesto que la condición es necesaria y suficiente.
- El origen del sistema de ecuaciones lineales puede aclarar si se trata o no de una matriz definida positiva (por ejemplo, la discretización mediante elementos finitos de un problema elíptico autoadjunto sí asegura que la matriz de rigidez es simétrica y definida positiva).
- Se tiene un ahorro relativo de memoria sobre la factorización LU. Si la matriz A era simétrica, seguramente sólo se habría reservado memoria para la mitad de la matriz, de modo que se asegura así que no será necesario reservar importante memoria adicional para resolver el sistema de ecuaciones lineales.
- El método de Cholesky asegura estabilidad ante errores de redondeo sin pivoteo (es decir, los errores de redondeo no se amplificarán enormemente), lo que permite abordar la búsqueda directa de los elementos de U a partir de las ecuaciones

  $$A = U^t U$$

  pudiéndose consultar los detalles, por ejemplo, en el texto de Kincaid y Cheney.

3.4. Métodos iterativos clásicos

Aunque el método de eliminación de Gauss (con las modificaciones presentadas para evitar la amplificación de los errores de redondeo) ofrece un algoritmo eficaz para la resolución de cualquier sistema de ecuaciones lineales, tal y como se ha mencionado ya, el coste computacional del método (medido en número de productos que se deben calcular) crece con el cubo del número de ecuaciones (de forma más precisa, viene dado por $\frac{1}{3}N^3 + \mathcal{O}(N^2)$).

Adicionalmente, es habitual que los grandes sistemas de ecuaciones estén asociados a matrices *huecas*, que presentan un elevado número de ceros en su estructura de modo que

sólo una pequeña fracción de los elementos son no nulos (recuérdense los comentarios hechos en la primera sección de este tema). Los métodos directos presentan entonces un segundo problema asociado al efecto de *rellenado* de las matrices de factorización, de modo que es preciso reservar memoria adicional para guardar los elementos de la factorización, lo cual puede ser grave en sistemas ya de por sí grandes.

Se plantea entonces la necesidad de encontrar métodos alternativos para la resolución de sistemas de ecuaciones lineales cuyo coste computacional sea más reducido (de modo que permitan resolver grandes sistemas de ecuaciones lineales). Se tratará de métodos iterativos que buscan una solución aproximada del sistema de ecuaciones lineales. Así, para estos métodos se tendrá no sólo un error de redondeo (como ocurría en los métodos directos) sino también un error de truncamiento o aproximación, que en todo caso se espera que pueda ser controlado.

Obsérvese, de todos modos, que la introducción de un pequeño error de aproximación no tiene por qué ser grave. Así, si el sistema de ecuaciones lineales proviene de una discretización de un problema continuo (o, en general, se ha cometido un cierto error de aproximación para llegar al sistema de ecuaciones lineales) puede asumirse un error en la resolución del sistema. Por ejemplo, con un error de discretización espacial del orden de 10^{-4} no hay diferencia práctica entre encontrar la solución *exacta* con un método directo (que contendrá solamente errores del orden del redondeo, típicamente del orden de 10^{-16}) o encontrar una solución aproximada con errores del orden de 10^{-6}.

Se considera en primer lugar una familia muy simple de metodos iterativos denominados genéricamente métodos iterativos clásicos. Como se verá, es fácil detectar algunas graves deficiencias de estos métodos, que se refieren tanto a las restrictivas hipótesis de convergencia como a la lenta convergencia en muchos casos de interés. Obsérvese que una lenta convergencia de los métodos hace que éstos pierdan todo su interés ya que, en la práctica, resultará preferible emplear un método directo (al devolver un coste computacional total más reducido).

La deficiencias de estos métodos iterativos clásicos se corregirán en una segunda familia de métodos (denominados métodos de tipo gradiente). Así, en la práctica los métodos clásicos son poco empleados como métodos de resolución. Sin embargo, sí se emplean como métodos de precondicionamiento de los (frecuentes) sistemas de ecuaciones mal condicionados (recuérdense las observaciones hechas en su momento sobre las técnicas de precondicionamiento). Este uso de los métodos clásicos, junto con su utilidad a la hora de entender la formulación de otros métodos más complejos, hace que se expongan aquí.

Como ya se ha comentado, los métodos iterativos para resolver un sistema $A\vec{x} = b$ se caracterizan por construir, a partir de un vector arbitrario, una sucesión de vectores que converja a la solución del sistema. A continuación se expone sobre un ejemplo la formulación de dos métodos elementales: los métodos de Jacobi y Gauss-Seidel.

Ejemplo 3.8 *Se considera (de nuevo) el siguiente sistema de ecuaciones*

$$\left\{\begin{array}{llll} 4x_1 & -x_2 & +x_3 & = 8 \\ 2x_1 & +5x_2 & +2x_3 & = 3 \\ x_1 & +2x_2 & +4x_3 & = 11 \end{array}\right.$$

Puesto que contamos con tres ecuaciones para determinar tres incógnitas, es claro que podría emplearse cada una de las ecuaciones para resolver una incógnita si conociésemos el valor del resto de las incógnitas. Obviamente desconocemos ese valor del resto de las incógnitas, pero también es cierto que si contásemos con algún valor aproximado podríamos emplear éste.

Así, el método de Jacobi propone lo siguiente: supongamos que tenemos una aproximación (x_1^n, x_2^n, x_3^n) de la solución del sistema de ecuaciones. Entonces, hagamos lo siguiente:

(1) se emplea la primera ecuación del sistema para obtener una (nueva) aproximación de x_1 (que se denotará mediante x_1^{n+1}) empleando para x_2 y x_3 los valores aproximados de partida (x_2^n y x_3^n)

$$4x_1^{n+1} - x_2^n + x_3^n = 8$$

(2) se emplea la segunda ecuación del sistema para obtener una (nueva) aproximación de x_2 (que se denotará mediante x_2^{n+1}) empleando para x_1 y x_3 los valores aproximados de partida (x_1^n y x_3^n)

$$2x_1^n + 5x_2^{n+1} + 2x_3^n = 3$$

(3) se emplea la tercera ecuación del sistema para obtener una (nueva) aproximación de x_3 (que se denotará mediante x_3^{n+1}) empleando para x_1 y x_2 los valores aproximados de partida (x_1^n y x_2^n)

$$x_1^n + 2x_2^n + 4x_3^{n+1} = 11$$

Cabe esperar que la aproximación de la solución $(x_1^{n+1}, x_2^{n+1}, x_3^{n+1})$ sea mejor que la aproximación (x_1^n, x_2^n, x_3^n) y que la sucesión generada por la repetición de estas iteraciones converja entonces a la solución del sistema de ecuaciones lineales. A continuación se examinará la convergencia de este esquema pero, en todo caso, parece razonable pensar que la convergencia del esquema depende en cierto modo de la magnitud (del valor absoluto) de los coeficientes en la diagonal de la matriz. Así, considerando por ejemplo la primera ecuación para un sistema de tres ecuaciones general

$$a_{11}\, x_1^{n+1} + a_{12}\, x_2^n + a_{13}\, x_3^n = b_1$$

y suponiendo que x_2^n y x_3^n están cerca de la solución, parece claro que si a_{11} toma un valor (absoluto) muy reducido no importa que se cometa un error muy grande en la aproximación de x_1 ya que su efecto sobre la ecuación es muy pequeño. Al contrario, si a_{11} es grande (en términos absolutos) no podrán cometerse errores grandes en la aproximación de x_1 pues

en otro caso no se verificaría la ecuación (al aportar el término en x_1 una contribución apreciable a esta ecuación).

El razonamiento anterior constituye, en todo caso, un análisis heurístico de la convergencia que será confirmado más tarde.

El método de Jacobi admite inmediatamente una (supuesta) mejora. Revisando el algoritmo aplicado al ejemplo considerado, y suponiendo que el método converge (de modo que se espera que $(x_1^{n+1}, x_2^{n+1}, x_3^{n+1})$ sea mejor que la aproximación (x_1^n, x_2^n, x_3^n)), puede hacerse lo siguiente

- *en la etapa (2), podría sustituirse x_1^n por x_1^{n+1} que ya ha sido calculado en la etapa (1) y suponemos que constituye una mejor aproximación de x_1, de modo que los cálculos quedan de la forma*
$$2x_1^{n+1} + 5x_2^{n+1} + 2x_3^n = 3$$
donde sólo x_2^{n+1} es desconocido

- *en la etapa (3), podría sustituirse x_1^n por x_1^{n+1} y x_2^n por x_2^{n+1} que ya han sido calculados en la etapas (1) y (2), respectivamente, y suponemos que constituyen mejores aproximaciones de x_1 y x_2, de modo que los cálculos quedan de la forma*
$$x_1^{n+1} + 2x_2^{n+1} + 4x_3^{n+1} = 11$$
donde sólo x_3^{n+1} es desconocido

La modificación propuesta del método de Jacobi se denomina método de Gauss-Seidel y propone, para la resolución del sistema de ecuaciones planteado, un algoritmo iterativo donde en cada iteración parte de una aproximación (x_1^n, x_2^n, x_3^n) para construir una nueva aproximación $(x_1^{n+1}, x_2^{n+1}, x_3^{n+1})$ de la forma siguiente

(1) se emplea la primera ecuación del sistema para obtener una (nueva) aproximación de x_1 (que se denotará mediante x_1^{n+1}) empleando para x_2 y x_3 los valores aproximados de partida (x_2^n y x_3^n)
$$4x_1^{n+1} - x_2^n + x_3^n = 8$$

(2) se emplea la segunda ecuación del sistema para obtener una (nueva) aproximación de x_2 (que se denotará mediante x_2^{n+1}) empleando para x_1 el valor aproximado obtenido en la etapa anterior (x_1^{n+1}) y para x_3 el valor aproximado de partida (x_3^n)
$$2x_1^{n+1} + 5x_2^{n+1} + 2x_3^n = 3$$

(3) se emplea la tercera ecuación del sistema para obtener una (nueva) aproximación de x_3 (que se denotará mediante x_3^{n+1}) empleando para x_1 y x_2 los valores aproximados obtenidos en las etapas anteriores (x_1^{n+1} y x_2^{n+1})
$$x_1^{n+1} + 2x_2^{n+1} + 4x_3^{n+1} = 11$$

Parece razonable esperar que, en aquellos casos donde el método de Jacobi converja, el método de Gauss-Seidel que se acaba de describir produzca una aceleración de la convergencia.

□

Los métodos de Jacobi y Gauss-Seidel pueden ser descritos desde un punto de vista matricial como métodos basados en una descomposición de la matriz del sistema.

Así, dado un sistema de ecuaciones lineales $A\vec{x} = \vec{b}$, si escribimos A de la forma $A = M - N$ donde M es una matriz fácilmente invertible (lo cual no quiere decir que se vaya a calcular su inversa sino que el sistema de ecuaciones asociado a la matriz M es fácil de resolver), se tiene que el sistema de ecuaciones

$$A\,\vec{x} \,=\, \vec{b}$$

puede también escribirse de la forma

$$M\,\vec{x} \,=\, N\,\vec{x} \,+\, \vec{b}$$

La escritura anterior, donde M conduce a sistemas fáciles de resolver (en los métodos de Jacobi y Gauss-Seidel se tratará de matrices diagonales y triangulares, respectivamente), sugiere el siguiente método iterativo

`se toma` $\vec{x}^{\,0}$ `arbitrario`

`para` $n = 0, 1, 2, \ldots$

$\quad$ `se calcula` $\vec{x}^{\,n+1}$ `resolviendo` $M\,\vec{x}^{\,n+1} \,=\, N\,\vec{x}^{\,n} \,+\, \vec{b}$

debiendo añadir, tras cada etapa, algún examen de la convergencia para saber si se debe detener o no el algoritmo iterativo.

Formalmente, teniendo en cuenta que M debe ser fácilmente invertible (recuérdese lo que esto significa), será habitual escribir cada iteración del método de la forma

$$\vec{x}^{\,n+1} \,=\, M^{-1}\,N\,\vec{x}^{\,n} \,+\, M^{-1}\,\vec{b}$$

o, como se hará más tarde, de forma más general

$$\vec{x}^{\,n+1} \,=\, B\,\vec{x}^{\,n} \,+\, \vec{c}$$

Los métodos iterativos (para la resolución de sistemas de ecuaciones lineales) que permiten su escritura de la forma anterior reciben el nombre de métodos lineales o también métodos estacionarios.

Consideremos entonces la matriz $A = (a_{ij})$ en la forma $A = D - E - F$ donde

$$D = \begin{pmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & \ddots & \\ 0 & & & a_{NN} \end{pmatrix}, \quad E = \begin{pmatrix} 0 & & & 0 \\ -a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ -a_{N1} & \cdots & -a_{N\,N-1} & 0 \end{pmatrix}$$

y

$$F = \begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1N} \\ & \ddots & \ddots & \vdots \\ & & 0 & -a_{N-1\,N} \\ 0 & & & 0 \end{pmatrix}$$

y veamos cómo llevar el método de Jacobi y Gauss-Seidel a la forma general de un método lineal.

Si se elige para las matrices M y N

$$\begin{aligned} M &= D \\ N &= E + F \end{aligned}$$

se tiene el método de Jacobi

$$D\,\vec{x}^{\,n+1} = (E+F)\,\vec{x}^{\,n} + \vec{b}, \quad n = 0, 1, \ldots$$

o, equivalentemente (es claro que la matriz diagonal D es fácilmente invertible),

$$\vec{x}^{\,n+1} = D^{-1}\,((E+F)\,\vec{x}^{\,n} + \vec{b})), \quad n = 0, 1, \ldots$$

Si, por otro lado, se elige para las matrices M y N

$$\begin{aligned} M &= D - E \\ N &= F \end{aligned}$$

se obtiene el método de Gauss-Seidel, que se escribe como

$$(D-E)\,\vec{x}^{\,n+1} = F\,\vec{x}^{\,n} + \vec{b}, \quad n = 0, 1, \ldots$$

o, equivalentemente (también la matriz $(D-E)$, que es triangular inferior, resulta fácilmente invertible),

$$\vec{x}^{\,n+1} = (D-E)^{-1}\,(F\,\vec{x}^{\,n} + \vec{b}), \quad n = 0, 1, \ldots$$

Ya se ha adelantado, al presentar el método de Jacobi, que la magnitud (del valor absoluto) de los coeficientes en la diagonal de la matriz juegan un papel importante en la convergencia del método. Así, existe un método que propone una modificación del algoritmo

de Gauss-Seidel a fin de poder ajustar precisamente la magnitud de estos términos. Este método se conoce como método de relajación y procede del modo siguiente.

Se introduce un parámetro real $\omega \neq 0$ y se considera una descomposición de A en la forma $A = M_\omega - N_\omega$, donde

$$\begin{aligned} M_\omega &= \frac{1}{\omega} D - E \\ N_\omega &= \frac{1-\omega}{\omega} D + F \end{aligned}$$

Entonces, el método de relajación se escribe

$$\left(\frac{1}{\omega} D - E\right) \vec{x}^{n+1} = \left(\frac{1-\omega}{\omega} D + F\right) \vec{x}^n + \vec{b}, \quad n = 0, 1, \ldots$$

o, equivalentemente (la matriz $\left(\frac{1}{\omega} D - E\right)$, que es triangular inferior, resulta fácilmente invertible),

$$\vec{x}^{n+1} = \left(\frac{1}{\omega} D - E\right)^{-1} \left[\left(\frac{1-\omega}{\omega} D + F\right) \vec{x}^n + \vec{b}\right], \quad n = 0, 1, \ldots$$

Análisis de convergencia

Tal y como se ha comentado, se puede escribir cualquiera de los métodos iterativos anteriores como

$$\vec{x}^{n+1} = B \vec{x}^n + c$$

con

$$B = M^{-1} N, \quad c = M^{-1} \vec{b}$$

y se tiene el siguiente resultado:

Proposición 3.5 *(Convergencia de los métodos lineales)*
Si existe una norma matricial $\| \cdot \|_M$ tal que $\|B\|_M < 1$ entonces el método converge a la solución del sistema de ecuaciones lineales para cualquier vector inicial.

Demostración: Prácticamente inmediata pues $\vec{x}$ (solución del sistema de ecuaciones lineales) verifica

$$\vec{x} = B\vec{x} + \vec{c}$$

de modo que al restar de la expresión del iterante $\vec{x}^{n+1}$

$$\vec{x}^{n+1} = B\vec{x}^n + \vec{c}$$

se obtiene

$$\|\vec{e}^{n+1}\| = \|\vec{x}^{n+1} - \vec{x}\| = \|B(\vec{x}^n - \vec{x})\| \leq \|B\|_M \|\vec{x}^n - \vec{x}\| = \|B\|_M \|\vec{e}^n\|$$

y así

$$\|\vec{e}^{n+1}\| \leq (\|B\|_M)^{n+1} \|e_0\|$$

Es claro entonces que si $\|B\|_M < 1$ se tiene

$$\lim_{n \to \infty} \|\vec{e}^{n+1}\| = 0$$

□

Veamos ahora una aplicación de este resultado al método de Jacobi, que confirma la observación que se hizo al presentar este método.

Proposición 3.6 *(Convergencia del método de Jacobi)*
Si la matriz A es de diagonal estrictamente dominante, es decir,

$$|a_{ii}| > \sum_{j=1,\, j\neq i}^{N} |a_{ij}|$$

entonces el método de Jacobi converge.

Demostración: Puesto que

$$I - B = I - M^{-1}N = M^{-1}M - M^{-1}N = M^{-1}(M - N) = M^{-1}A$$

se tiene que

$$\|B\|_\infty = \|I - M^{-1}A\|_\infty = \|I - D^{-1}A\|_\infty = \max_{1\leq i\leq N} \sum_{j=1,\, j\neq i}^{N} |\frac{a_{ij}}{a_{ii}}| < 1$$

□

En todo caso, el resultado general expuesto en la proposición 3.5 resulta poco satisfactorio pues obligaría, para estudiar la convergencia de un cierto esquema, a examinar todas las normas matriciales hasta comprobar si alguna de ellas es menor que la unidad. A continuación se expone un resultado que aclara algo dicha búsqueda.

Antes de exponer dicho resultado, se definirá el radio espectral de una matriz.

Definición 3.1 *Sea A una matriz (real o compleja) de tamaño $N \times N$. Supongamos que sus autovalores son λ_1, λ_2, λ_3, ..., λ_N. Entonces se define el radio espectral de la matriz A (que se denotará mediante $\rho(A)$) como el mayor módulo de los autovalores, esto es*

$$\rho(A) = \max_{1\leq i\leq N} |\lambda_i|$$

Con este definición se tiene ahora el siguiente resultado:

Proposición 3.7 *Se verifica que*

$$\rho(A) = \inf_{\|\cdot\|_M} \|A\|_M$$

Demostración: Puede encontrarse, por ejemplo, en el libro de Kincaid y Cheney (página 192). En todo caso, obsérvese que es inmediato comprobar que $\rho(A)$ constituye una cota inferior para cualquier norma subordinada pues (tomando como j aquel índice para el cual $\rho(A) = |\lambda_j|$)

$$\rho(A) = \frac{\|A\vec{v}_j\|}{\|\vec{v}_j\|} \leq \|A\|_M$$

de modo que se trata de demostrar

$$\forall \epsilon > 0 \;\; \exists \, \|\cdot\|_{M_\epsilon} \text{ tal que } \|A\|_{M_\epsilon} \leq \rho(A) + \epsilon$$

□

Como aplicación del resultado anterior se tiene de forma inmediata que un método iterativo lineal

$$\vec{x}^{n+1} = B\vec{x}^n + \vec{c}$$

converge si y sólo si $\rho(B) < 1$. El hecho de que sea una condición necesaria se basa en que si no se cumpliese y, por lo tanto existiese un autovalor con módulo unitario o mayor que la unidad, si el algoritmo arranca con un error $\vec{e}^0 = \vec{x}^0 - \vec{x}$ (donde $\vec{x}$ representa la solución exacta del sistema) éste se mantendría o aumentaría impidiendo la convergencia.

En relación con el resultado anterior cabe hacer varias observaciones:

- La velocidad de convergencia depende de $\rho(B)$, ya que este valor actúa como el factor de reducción del error en cada paso.
- $\rho(B)$ es muy costoso de obtener, ya que obliga a calcular todos los autovalores de la matriz B (en realidad, se puede calcular solamente el de mayor módulo, tal y como se verá en el siguiente tema) por lo que en la práctica resulta poco útil como criterio para examinar la convergencia de un determinado esquema (el coste de evaluar esta condición sería más elevado que el de probar directamente si converge o no).
- En todo caso, para cierto tipo de matrices A con algunas características se tiene alguna información sobre $\rho(B)$ para la matriz B asociada a métodos iterativos clásicos, lo que permite estudiar la convergencia de estos métodos a la hora de resolver algunos sistemas de ecuaciones lineales. Pueden consultarse, por ejemplo, algunos resultados de convergencia para sistemas tridiagonales en el texto de Ciarlet citado al final de este tema.

Con carácter general, es cierto que los métodos iterativos clásicos son poco eficientes pues en la mayor parte de los casos el número de iteraciones para calcular la solución del sistema de ecuaciones lienales con una precisión razonable es aproximadamente del orden de magnitud del tamaño del sistema y, por lo tanto, no presentan ninguna ventaja sobre el método de Gauss. En consecuencia (salvo para matrices muy especiales) no son utilizados como métodos para resolver sistemas de ecuaciones lineales. No obstante, estos métodos se utilizan con cierta frecuencia en la construcción de precondicionadores (así como *suavizadores* en una clase especial de métodos iterativos que son los denominados métodos multimalla).

3.5. Métodos de tipo gradiente

Como se ha visto, al resolver grandes sistemas de ecuaciones, los métodos directos presentan graves deficiencias asociadas al número de operaciones necesarias (que originan costes computacionales muy altos) así como problemas de almacenamiento derivados del efecto de *rellenado*.

Por su parte, tampoco los métodos iterativos clásicos proporcionan una técnica eficiente de resolución de estos grandes sistemas de ecuaciones lineales puesto que su velocidad de convergencia es, con carácter general, baja.

Se plantea entonces el uso de métodos alternativos que, como se va a ver, están basados en la minimización del residuo (definido como $r = A\vec{x} - \vec{b}$, tal y como se vió previamente) que tratarán de reducir éste lo mas rápidamente posible, garantizando así una velocidad de convergencia aceptable. La idea de estos métodos es transformar la resolución de un sistema $A\,\vec{x} = \vec{b}$ con A una matriz simétrica y definida positiva en un problema equivalente de minimización de una forma cuadrática. Más adelante se considerará la extensión al caso general, donde A representará una matriz cualquiera.

Se comenzará por establecer la mencionada propiedad de equivalencia (de la resolución del sistema de ecuaciones lineales) con la minimización de un cierto funcional.

Proposición 3.8 *Dado el sistema $A\,\vec{x} = \vec{b}$ con A matriz simétrica y definida positiva se construye*

$$J(\vec{x}) = \frac{1}{2}\vec{x}^t\,A\,\vec{x} - \vec{x}^t\,\vec{b}.$$

Entonces, resolver $A\,\vec{x} = \vec{b}$ es equivalente a calcular el mínimo de J.

Demostración: En efecto:

a) Si $\vec{x}$ es un mínimo local de J entonces $\vec{\nabla}J(\vec{x}) = 0$, es decir, $\vec{\nabla}J(\vec{x}) = A\,\vec{x} - \vec{b} = 0$ y, por tanto, $A\,\vec{x} = \vec{b}$.

b) Si $\vec{x}$ es tal que $A\,\vec{x} = \vec{b}$ entonces

$$\vec{\nabla}J(\vec{x}) = A\,\vec{x} - \vec{b} = 0, \quad \text{y} \quad \hat{\vec{x}}^t\,HJ(\vec{x})\,\hat{\vec{x}} = \hat{\vec{x}}^t\,A\,\hat{\vec{x}} > 0, \ \ \forall \hat{\vec{x}} \in \mathbf{R^N}$$

y, por tanto, $\vec{x}$ es un mínimo de J.

□

La idea general de los métodos de tipo gradiente consiste en construir una sucesión de la siguiente manera: dada $\vec{x}^n$, aproximación de la solución $\vec{x}$, se calcula $\vec{x}^{n+1}$ mediante

a) se elige una cierta dirección de descenso de J, $\vec{d}^n$

b) en la dirección $\vec{d}^{\,n}$ se elige un paso óptimo (en algún cierto sentido) τ^n y se toma

$$\vec{x}^{\,n+1} = \vec{x}^{\,n} + \tau^n \vec{d}^{\,n}$$

Obsérvese que, dado $\vec{x} \in \mathbf{R^N}$, d es la dirección de descenso en $\vec{x}$ si

$$\exists \tau^0 \in \mathbf{R}^+ \ / \ J(\vec{x} + \tau d) < J(\vec{x}), \quad 0 < \tau \leq \tau_0$$

Las diferentes elecciones de la dirección de descenso dan lugar a distintos métodos.

3.5.1. Método del máximo descenso (o método de gradiente)

El denominado método del máximo descenso o método de gradiente propone en particular que, en cada paso, se asegure la mayor reducción posible del funcional J (en ese paso) para lo cual

- se tomará como dirección de descenso aquella marcada por $-\vec{\nabla} J$ en ese punto
- se tomará como paso áquel paso que de lugar a la mayor reducción de J en esa dirección

Obsérvese que, dado el carácter *no lineal* o *no estacionario* del método (asociado a que los gradientes de J cambian en cada punto) asegurar que cada paso conduce a la mayor reducción de J no asegura que, globalmente, el método devuelva la convergencia más rápida posible. Este punto quedará más claro posteriormente.

En primer lugar, para la dirección de máximo descenso, es claro que con

$$J(\vec{x}) = \frac{1}{2}\vec{x}^t A \vec{x} - \vec{x}^t \vec{b}$$

o, de forma más explícita,

$$J(x_1, x_2, \ldots, x_N) = \frac{1}{2} \sum_{i=1}^{N} \left(\sum_{j=1}^{N} a_{ij} x_j \right) x_i - \sum_{i=1}^{N} x_i b_i$$

se tendrá

$$\frac{\partial J}{\partial x_k}(x_1, x_2 \ldots x_N) = \sum_{i=1}^{N} a_{ki} x_i - b_k$$

Así, se tiene que

$$\vec{d}^{\,n} = -\vec{\nabla} J(\vec{x}^{\,n}) = -A\vec{x}^{\,n} + \vec{b}$$

de modo que el residuo $\vec{r}^{\,n} = A\,\vec{x}^{\,n} - \vec{b}$ marca precisamente la dirección de máximo descenso

Obtención de paso óptimo τ^n y formulación del método.

Es preciso ahora calcular τ^n para hacer $J(\vec{x}^{\,n} + \tau\,\vec{d}^{\,n})$ mínimo en la dirección dada por el residuo.

La siguiente propiedad devuelve, para una dirección prefijada, el cálculo del paso óptimo (determinado como aquel paso que hace mínimo el funcional J en la dirección dada).

Obsérvese que se trata entonces de un resultado genérico, que servirá más tarde para otros esquemas de tipo gradiente.

Proposición 3.9 *Sea A una matriz simétrica y definida positiva y sea $J(\vec{x}) = \frac{1}{2}\vec{x}^t A\vec{x} - \vec{x}^t\vec{b}$. Supongamos dado un cierto $\vec{x}^n$ y una cierta dirección $\vec{d}^n$. Sea finalmente τ^n tal que*

$$J(\vec{x}^n + \tau^n \vec{d}^n) \leq J(\vec{x}^n + \tau \vec{d}^n) \quad \forall \tau \in \mathbf{R}$$

Entonces

$$\tau^n = \frac{(\vec{d}^n)^t \vec{d}^n}{(\vec{d}^n)^t A \vec{d}^n}$$

Demostración: Se tiene que

$$J(\vec{x}^n + \tau \vec{d}^n) = \frac{1}{2}(\vec{x}^n + \tau \vec{d}^n)^t A (\vec{x}^n + \tau \vec{d}^n) - (\vec{x}^n + \tau \vec{d}^n)^t \vec{b} = \frac{1}{2}(\vec{x}^n)^t A \vec{x}^n$$
$$+\frac{1}{2}\tau(\vec{x}^n)^t A \vec{d}^n + \frac{1}{2}\tau(\vec{d}^n)^t A \vec{x}^n + \frac{1}{2}\tau^2(\vec{d}^n)^t A \vec{d}^n - (\vec{x}^n)^t \vec{b} - \tau(\vec{d}^n)^t \vec{b}$$

y, por tanto,

$$\frac{d\, J(\vec{x}^n + \tau \vec{d}^n)}{d\,\tau} = \frac{1}{2}(\vec{x}^n)^t A \vec{d}^n + \frac{1}{2}(\vec{d}^n)^t A \vec{x}^n + \tau\,(\vec{d}^n)^t A \vec{d}^n - (\vec{d}^n)^t \vec{b}.$$

Entonces

$$\frac{d\, J(\vec{x}^n + \tau \vec{d}^n)}{d\,\tau}(\tau^n) = 0 \Longleftrightarrow \tau^n = -\frac{\frac{1}{2}(\vec{x}^n)^t A \vec{d}^n + \frac{1}{2}(\vec{d}^n)^t A \vec{x}^n - (\vec{d}^n)^t \vec{b}}{(\vec{d}^n)^t A \vec{d}^n}$$

Puesto que la matriz A es simétrica, se tiene que

$$(\vec{x}^n)^t A \vec{d}^n = ((\vec{x}^n)^t A \vec{d}^n)^t = (\vec{d}^n)^t A \vec{x}^n$$

y, por tanto,

$$\tau^n = -\frac{(\vec{d}^n)^t A \vec{x}^n - (\vec{d}^n)^t \vec{b}}{(\vec{d}^n)^t A \vec{d}^n} = -\frac{(\vec{d}^n)^t (A \vec{x}^n - \vec{b})}{(\vec{d}^n)^t A \vec{d}^n} = \frac{(\vec{d}^n)^t \vec{d}^n}{(\vec{d}^n)^t A \vec{d}^n}\,.$$

□

Entonces, el algoritmo del método de máximo descenso queda del modo siguiente:

`se toma iterante inicial` $\vec{x}^0$

`para` $n = 0, 1, 2, \ldots$

- `se calcula dirección de descenso`

$$\vec{d}^n = -A\,\vec{x}^n + \vec{b}$$

- `se calcula paso`

$$\tau^n = \frac{(\vec{d}^{\,n})^t \vec{d}^{\,n}}{(\vec{d}^{\,n})^t \, A \, \vec{d}^{\,n}}$$

- `se actualiza la aproximación`

$$\vec{x}^{\,n+1} = \vec{x}^{\,n} + \tau^n \, \vec{d}^{\,n}$$

Al algoritmo descrito se le añadirá algún criterio de parada para detectar la convergencia de la aproximación. Parece desde luego razonable manejar la norma del residuo $\vec{r}^{\,n}$ (que coincide con la dirección de descenso), aunque deben recordarse las observaciones hechas en el apartado dedicado al condicionamiento de los sistemas de ecuaciones lineales (en un caso muy mal condicionado, el residuo podría ser muy pequeño sin que lo sea el error).

A priori, parece esperable que el método de máximo descenso proporcione la convergencia mas rápida posible ya que este método propone reducir lo más posible el funcional J en cada paso. Sin embargo, se va a presentar a continuación un ejemplo que pone de manifiesto las dificultades del método.

Ejemplo 3.9 *Se considera la resolución, mediante el método de máximo descenso, del siguiente sistema de ecuaciones elemental*

$$\begin{array}{rcrcl} a_{11}\, x_1 & + & 0\, x_2 & = & 0 \\ 0\, x_1 & + & a_{22}\, x_2 & = & 0 \end{array}$$

con $a_{11} > 0$ y $a_{22} > 0$. Es claro que se trata de un sistema cuya única solución es $x_1 = x_2 = 0$. Desde luego, se trata de un sistema especialmente simple, pero en todo caso cualquier sistema de dos ecuaciones lineales asociado a una matriz simétrica y definida positiva puede ser llevado a esta forma mediante un cambio de coordenadas (que incluya un descentrado que lleve el origen a la única solución del sistema y un giro asociado a los autovectores).

Consideremos ahora la resolución de este sistema de ecuaciones mediante el método de máximo descenso. Obsérvese que resulta sencillo prever la sucesión generada por el método si se representan las curvas de nivel del funcional J (obsérvese que en este ejemplo $J(\vec{x}) = \frac{1}{2}a_{11}x_1^2 + \frac{1}{2}a_{22}x_2^2$). En efecto, puesto que el método de máximo descenso propone como dirección aquella fijada por el gradiente de J, se buscará un punto sobre la recta normal a la curva de nivel en ese punto. Puesto que además se elige el paso de modo que el descenso sea máximo en esa dirección la búsqueda sobre esa recta llegará hasta encontrar un punto de tangencia a una curva de nivel.

Comenzaremos por un caso especialmente fácil, donde $a_{11} = a_{22}$. En este caso (obsérvese la figura 3.1, que corresponde al caso $a_{11} = a_{22} = 1$, con $\vec{x}^{\,0} = (\frac{5}{2}, \frac{1}{2})$), resulta inmediato ver que el método necesita una sola iteración para calcular de forma exacta la solución del sistema de ecuaciones. La razón está en que la dirección de máximo descenso (normal a las curvas de nivel, que son ahora circunferencias) para cualquier iterante inicial pasa por el origen y, fijada esa dirección, el punto donde J toma el valor más reducido es claramente el origen.

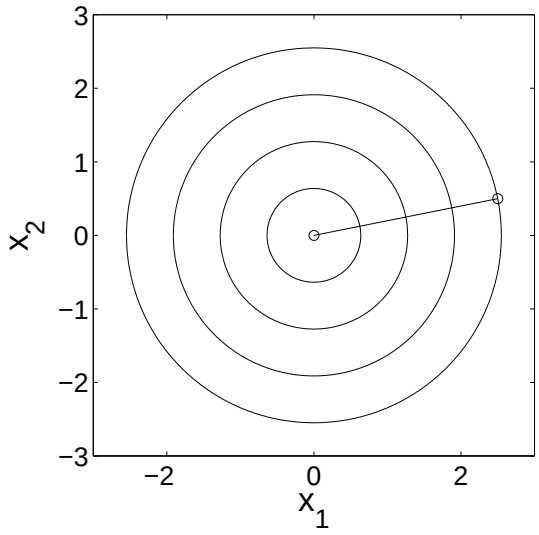

Figura 3.1: Iteraciones del método de máximo descenso para el caso $a_{11} = a_{22}$

Sea ahora el caso $a_{11} \neq a_{22}$. Ahora (obsérvese la figura 3.2, que corresponde al caso $a_{11} = \frac{1}{9}$ y $a_{22} = 1$, con $\vec{x}^0 = (\frac{5}{2}, \frac{1}{2})$) la sucesión generada por el método de máximo descenso recorre un camino en zig-zag *para acercarse al origen (debido a que las direcciones de descenso verifican $(\vec{d}^k)^t \vec{d}^{k-1} = 0$), por lo que la convergencia puede ser relativamente lenta. En particular, la convergencia se deteriorará enormemente cuando los valores de a_{11} y a_{22} sean muy distintos.*

Convergencia del método de máximo descenso

A continuación se enuncia un resultado de convergencia del método de máximo descenso, que estudia el error en la norma $\|\cdot\|_A$ asociada a la matriz A (que suponemos simétrica y definida positiva)

$$\|\vec{x}\|_A = \sqrt{\vec{x}^t A \vec{x}}$$

Obsérvese que, de hecho, se trata de la norma inducida por un producto interior asociado la matriz simétrica y definida positiva A

$$(\vec{x}, \vec{y})_A = \vec{x}^t A \vec{y}$$

Proposición 3.10 *Sea A simétrica y definida positiva, con número de condición κ. Entonces, la sucesión $\{\vec{x}^n\}$ generada por el método de máximo descenso verifica*

$$\|\vec{x} - \vec{x}^n\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^n \|\vec{x} - \vec{x}^0\|_A \,.$$

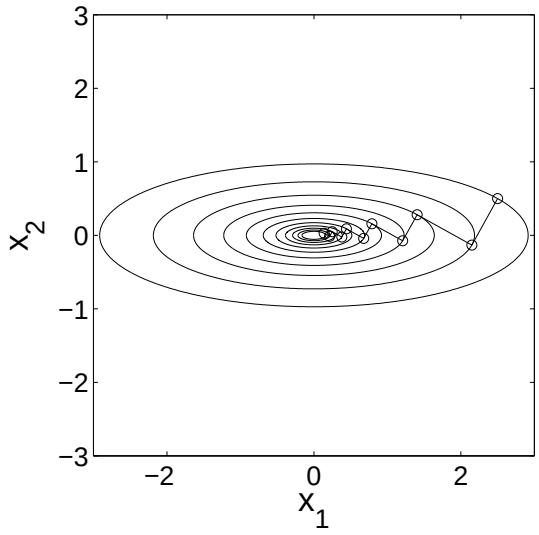

Figura 3.2: Iteraciones del método de máximo descenso para el caso $a_{11} \neq a_{22}$

Demostración: Puede consultarse en el texto de Quarteroni, Sacco y Saleri (página 148).

□

Recuérdese que el número de condición de A es

$$\kappa = \|A\|_M \, \|A^{-1}\|_M$$

y obsérvese que para el ejemplo elemental anterior el número de condicionamiento viene dado por

$$\kappa = \text{máx}\left(\frac{a_{11}}{a_{22}}, \frac{a_{22}}{a_{11}}\right)$$

Así, el resultado de convergencia efectivamente predice para el ejemplo considerado que la convergencia puede ser muy lenta cuando a_{11} y a_{22} son muy distintos.

En la práctica, los sistemas mal condicionados son bastante frecuentes, por lo que el uso del método de máximo descenso no es muy aconsejable. En su lugar, sería necesario contar con un método que evite la ralentización de la convergencia para dichos sistemas mediante una elección más adecuada de las direcciones de descenso. Éste es precisamente el objeto de la siguiente sección.

3.5.2. Método de gradiente conjugado

Retomando el ejemplo elemental anterior, consideramos un enfoque distinto para la obtención de las direcciones de descenso.

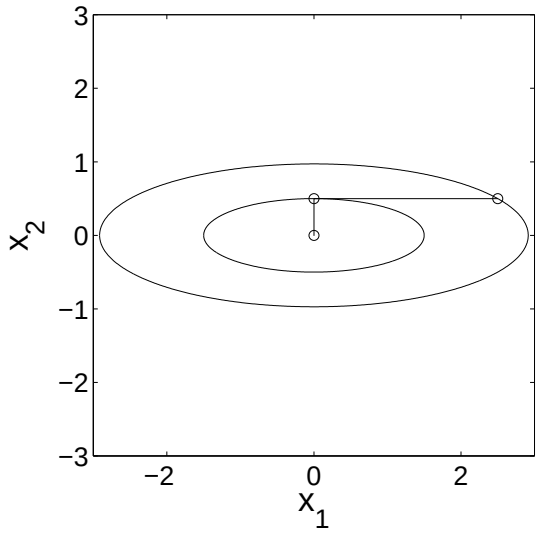

Figura 3.3: Búsqueda según direcciones $\vec{e}_1$ y $\vec{e}_2$

Ejemplo 3.10 *Se toma de nuevo el sistema elemental*

$$\begin{array}{rcrcl} a_{11}\, x_1 & + & 0\, x_2 & = & 0 \\ 0\, x_1 & + & a_{22}\, x_2 & = & 0 \end{array}$$

para el caso $a_{11} = \frac{1}{9}$ *y* $a_{22} = 1$*, con* $\vec{x}^0 = (\frac{5}{2}, \frac{1}{2})$*). Como se ha visto el método de máximo descenso genera una sucesión de aproximaciones en* zig-zag *con una lenta convergencia a la solución.*

Alternativamente, podría haberse planteado la búsqueda de la solución según otras direcciones del siguiente modo:

- *en el primer paso, se busca el mínimo de* J *según la dirección* $\vec{e}_1$
- *en el segundo paso, se busca el mínimo de* J *según la dirección* $\vec{e}_2$

Resulta inmediato comprobar entonces que el segundo paso conduce ya a la solución exacta del problema (véase la figura 3.3).

En el ejemplo anterior, la elección de las direcciones es especialmente afortunada (lo cual tiene que ver con que se trata de autovectores de la matriz A) pero, en todo caso, es posible tratar de generalizar la elección de unas buenas direcciones para la búsqueda de las soluciones del sistema de ecuaciones lineales.

En primer lugar, se define una cierta base de $\mathbf{R}^N$ de la forma siguiente

Definición 3.2 *Se dice que* $\{\vec{u}^j\}_{j=1}^N$ *es una base A-ortonormal de* $\mathbf{R}^N$ *si constituye una base de* $\mathbf{R}^N$ *y verifica*

$$(\vec{u}^i)^t A \vec{u}^i = \delta_{ij} \qquad \forall i, j \in \{1, 2, \ldots, N\}$$

La siguiente propiedad asegura que una base A-ortonormal constituye una buena elección de las direcciones de descenso ya que permite calcular la solución exacta del sistema de ecuaciones lineales tras N etapas de un método de descenso.

Proposición 3.11 *Sea A una matriz de tamaño $N \times N$ y $\{\vec{u}^j\}_{j=1}^N$ una base A-ortonormal de $\mathbf{R}^N$. Dado $\vec{x}^0 \in \mathbf{R}^N$ arbitrario, se genera la sucesión*

$$\vec{x}^i = \vec{x}^{i-1} - (\vec{u}^{i-1})^t(A\vec{x}^{i-1} - \vec{b})\vec{u}^{i-1} \qquad i = 1, 2, \ldots, N$$

Entonces $\vec{x}^N = \vec{x}$.

Demostración: Puede consultarse en el texto de Kincaid y Cheney (página 213).

□

Cabe hacer aquí dos observaciones:

- El resultado anterior asegura que es posible calcular de este modo la solución exacta del sistema de ecuaciones lineales en un número finito de pasos (igual al número de ecuaciones). Así, dicho resultado puede considerarse la base de un método directo de resolución del sistema. No obstante, en la práctica no resulta satisfactorio ya que el coste de cálculo de dicho método sería del mismo orden que los métodos directos ya estudiados (en particular, del método de Cholesky, que resulta el método adecuado para matrices simétricas y definidas positivas). De este modo, se espera contar con una rápida convergencia del método que permita detener los cálculos después de un número no muy grande de iteraciones (desde luego significativamente menor que el número de ecuaciones) y considerar entonces el método como un método iterativo que devuelve una solución aproximada.

- Para contar con un método eficiente, el coste de la base A-ortonormal debe ser reducido. Como se verá, el método de gradiente conjugado propone una forma recursiva de calcular una base, de modo que no debe ser almacenada, evitando así las dificultades de reserva de memoria que se originarían al resolver grandes sistemas (en realidad, la base que se construye es A-ortogonal).

Algoritmo del método de gradiente conjugado

El algoritmo del método de gradiente conjugado (que emplea una base A-ortogonal para las direcciones de descenso), con un criterio de parada sobre los residuos se escribe del modo siguiente

`Dado` $\vec{x}^0$ `arbitrario, se construyen`

`residuo inicial:` $\vec{r}^0 = A\vec{x}^0 - \vec{b}$

`dirección inicial de descenso:` $\vec{v}^0 = -\vec{r}^0$

`Para` $n = 1, 2, 3 \cdots$

`cálculo del paso:` $\tau_n = \|\vec{r}^n\|^2/((\vec{v}^n)^t A\vec{v}^n)$

`cálculo de nuevo iterante:` $\vec{x}^{(n+1} = \vec{x}^n + \tau_n \vec{v}^n$

`cálculo de nuevo residuo:` $\vec{r}^{n+1} = \vec{r}^n + \tau_n A\vec{v}^n$

`si` $\|r^n\| <$ `tol`$\|\vec{b}\|$ `terminar`

`cálculo de parámetro` s_n`:` $s_n = \|\vec{r}^{n+1}\|^2/\|\vec{r}^n\|^2$

`cálculo de nueva dirección de descenso:` $\vec{v}^{n+1} = -\vec{r}^{n+1} + s_n\vec{v}^n$

`Terminar en n`

Nota 3.1 *Reordenando los cálculos del método en cada etapa, se debe hacer*

(a) `calcular` $\vec{z}^n = A\vec{v}^n$

(b) `calcular` $d_n = (\vec{v}^n)^t\vec{z}^n$

(c) `calcular` $\tau_n = c_n/d_n$

(d) `calcular` $\vec{x}^{n+1} = \vec{x}^n + \tau_n\vec{v}^n$

(e) `calcular` $\vec{r}^{n+1} = \vec{r}^n - \tau_n\vec{z}^n$

(f) `calcular` $c^{n+1} = (\vec{r}^{n+1})^t\vec{r}^n$

(g) `calcular` $s_n = c^{n+1}/c^n$

(h) `calcular` $\vec{v}^{n+1} = \vec{r}^{n+1} + s_n\vec{v}^n$

de modo que se requiere (al margen de las operaciones sobre escalares)

- 1 *producto matriz vector (cálculo en (a))*
- 2 *productos escalares (cálculos en (b) y (g))*
- 3 *sumas de vectores (cálculos en (d), (e) y (h))*

El coste por paso se consume prácticamente en el producto de la matriz A por el vector $\vec{v}^n$*. Así, es conveniente que dicho producto (y, aunque en menor medida, los productos escalares y suma de vectores) se haga del modo lo más eficiente posible. En la práctica, los códigos de resolución de sistemas de ecuaciones lineales basados en el método de gradiente conjugado (y también los códigos basados en otros métodos iterativos) emplean bibliotecas de subrutinas de bajo nivel adaptadas a cada procesador (como BLAS, por ejemplo) para llevar a cabo estos productos.*

Convergencia del método de gradiente conjugado

Para el método de gradiente conjugado se cuenta con el siguiente resultado de convergencia

Proposición 3.12 *Sea A de tamaño $N \times N$, simétrica y definida positiva y con número de condición κ. Entonces, la sucesión $\{\vec{x}^n\}$ generada por el método de gradiente conjugado verifica*

$$\|\vec{x} - \vec{x}^n\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n \|\vec{x} - \vec{x}^0\|_A .$$

En todo caso, el método devuelve la solución exacta en, como mucho, N iteraciones.

Demostración: Puede consultarse en el texto de Quarteroni, Sacco y Saleri (página 154), que en realidad hace una acotación ligeramente más fina del error (aunque cualitativamente devuelve la misma información).

□

Obsérvese que con $\kappa >> 1$ se tiene

$$\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \simeq 1 - \frac{2}{\sqrt{\kappa}} + \mathcal{O}(\frac{1}{\kappa})$$

de modo que, en un sistema mal condicionado, sólo se garantiza un factor de reducción del error por paso de aproximadamente $1 - \frac{2}{\sqrt{\kappa}}$, lo que devuelve un valor muy cercano a la unidad (y, en consecuencia, sólo puede asegurarse una reducción muy lenta del error).

En la práctica efectivamente ocurre que si el sistema de ecuaciones lineales está mal condicionado, la convergencia es lenta y se hace conveniente precondicionarlo de algún modo. En todo caso, debe prestarse atención a la necesidad de preservar la simetría de la matriz original así como su carácter de definida positiva.

Así, dado un sistema de ecuaciones lineales

$$A\vec{x} = \vec{b}$$

mal condicionado (con matriz A simétrica y definida positiva), se plantearía la resolución de un sistema modificado a través de una matriz P^{-1} de precondicionamiento

$$P^{-1}A\vec{x} = P^{-1}\vec{b}$$

Sin embargo, $P^{-1}A$ podría no ser simétrica y definida positiva (incluso si P se ellige simétrica y definida positiva). No obstante, si se define $\vec{y} = P\vec{x}$ se puede resolver el sistema de ecuaciones lineales

$$P^{-1}AP^{-1}\vec{y} = P^{-1}\vec{b}$$

que sí corresponde a un sistema con matriz simétrica y definida positiva y podría ser resuelto con el método de gradiente conjugado (y logrando una velocidad de convergencia adecuada si la matriz $P^{-1}AP^{-1}$ está bien condicionada). Es claro que a fin de lograr un problema bien condicionado, la matriz P debe elegirse lo más parecida posible a la matriz $A^{1/2}$.

Cabe, por último observar, que en la práctica es preciso reordenar el algoritmo para evitar construir explícitamente la matriz $P^{-1}AP^{-1}$.

Transformación de sistemas con matriz cualquiera

Dado un sistema de ecuaciones $A\,\vec{x} = \vec{b}$ donde A no es simétrica y definida positiva (pero sí regular), puede transformarse el sistema en

$$A^t\,A\,\vec{x} = A^t\,\vec{b}$$

donde la nueva matriz del sistema $A^t\,A$ sí es simétrica y definida positiva:

- $(A^t\,A)^t = A^t\,A$
- $(\vec{v})^t\,(A^t\,A)\,\vec{v} = (\vec{v})^t\,A^t\,A\,\vec{v} = (A\,\vec{v})^t\,A\,\vec{v} = \|A\,\vec{v}\|^2 > 0$ si $\vec{v} \neq \vec{0}$ ya que $A\vec{v} \neq \vec{0}$ si $v \neq \vec{0}$ al ser A regular.

En cualquier caso, deben hacerse algunas observaciones sobre la resolución de sistemas de ecuaciones con matrices que no son simétricas y definidas positivas:

- La generación de la nueva matriz para el sistema $A^t\,A$ puede originar graves problemas de *rellenado* de la matriz (pues la nueva matriz $A^t\,A$ puede tener muchos más elementos no nulos que A), de modo que se complique el almacenamiento en grandes sistemas de ecuaciones asociados a matrices huecas (que son precisamente los sistemas para los cuales se piensa utilizar este tipo de métodos).

- En la práctica no se construye la matriz $A^t\,A$ (tanto para evitar el coste asociado como para reducir los problemas de *rellenado* que se acaban de mencionar) por lo que es precio reescribir el algoritmo del método de gradiente conjugado para abordar este caso.

- En todo caso, cuando se trata de resolver grandes sistemas de ecuaciones lineales con matrices que no son simétricas y definidas positivas, existen métodos numéricos iterativos para la resolución eficiente de dichos sistemas. En particular, el método GMRES (*Generalized Minimal Residual Method*) constituye,e n cierta medida, una adaptación de las ideas del método de gradiente conjugado a la resolución de sistemas de ecuaciones lineales generales. Pueden consultarse algunos de estos métodos en la obra de G. Dahlquist y A. Björck citada al final del tema.

3.6. Métodos numéricos para sistemas de ecuaciones no lineales

Consideramos ahora el problema de resolver un sistema de N ecuaciones generales (para encontrar N incógnitas: $x_1,\ x_2\ \ldots x_N$)

$$F_1(x_1, x_2, x_3, \ldots x_N) = 0$$
$$F_2(x_1, x_2, x_3, \ldots x_N) = 0$$
$$\ldots$$
$$F_N(x_1, x_2, x_3, \ldots x_N) = 0$$

Así, un sistema de ecuaciones no lineales corresponderá a una expresión de la forma $\vec{F}(\vec{x}) = \vec{0}$ donde $\vec{F} : D \subset \mathbf{R^N} \to \mathbf{R^N}$. Resolver un sistema de ecuaciones no lineales consistirá en encontrar $\vec{x}_* \in D$ tal que $\vec{F}(\vec{x}_*) = \vec{0}$.

Para la resolución de sistemas de ecuaciones no lineales puede extenderse de modo más o menos inmediato la idea del método de Newton para resolver ecuaciones escalares. Así, supongamos que se tiene una cierta aproximación $\vec{x}^n$ de $\vec{x}_*$, solución del sistema de ecuaciones lineales. Localmente, podemos sustituir $\vec{F}(\vec{x})$ por el desarrollo de Taylor de grado uno de $\vec{F}$ en torno a $\vec{x}^n$, $\vec{P}_n(\vec{x})$, y buscar (resolviendo el correspondiente sistema de ecuaciones lineales) el vector que hace nulo este desarrollo, que se tomará como $\vec{x}_{n+1}$.

Teniendo en cuenta que (siempre que $\vec{F}$ sea suficientemente regular) $\vec{P}_n$ vendrá dado por

$$\vec{P}_n(\vec{x}) = \vec{F}(\vec{x}^n) + D\vec{F}(\vec{x}^n)\ (\vec{x} - \vec{x}^n)$$

y que $\vec{x}_{n+1}$ se define (tal y como se acaba de exponer) resolviendo $\vec{P}_n(\vec{x}_{n+1}) = \vec{0}$, es claro que $\vec{x}_{n+1}$ se obtendrá resolviendo

$$\vec{F}(\vec{x}^n) + D\vec{F}(\vec{x}^n)\ (\vec{x}_{n+1} - \vec{x}^n) = \vec{0}$$

Así, dada la aproximación $\vec{x}_n$, se calcula una nueva aproximación $\vec{x}_{n+1}$ resolviendo el sistema de ecuaciones lineales

$$D\vec{F}(\vec{x}^n)\ \vec{x}_{n+1} = D\vec{F}(\vec{x}^n)\ \vec{x}_n\ -\ \vec{F}(\vec{x}^n)$$

y el algoritmo global quedará (reorganizando los términos del sistema de ecuaciones lineales para reducir el número de operaciones)

`Se toma` $\vec{x}_0$ `próximo a` $\vec{x}_*$

`Para` $n = 0, 1, 2 \ldots$

- `Se calcula` $A = D\vec{F}(\vec{x}^n)$
- `Se calcula` $\vec{b} = -\ \vec{F}(\vec{x}^n)$

- `Se calcula` $\Delta^n\vec{x}$ `resolviendo`

$$A\Delta^n\vec{x} = \vec{b}$$

- `Se toma` $\vec{x}_{n+1} = \vec{x}_n + \Delta^n\vec{x}$

En relación con las propiedades de convergencia del método, cabe esperar que las propiedades de convergencia local sean similares a las del método de Newton para ecuaciones escalares. Obsérvese que el resultado de convergencia local del método de Newton para una ecuación escalar empleaba la hipótesis de que la raíz fuese simple (lo que implicaba en particular que $f'(x_*) \neq 0$) para asegurar una convergencia cuadrática del esquema. Para el método de Newton en sistemas de ecuaciones de no lineales, parece natural que la condición anterior se transforme en una hipótesis de regularidad de la matriz $D\vec{F}(\vec{x}_*)$.

Así, en efecto puede demostrarse que el método de Newton para la resolución de sistemas de ecuaciones no lineales converge localmente, con convergencia cuadrática, siempre que $\vec{F}$ sea de clase C^2 en el entorno de la solución $\vec{x}_*$ y se verifique la matriz $D\vec{F}(\vec{x}_*)$ es regular. Puede consultarse la demostración (con unas condiciones algo más débiles sobre la regularidad de $\vec{F}$) en el texto de Quarteroni, Sacco y Saleri citado al final de este tema.

Obsérvese que, en relación con la convergencia global del método de Newton, cabe esperar (en el mejor de los casos) las mismas dificultades que presentaba el método para la resolución de una ecuación escalar. De este modo, sólo si el iterante inicial del método de Newton $\vec{x}_0$ está cerca de la solución $\vec{x}_*$ cabe esperar la convergencia. En la práctica, al margen de la necesidad de contar siempre con una buena estimación de la solución, el método de Newton para la resolución de sistemas debe ser complementado con técnicas más robustas (más adelante se comentará algo sobre el modo de hacer más robusto el propio método).

Retomando de nuevo el análisis del problema escalar, la condición $f'(x_*) \neq 0$ tenía también que ver con el condicionamiento del problema, ya que valores muy pequeños de $|f'(x_*)|$ daban lugar a problemas muy mal condicionados. Es fácil observar que también aquí la regularidad de la matriz $D\vec{F}(\vec{x}_*)$ determina el condicionamiento de la resolución del sistema de ecuaciones no lineales.

Así, se condidera el problema original

$$\vec{F}(\vec{x}_*) = \vec{0}$$

y un problema *perturbado*

$$\vec{F}(\vec{x}_* + \delta\vec{x}_*) + \vec{\epsilon} = \vec{0}$$

donde la introducción de una perturbación (vectorial) $\vec{\epsilon}$ en los valores de la función $\vec{F}$ origina una perturbación $\delta\vec{x}_*$ de la solución.

Si, como en el caso escalar, se considera $\vec{F}$ regular y se supone que $\delta\vec{x}_*$ es pequeño, se puede desarrollar

$$\vec{F}(\vec{x}_* + \delta\vec{x}_*) = \vec{F}(\vec{x}_*) + D\vec{F}(\vec{x}_*)\delta\vec{x}_*) + \mathcal{O}(\|\delta\vec{x}_*\|^2)$$

y, truncando el desarrollo, se obtiene una ecuación aproximada para determinar $\delta\vec{x}_*$:

$$D\vec{F}(\vec{x}_*)\delta\vec{x}_* \simeq -\vec{\epsilon} = \vec{0}$$

De este modo, olvidando que tan sólo se trata de una aproximación, si se toma

$$\delta\vec{x}_* = -\left(D\vec{F}(\vec{x}_*)\right)^{-1}\vec{\epsilon}$$

puede acotarse (suponiendo $D\vec{F}(\vec{x}_*$ regular)

$$\|\delta\vec{x}_*\| \leq \| \left(D\vec{F}(\vec{x}_*)\right)^{-1} \| \, \|\vec{\epsilon}\|$$

representando entonces $\| \left(D\vec{F}(\vec{x}_*)\right)^{-1} \|$ el factor de amplificación de los errores (absolutos). De este modo, se ha visto que el condicionamiento de la matriz $D\vec{F}(\vec{x}_*)$ determina el condicionamiento del problema de resolución del sistema de ecuaciones lineales.

Obsérvese también que el condicionamiento del sistema de ecuaciones lineales asociado a la matriz $D\vec{F}(\vec{x}_*)$ (que, como se acaba de ver, condiciona la resolución del sistema de ecuaciones no lineales *independientemente del método que se emplee*) interviene también en el condicionamiento de cada etapa del método de Newton, pues obliga a resolver sistemas de ecuaciones asociados a la matriz $D\vec{F}(\vec{x}_n)$ (y ésta será próxima a $D\vec{F}(\vec{x}_*)$ si, como esperamos, $\vec{x}_n$ es próximo a $\vec{x}_*$).

Algunas modificaciones del método de Newton

Tal y como se ha mencionado, el método de Newton para sistemas presenta problemas de convergencia si el iterante no está suficientemente cerca de la solución. Por esta razón la implementación práctica del método de Newton se hace siempre en combinación con otro tipo de técnicas (como ya se mencionó en el caso de ecuaciones escalares) o modificando el método para hacerlo más robusto. En particular, hay dos enfoques para hacer el método de Newton más robusto:

- Modicar el *paso* del método de forma que cada iteración de Newton calcula (empleando la notación usada previamente para describir el algoritmo)

$$\vec{x}_{n+1} \; = \; \vec{x}^{\,n} \; + \; \omega^{\,n} \, \Delta^{\,n}\vec{x}$$

  dando lugar a un esquema habitualmente denominado método de Newton amortigüado. La idea central está en que el método de Newton puede proporcionar una buena dirección para buscar $\vec{x}_{n+1}$ pero, en cambio, el valor del paso dado en esa dirección puede ser inadecuado si se está lejos de la solución (ya que el método se basa en aceptar como buena la aproximación lineal que devuelve el polinomio de Taylor de grado uno, lo cual sólo parece justificado cuando se está ya muy cerca de la solución).

- Eligir una dirección para el vector $\vec{x}_{n+1} - \vec{x}^n$ que combine la proporcionada por el método de Newton (determinada por $\Delta^n\vec{x}$) con la que devuelven otras técnicas más robustas. De este modo se espera combinar la robustez de esas otras técnicas (otorgando, cuando todavía se está lejos de la solución, un mayor peso a la dirección de búsqueda que proponen) con la rápida convergencia local del método de Newton (dando, una vez cerca de la solución, un mayor peso a la predicción de este método).

En el primero de los dos enfoques, el parámetro ω^n se elegirá de modo que se logre, en ese paso, la mayor reducción posible del residuo en la dirección $\Delta^n\vec{x}$ de modo que, idealmente, ω^n se elige de forma que

$$\|\vec{F}(\vec{x}^n + \omega^n\Delta^n\vec{x})\| \leq \|\vec{F}(\vec{x}^n + \omega\Delta^n\vec{x})\| \qquad \forall\omega$$

En la práctica, este problema de minimización (se trata de buscar el mínimo de una función escalar que depende de un único parámetro ω) se resuelve solamente de forma aproximada (a fin de poder obtener el paso ω^n con un coste computacional asumible, que haga que el método global resulte eficiente). En el texto de Dahlquist y Björck mencionado al final del tema pueden encontrarse los detalles de alguna de estas técnicas (como la de Armijo-Goldstein).

Por otro lado, el segundo enfoque trata de combinar las propiedades de robustez de algún método alternativo con la convergencia local cuadrática del método de Newton. Por ejemplo, puede combinarse (en lo que se denomina el método híbrido de Powell) la dirección que proporciona un método de máximo descenso para el problema de minimización del funcional $J(\vec{x}) = \frac{1}{2}\|\vec{F}(\vec{x})\|^2$ (que se formula de un modo similar al método del mismo nombre para resolver sistemas de ecuaciones lineales asociados a matrices simétricas y definidas positivas, donde se minimizaba un funcional cuadrático), que viene dada por

$$\vec{d}^n = -\vec{\nabla}J(\vec{x}) = -D\vec{F}(\vec{x})^T\vec{F}(\vec{x})$$

y la dirección propuesta por el método de Newton. Este método elige en cada paso un parámetro β^n (en función de lo cerca que parezca encontrarse de la solución) y construye $\vec{x}_{n+1}$ mediante

$$\vec{x}_{n+1} = \vec{x}_n + \beta_n\vec{d}^n + (1-\beta_n)\Delta^n\vec{x}$$

Por otro lado, además de las dificultades de convergencia global, el método de Newton presenta una grave deficiencia desde el punto de vista computacional ligada al empleo de la matriz jacobiana, $D\vec{F}$, en cada paso. Obsérvese que ello implica no sólo un alto coste computacional de cada iteración del método de Newton sino, además, la necesidad de conocer explícitamente dicha matriz. Así, la resolución de un cierto sistema de N ecuaciones no lineales mediante un código basado en el método de Newton obliga a escribir (en algún archivo) N funciones para describir el sistema que se debe resolver (algo que habrá de hacerse con cualquier método de resolución) y, después, calcular y programar N^2 funciones para describir cada uno de los elementos de la matriz jacobiana.

De este modo, el cálculo y la programación de las N^2 funciones puede convertirse (para N grande) en una tarea enormemente tediosa y además fuente de probables errores (tanto

en el cálculo de las derivadas como en la escritura de éstas). Existen dos alternativas para tratar de aliviar estas tareas y evitar los posibles errores

- Cálculo automático de la matriz jacobiana: mediante cálculo simbólico pueden leerse las expresiones que definen las componente de $\vec{F}$, obtenerse las expresiones de las derivadas y generar un código que permita su cálculo.
- Empleo de valores aproximados de la matriz jacobiana: no se utiliza exactamente $D\vec{F}$ en las iteraciones del método de Newton sino una aproximación de dicha matriz (obtenida mediante derivación numérica o generando simultáneamente una sucesión de matrices que aproximan dicha matriz), denominando a los métodos así construidos métodos de cuasi-Newton.

En todo caso, persiste la dificultad asociada al elevado coste por paso del método al tener que evaluar una matriz de tamaño N^2 (donde N representa el número de ecuaciones) y resolver un sistema asociado a dicha matriz en cada paso. Debe observarse que si dicha matriz no cambiase con cada iteración no sólo se evitaría el coste asociado a su construcción sino que la resolución del sistema de ecuaciones sería mucho menos costoso (ya que podría factorizarse la matriz una sola vez de forma que en cada iteración sólo se resolviesen dos sistemas con matrices triangulares). De este modo, es habitual en la implementación del método de Newton que pueda actualizarse la matriz jacobiana sólo cada cierto numéro de etapas (que deberá controlarse con cuidado para evitar que el ahorro computacional por paso no deteriore la convergencia del esquema conduciendo a mayores costes computacionales globales).

3.7. Códigos disponibles

La ubicuidad del problema de resolución de S.E.L. dentro del cálculo científico hace, como ya se ha mencionado, que se haya desarrollado un abundante software muy eficiente para la resolución de S.E.L. tanto con métodos directos como con métodos iterativos. Para convencerse de este hecho, basta con comprobar las centenares de entradas referidas al álgebra lineal en NETLIB o GAMS (ver dirección en Tema 1).

El propio programa MATLAB, inicialmente creado como una interfaz de acceso sencillo a las bibliotecas LINPACK y EISPACK, que después fueron sustituidas por LAPACK y ampliada con otras bibliotecas, es un buen ejemplo del amplio desarrollo de software para la resolución de sistemas lineales. Además, LAPACK constituye un software enormemente eficiente (todas las operaciones de bajo nivel emplean funciones programadas para cada procesador) y fiable (ha sido actualizado y depurado durante décadas a través de una amplísima comunidad de desarrolladores y usuarios).

También se ha mencionado que la resolución de grandes sistemas de ecuaciones plantea retos computacionales notables, lo que requiere un cuidadoso desarrollo del software correspondiente. Puede encontrarse una lista del software disponible en

`http://www.netlib.org/utk/people/JackDongarra`
ordenado en los siguientes grupos:

- métodos de resolución directos
- métodos directos para grandes matrices huecas
- métodos iterativos para grandes matrices huecas
- precondicionadores
- subrutinas básicas

Para el caso particular de los métodos directos para grandes sistemas con matrices huecas existe una lista algo más completa en la página
`http://www.cise.ufl.edu/research/sparse/codes/`

Puede, por otro lado, consultarse un estudio comparativo de diferentes paquetes basados en métodos directos para grandes sitemas en

- N.I.M. Gould; Y. Hu; J.A. Scott: *A numerical evaluation of sparse direct solvers for the solution of large sparse, symmetric linear systems of equations*. Technical Report CCLRC, May 2005, accesible en `http://www.clrc.ac.uk`

Si se accede a la documentación de cualquiera de los paquetes contenidos en las listas anteriores se verá que la programación ha de ser muy cuidadosa para lograr una resolución eficiente que debe responder a numerosas cuestiones importantes. En particular deben aprovechar (de un modo transparente para el usuario) las posibilidades de paralelización, de vectorización (a través de subrutinas básicas, como BLAS, adaptadas a cada arquitectura), emplear algoritmos por bloques adaptados al tamaño de la memoria disponible, etc.

3.8. Referencias

- **Saad, Y.; *Iterative Methods for Sparse Linear Systems*. SIAM, 2003**

  Amplísimo texto sobre métodos iterativos. En la página del autor puede encontrarse una copia de la primera edición del libro (publicada por PWS en 1996):

  `http://www-users.cs.umn.edu/~saad/books.html`

- **Ciarlet, P.G.; *Introduction to numerical linear algebra and optimisation*. Cambridge University Press, 1989**

  Referencia más básica que la anterior que detalla el análisis numérico de los métodos más elementales.

- **Dahlquist, G; Björck, A.; *Numerical Methods in Scientific Computing.* SIAM, en preparación**

  Este texto corresponde a una referencia completísima que amplia notablemente un texto anterior de los mismos autores. El segundo autor mantiene una copia de los borradores en la dirección

  `http://www.mai.liu.se/~akbjo/NMbook.html`

- **Varios autores; *LAPACK Users' Guide.* SIAM, 1999.**

- **Varios autores; *LAPACK95 Users' Guide.* SIAM, 2001.**

  Estos manuales de la biblioteca LAPACK permite comprender las diversas cuestiones que aparecen en el diseño de subrutinas eficientes del álgebra matricial. El texto completo del manual puede consultarse en la dirección

  `http://www.netlib.org/lapack/lug/index.html`

  `http://www.netlib.org/lapack95/lug95/`

  Por otro lado, en relación con la escalabilidad de los algoritmo y los retos que plantea la resolución de (muy) grandes sistemas de ecuaciones (o, en general, los problemas de gran tamaño que se plantean con frecuencia en el cálculo científico) pueden consultarse diversos trabajos en la página de J. Dongarra

  `http://www.netlib.org/utk/people/JackDongarra/`

- **D. Kincaid; W. Cheney; *Análisis Numérico. Las Matemáticas del Cálculo Científico.* Addison-Wesley Iberoamericana, 1994.**

  Existe una edición actualizada de este texto titulada *Numerical Analysis: Mathematics of Scientific Computing. 3rd ed.* publicada por Brooks/Cole en 2002, de la que no existe aún traducción.

- **A. Quarteroni; R. Sacco; F. Saleri; *Numerical Mathematics*: Springer, 2000.**

Capítulo 4

Cálculo de autovalores y autovectores

4.1. Motivación

Existen multitud de problemas que conducen, directa o indirectamente, a la resolución de un problema de cálculo de autovalores (y autovectores) de una matriz. Agunos ejemplos de problemas de este tipo son los siguientes:

- El cálculo del índice *PageRank* de *Google*, ya descrito en el Tema 1, consiste (en su formulación básica) en la obtención de un autovector asociado al autovalor principal de una matriz de conectividad de las páginas de Internet (obtenida a través de sus enlaces y que, de hecho, correspondería a la matriz de transición en un proceso de navegación aleatoria)

- El cálculo de modos y frecuencias propias en el campo de la acústica o el cálculo de los modos de propagación en guías de ondas (electromagnéticas) consituyen problemas de cálculo de autovalores algo más generales, puesto que se trata del cálculo de autovalores de operadores diferenciales. En todo caso, el cálculo efectivo de estas frecuencias y modos en problemas de interés práctico (donde no se dispone de herramientas analíticas que permitan devolver dicha información) pasa por la discretización espacial de las correspondientes ecuaciones en derivadas parciales (en la mayor parte de los casos mediante técnicas de diferencias finitas o de elementos finitos), lo que conduce directamente a un problemas de autovalores y autovectores de matrices (donde el tamaño de estas matrices viene dado por el número de grados de libertad empleados en la discretización espacial y que deberá ser alto si deseamos obtener una buena aproximación de las frecuencia propias y sus modos asociados)

- El análisis (lineal) de estabilidad de sistemas (que pueden ser distribuidos o no) pasa por el estudio del sistema linealizado en torno a la solución (estacionaria) cuya estabilidad queremos analizar (habitualmente, se tratará del régimen de funcionamiento para el cual se ha configurado determinado dispositivo, sistema o proceso). De este modo, se estudiará la evolución de las perturbaciones con respecto a la solución básica y el

problema linealizado consistirá, con carácter general, en un conjunto de ecuaciones diferenciales ordinarias lineales de coeficientes constantes (si se trataba de un sistema no distribuido) o un conjunto de ecuaciones en derivadas parciales lineales de coeficientes constantes (si se trataba de un sistema distribuido). El comportamiento cualitativo de las soluciones de este problema queda completamente caracterizado por los autovalores de la matriz (en el caso de los sistemas no distribuidos) o del operador diferencial (en el caso de sistemas distribuidos). De nuevo, como en el ejemplo anterior, para los sistemas distribuidos será preciso en la práctica proceder en primer lugar a una discretización espacial que convertirá el problema original en un problema de cálculo de autovalores de una matriz. Así las cosas, la existencia de autovalores con parte real positiva para la correspondiente matriz permitirá detectar la inestabilidad del sistema considerado (véase, por ejemplo, el texto *Ecuaciones diferenciales y en diferencias. Sistemas dinámicos* reseñado al final del tema).

- El cálculo de polos y ceros de funciones de transferencia de sistemas lineales consiste en la búsqueda de (todas las) raíces de dos polinomios y, formulado así, cae en el rango de problemas abordados en el Tema 2. Sin embargo, en la práctica es posible reescribir este problema (tal y como se verá en la sección siguiente) como la búsqueda, para cada uno de los polinomios, de todos los autovalores de una cierta matriz. Además, existen diversas razones (algunas de las cuales serán apuntadas más adelante) para preferir esta segunda formulación.

Cabe hacer, en este momento, algunas observaciones sobre los problemas de cálculo de autovalores y autovectores de matrices.

- El tamaño de las matrices que aparecen en los problemas descritos anteriormente es muy variable y va desde matrices de tamaño muy reducido (matrices de tamaño muchas veces inferior a 10×10 en el cálculo de polos y ceros de funciones de transferencia o en el análisis de estabilidad de sistemas no distribuidos simples) a matrices enormemente grandes (en septiembre de 2005 *Google* referenciaba unos 8000 millones de páginas web, lo que devolvía una matriz de tamaño $8 \cdot 10^9 \times 8 \cdot 10^9$).

- En la práctica totalidad de los problemas asociados a matrices de gran tamaño, éstas tienen una estructura especial: su carácter hueco, de modo que sólo una pequeña fracción de los elementos de dicha matriz toma valores no nulos (por ejemplo, en el caso de la matriz de conectividad de Internet, esta propiedad se deriva del hecho de que cada página contiene enlaces a un número de páginas claramente muy pequeño en comparación con el número total de páginas en la Web). Esta estructura debe ser tenida en cuenta a la hora de almacenar dichas matrices y operar con ellas. De hecho, sólo si se explota esta estructura se obtienen problemas tratables desde el punto de vista computacional. En algunos problemas incluso, como es el caso del cálculo de *PageRank* éstas matrices no pueden ser almacenadas y es preciso recurrir a una navegación aleatoria sobre la Web para obtener la acción de la matriz de conectividad sobre un cierto vector (que puede ser interpretado como una distribución de navegadores en distintas páginas).

- Dentro de los problemas asociados al cálculo de autovalores y autovectores, existen diferentes tipos según cuál sea el interés del cálculo. Para comenzar, en algunos problemas interesa calcular sólo los autovalores (por ejemplo, en la obtención de polos y ceros de una función de transferencia, donde los autovectores no tienen ningún interés), en otros sólo los autovectores (por ejemplo, en el cálculo del índice *PageRank*) y en algunos interesan tanto los autovalores como los autovectores (así ocurre en el cálculo de frecuencias y modos propios en acústica o guías de ondas). Además, aunque hay problemas (pocos) en los que el interés es calcular todos los autovalores o autovectores (el cálculo de polos y ceros de una función de transferencia vuelve a ser un ejemplo), lo habitual es que el interés se centre en uno o unos pocos autovalores o autovectores (las primeras frecuencias propias en acústica o guías de ondas, los autovalores con mayor parte real en el análisis de estabilidad o el autovector asociado al autovalor principal en el caso del índice *PageRank*).
- Existen algunos problemas de autovalores más generales y problemas relacionados (como la descomposición en valores singulares) cuya resolución se aborda mediante técnicas similares a las expuestas en este tema que, sin embargo, no será abordados aquí.

4.2. Algunas cuestiones generales

En esta sección se abordarán diversas cuestiones relativas al cálculo de autovalores (y autovectores) de matrices, como son

- relación entre el cálculo de autovalores de una matriz y las raíces de un polinomio
- condicionamiento del problema de cálculo de autovalores (y autovectores) de matrices
- la clasificación de los métodos numéricos para el cálculo de autovalores (y autovectores) de matrices

Autovalores y ceros de polinomios

De la definición de autovalor, λ, de una matriz A como escalar para el cual existen vectores no nulos, $\vec{v}$, tales que $A\vec{v} = \lambda\vec{v}$ se deduce que λ es raíz del polinomio característico $p(\lambda) = det(A - \lambda I)$. Así, siempre sería posible calcular los autovalores de una matriz a través del cálculo de las raíces (posiblemente complejas) de dicho polinomio. Sin embargo, existen varias razones que desaconsejan, con carácter general, buscar los autovalores de este modo. Entre éstas se encuentran la necesidad de acotar previamente dichas raíces, la necesidad de trabajar con métodos capaces de aproximar raíces complejas o la enorme sensibilidad de las raíces de un polinomio de orden elevado a pequeñas perturbaciones de los valores de sus coeficientes (esto es, el mal condicionamiento del cálculo de las raíces de polinomios de grado elevado, como se pone de manifiesto en el conocido como polinomio de Wilkinson).

Alternativamente, dado un cierto polinomio, es posible construir una matriz (conocida habitualmente como *matriz de acompañamiento* o *matriz de compañía*) cuyos autovalores sean las raíces del polinomio dado. En particular, dado el polinomio

$$p(x) = x^n + a_1 x^{n-1} + a_2 x^{n-1} + \cdots + a_{n-1} x + a_n$$

puede comprobarse que la matriz definida mediante

$$A = \begin{pmatrix} -a_1 & -a_2 & -a_3 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & & 1 & 0 \end{pmatrix}$$

tiene como autovalores las raíces del polinomio dado.

Gracias a este resultado es posible entonces emplear los métodos numéricos de cálculo de autovalores de una matriz para obtener las raíces de un polinomio, para lo cual basta con construir su matriz de acompañamiento y calcular sus autovalores.

Condicionamiento del cálculo de autovalores

Dado el problema del cálculo de un autovalor λ_i de una cierta matriz $A \in \mathcal{M}_{n\times n}$ (de elementos reales), nos planteamos el estudio del efecto de pequeñas perturbaciones en los datos (esto es, en los coeficientes de la matriz) en los valores calculados del autovalor. Sea así un problema perturbado, donde se calcula el autovalor $\lambda_i + \delta\lambda_i$ asociado a la matriz perturbada $A + \delta A$.

El problema original y el problema perturbado se escriben

$$A\vec{v}_i = \lambda_i \vec{v}_i$$

$$(A + \delta A)(\vec{v}_i + \delta\vec{v}_i) = (\lambda_i + \delta\lambda_i)(\vec{v}_i + \delta\vec{v}_i)$$

de modo que tomando la diferencia se obtiene

$$A\delta\vec{v}_i + \delta A\vec{v}_i + \delta A\delta\vec{v}_i = \lambda_i\delta\vec{v}_i + \delta\lambda_i\vec{v}_i + \delta\lambda_i\delta\vec{v}_i$$

y si se desprecian los términos cuadráticos (lo cual parece razonable si la perturbación introducida es pequeña) se logra la relación

$$(A - \lambda_i I)\delta\vec{v}_i + \delta A\vec{v}_i = \delta\lambda_i\vec{v}_i \tag{4.1}$$

Distingamos ahora dos situaciones:

- A es simétrica
- A no es simétrica

Comenzamos por analizar el caso en que A es simétrica. Multiplicando la relación (4.1) por el vector $\vec{v}_i^t$ a la izquierda se obtiene

$$\vec{v}_i^t(A-\lambda_i I)\delta\vec{v}_i + \vec{v}_i^t\delta A\vec{v}_i = \delta\lambda_i\vec{v}_i^t\vec{v}_i$$

Puede observarse que el primer término de esta relación es nulo ya que

$$\vec{v}_i^t(A-\lambda_i I) = \left((A^t-\lambda_i I)\vec{v}_i\right)^t = ((A-\lambda_i I)\vec{v}_i)^t = \vec{0}^t$$

obteniéndose así

$$\delta\lambda_i\|\vec{v}_i\|^2 = \vec{v}_i^t\delta A\vec{v}_i$$

y, finalmente, tomando normas y acotando

$$|\delta\lambda_i|\|\vec{v}_i\|^2 = |\vec{v}_i^t\delta A\vec{v}_i| \leq \|\vec{v}_i\|\ \|\delta A\vec{v}_i\| \leq \|\vec{v}_i\|\ \|\delta A\|_M\|\vec{v}_i\|$$

que permite concluir

$$|\delta\lambda_i| \leq \|\delta A\|_M \tag{4.2}$$

de modo que el problema tiene un condicionamiento *óptimo*, ya que los errores sobre la matriz no se amplifican en absoluto al calcular sus autovalores (obsérvese que no puede obtenerse nada mejor, ya que si la matriz A es diagonal y la perturbación δA también lo es, las perturbaciones de los autovalores coinciden exactamente con las perturbaciones en la matriz).

Abordemos ahora el segundo caso, donde A es una matriz no simétrica. Sea ahora $\vec{w}_i$ un autovector a la izquierda de la matriz A asociado al autovalor λ_i (recuérdese que los autovectores a la izquierda de una matriz son –o pueden verse como– los autovectores asociados a la matriz traspuesta, que tiene los mismos autovalores que la matriz de partida).

Multiplicando ahora la relación (4.1) por el vector $\vec{w}_i^t$ a la izquierda se obtiene

$$\vec{w}_i^t(A-\lambda_i I)\delta\vec{v}_i + \vec{w}_i^t\delta A\vec{v}_i = \delta\lambda_i\vec{w}_i^t\vec{v}_i$$

y de nuevo el primer término resulta nulo de modo que

$$|\delta\lambda_i|\ |\vec{w}_i^t\vec{v}_i| = |\vec{w}_i^t\delta A\vec{v}_i| \leq \|\vec{w}_i^t\|\ \|\delta A\|_M\ \|\vec{v}_i\|$$

Así, para el caso de una matriz no simétrica se ha logrado la acotación

$$|\delta\lambda_i| \leq \|\delta A\|_M\ \frac{1}{|\frac{\vec{w}_i^t}{\|\vec{w}_i\|}\frac{\vec{v}_i}{\|\vec{v}_i\|}|}$$

o bien, eligiendo autovectores normalizados

$$|\delta\lambda_i| \leq \|\delta A\|_M\ \frac{1}{|\vec{w}_i^t\vec{v}_i|}$$

De este modo, a diferencia del caso con matriz simétrica, cabe esperar ahora una amplificación de las perturbaciones en los elementos de la matriz a través del factor

$$\frac{1}{|\vec{w}_i^t \vec{v}_i|}$$

que, como se verá posteriormente, es fácilmente calculable *a posteriori* (es decir, una vez obtenida la aproximación del autovalor λ_i), permitiendo estimar si cabe o no esperar un grave deterioro de la aproximación del autovalor debido a los errores de redondeo.

Clasificación de los métodos

Desde luego, el cálculo analítico de los autovalores de una matriz sólo es posible si ésta tiene un tamaño muy reducido (en teoría es posible obtener expresiones analíticas para matrices de tamaño igual o inferior a 4×4, pero las fórmulas son complicadas por lo que apenas se usarán salvo para matrices 2×2) o si su estructura es verdaderamente particular. Así, con carácter general, será preciso emplear métodos numéricos que devolverán valores aproximados de dichos autovalores.

Podemos clasificar los métodos numéricos para la aproximación de autovalores de matrices en dos clases, dependiendo del objeto de la aproximación:

- métodos que buscan un (o unos pocos) autovalor de una matriz
- métodos que buscan todos los autovalores de una matriz

Las siguientes secciones abordan un método de cada uno de los dos tipos: el método de la potencia (como ejemplo de método que busca aproximar un único autovalor) y el método basado en la factorización QR (como ejemplo de método que busca aproximar todos los autovalores de una matriz). Existen, desde luego, muchos más métodos pero, en cierta medida, presentan muchas similitudes con los que se van a a presentar aquí y los métodos expuestos tienen la ventaja de su sencillez (que permitirá entender mejor las ideas principales de los métodos para la aproximación de autovalores de matrices).

En todo caso, la presentación que se hará en este tema se refiere al cálculo de autovalores para matrices de un tamaño moderado. La aproximación de autovalores de grandes matrices huecas presenta muchas particularidades, algunas de ellas puramente computacionales (como el almacenamiento de las matrices y el modo de implementar de forma eficiente las operaciones con estas matrices, incluyendo la paralelización de los cálculos) pero también otras más básicas (que se refieren al diseño de métodos numéricos especialmente adaptados a este tipo de problemas). Nada de esto será abordado aquí, pudiendo consultarse (por ejemplo) la obra *Numerical Linear Algebra for High-Performance Computers* descrita al final del tema como introducción al cálculo de autovalores de grandes matrices huecas.

4.3. Método de la potencia y sus variantes

Se considera una matriz $A \in \mathcal{M}_{n\times n}$ (de elementos reales). Supongamos que A es diagonalizable y sus autovalores verifican

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \tag{4.3}$$

Sea, por otro lado, $\vec{u}_0$ un cierto vector de $\mathbf{R}^n$. El método de la potencia propone generar, a partir de $\vec{u}_0$, una sucesión mediante

$$\vec{u}_k = A\vec{u}_{k-1}\,, \quad k = 1, 2, \ldots$$

basándose en la idea de que el producto de la matriz A por un vector *amplifica* cada componente del vector (dada su descomposición en una base de autovectores) en un factor dado por el autovalor asociado y, por lo tanto, crecerán más las correspondientes a autovalores de mayor módulo.

Así, dada la descomposición de $\vec{v}_0$ en una base de autovectores ($\vec{v}_j$ representa un autovector asociado al autovalor λ_j):

$$\vec{u}_0 = c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_n\vec{v}_n$$

se tendrá para un iterante genérico

$$\vec{u}_k = A\vec{u}_{k-1} = A^k\vec{u}_0 = c_1A^k\vec{v}_1 + c_2A^k\vec{v}_2 + \cdots + c_nA^k\vec{v}_n = c_1\lambda_1^k\vec{v}_1 + c_2\lambda_2^k\vec{v}_2 + \cdots + c_n\lambda_n^k\vec{v}_n$$

y de aquí

$$\vec{u}_k = \lambda_1^k\left(c_1\vec{v}_1 \underbrace{+c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\vec{v}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\vec{v}_n}\right)$$

donde, con $k \to \infty$, todos los términos marcados tienden a cero y así

$$\frac{1}{\lambda_1^k}\vec{u}_k \to c_1\vec{v}_1 \quad \text{para} \quad k \to \infty.$$

Se comprueba así que, en efecto, para k suficientemente grande (es decir, tras un número suficiente de pasos) se obtiene si $c_1 \neq 0$

$$\vec{u}_k \simeq \lambda_1^k c_1\vec{v}_1$$

de modo que $\vec{u}_k$ devuelve una aproximación del autovector asociado a λ_1 .

Si se desea ahora aproximar el autovalor λ_1, basta con comparar dos iterantes consecutivos (para k suficientemente grande) ya que, denotando mediante $u_{k,j}$ la componente j-ésima del vector $\vec{u}_k$, se tiene

$$\frac{u_{k+1,j}}{u_{k,j}} = \lambda_1 \frac{c_1 v_{1,j} + c_2(\frac{\lambda_2}{\lambda_1})^k v_{2,j} + \cdots + c_n(\frac{\lambda_n}{\lambda_1})^k v_{n,j}}{c_1 v_{1,j} + c_2(\frac{\lambda_2}{\lambda_1})^{k-1} v_{2,j} + \cdots + c_n(\frac{\lambda_n}{\lambda_1})^{k-1} v_{n,j}}$$

y, por lo tanto, siempre que $c_1 v_{1,j} \neq 0$ se tendrá

$$\frac{u_{k+1,j}}{u_{k,j}} \to \lambda_1 \quad \text{si} \quad k \to \infty$$

Obsérvese que la condición $c_1 v_{1,j} \neq 0$ se cumplirá para, al menos, un valor de j siempre y cuando $c_1 \neq 0$ (ya que, en otro caso, $\vec{v}_1 = \vec{0}$, lo que resulta imposible al tratarse de un autovector y, además, constituir un elemento de una base de $\mathbf{R}^n$).

En consecuencia, se ha obtenido un método que permite aproximar el autovalor dominante de una matriz (tal y como se denomina al autovalor con mayor módulo) así como un autovector asociado a éste.

Debemos hacer, no obstante, algunas observaciones sobre el método bajo la formulación propuesta.

- Desde el punto de vista computacional, la formulación presenta un grave inconveniente ya que:

  si $|\lambda_1| > 1$, entonces $\|\vec{u}_k\| \to \infty$

  si $|\lambda_1| < 1$, entonces $\|\vec{u}_k\| \to 0$

  y en ambos casos nos encontraremos con graves errores de redondeo en las iteraciones del método. Como se verá posteriormente, es preciso *normalizar* los iterantes del método para evitar esta dificultad.

- Se ha supuesto, para que el método proporcione aproximaciones del autovalor dominante y su autovector asociado, que $c_1 \neq 0$. Obsérvese que esta condición impone una restricción sobre los iterantes iniciales y, en sentido estricto, no permite garantizar la convergencia al autovalor dominante desde cualquier iterante inicial. De hecho, es fácil probar (repitiendo los razonamientos anteriores) que si $|\lambda_2| > |\lambda_3|$, $c_1 = 0$ y $c_2 \neq 0$, las sucesiones generadas por el método devuelven aproximaciones de λ_2 y un autovector asociado. En la práctica, no obstante, salvo que tanto el problema como el iterante inicial presenten numerosas simetrías o componentes nulas, lo que ocurrirá realmente es que c_1 tomará un valor muy pequeño (pero no nulo) y las sucesiones construidas sí devolverán una aproximación de λ_1, aunque la convergencia será muy lenta (ya que k deberá ser muy grande para que los términos marcados anteriormente sean pequeños en comparación con el término asociado a λ_1).

- Si no existe un autovalor (estrictamente) dominante, sino que la matriz (que suponemos aún diagonalizable) tiene dos autovalores dominantes (esto es, con el mismo módulo que es, además estrictamente mayor que el módulo del resto de los autovalores) la convergencia puede o no fallar, dependiendo de la situación que se presente:

si $\lambda_1 = \lambda_2$, se mantiene la convergencia al autovalor ya que

$$\frac{u_{k+1,j}}{u_{k,j}} = \lambda_1 \frac{c_1 v_{1,j} + c_2 v_{2,j} + c_3 (\frac{\lambda_3}{\lambda_1})^k v_{3,j} + \cdots + c_n (\frac{\lambda_n}{\lambda_1})^k v_{n,j}}{c_1 v_{1,j} + c_2 v_{2,j} + c_3 (\frac{\lambda_3}{\lambda_1})^{k-1} v_{3,j} + \cdots + c_n (\frac{\lambda_n}{\lambda_1})^{k-1} v_{n,j}}$$

al tiempo que se recuperará también la convergencia a algún autovector (combinación de $\vec{v}_1$ y $\vec{v}_2$)

si $\lambda_1 \neq \lambda_2$, se perderá la convergencia puesto que

$$\vec{u}_k = \lambda_1^k \left(c_1 \vec{v}_1 + \underbrace{c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \vec{v}_2} + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \vec{v}_n \right)$$

contiene un término (marcado en la ecuación) oscilante. En el caso $\lambda_1 = -\lambda_2$ aún puede salvarse la convergencia (comparando $\vec{u}_{k+2}$ y $\vec{u}_k$) pero para el caso general se hace preciso encontrar otro modo de aproximar estos autovalores (como se verá más adelante).

Como se ha observado, la formulación del método que se ha presentado no resulta adecuada desde el punto de vista computacional, haciéndose necesaria una *normalización* de la sucesión generada. Así, en la práctica, se toma un vector $\vec{u}_0$ con $\|\vec{v}_0\| = 1$ y se genera una sucesión de la forma

$$\vec{w}_k = A\vec{u}_{k-1}$$

$$\vec{u}_k = \frac{\vec{w}_k}{\|\vec{w}_k\|}$$

para $k = 1, 2, \ldots$. Denominaremos a este esquema como el método de la potencia normalizada y genera, como es claro, una sucesión $\{\vec{u}_k\}_{k=0}^{\infty}$ cuyos elementos tienen todos una norma unitaria.

Puede comprobarse fácilmente (empleando el método de inducción) que la sucesión generada verifica

$$\vec{u}_k = \frac{A^k \vec{u}_0}{\|A^k \vec{u}_0\|}$$

lo que permite examinar las propiedades de convergencia del método de la potencia normalizada repitiendo el tipo de argumentos empleado más arriba. En particular, puede comprobarse que si A es diagonalizable con autovalor (estrictamente) dominante λ_1 y se elige $\vec{u}_0$ tal que $c_1 \neq 0$ (donde c_1 representa de nuevo el coeficiente en la descomposición de $\vec{u}_0$ en la base de autovectores), se recupera

$$(\vec{u}_k)^t A \vec{u}_k \to \lambda_1$$

Método de la potencia inversa

Se plantea ahora la cuestión de aproximar el autovalor de menor módulo de una matriz. El método de la potencia expuesto anteriormente permite una adaptación inmediata para resolver este problema: basta con trabajar con la matriz inversa de A. Obsérvese que, puesto que los autovalores de la matriz inversa son los inversos de los autovalores de la matriz original, el autovalor dominante de A^{-1} será precisamente el autovalor de menor módulo de la matriz A y el método de la potencia empleando la matriz A^{-1} devolverá entonces una aproximación del inverso de este autovalor.

Así, el denominado método de la potencia inversa propone, dado un cierto vector $\vec{u}_0$, construir una sucesión de la forma

$$\vec{u}_k = A^{-1}\vec{u}_{k-1}\,, \quad k = 1, 2, \ldots$$

que, bajo determinadas condiciones, se espera que devuelva una aproximación de un autovector asociado al autovalor de menor módulo y permita, a su vez, obtener una aproximación de dicho autovalor.

Supongamos ahora que $A \in \mathcal{M}_{n\times n}$ (de elementos reales) es diagonalizable, verifica

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-2}| \geq |\lambda_{n-1}| > |\lambda_n| > 0$$

y $\vec{u}_0$ se toma de modo que al hacer su descomposición en la base de autovectores

$$\vec{u}_0 = c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_{n-1}\vec{v}_{n-1} + c_n\vec{v}_n$$

se tiene $c_n \neq 0$.

Recuperando el análisis llevado a cabo para el método de la potencia, es fácil concluir entonces que

$$\lambda_n^k\vec{u}_k \to c_n\vec{v}_n \quad \text{para } k \to \infty$$

así como que existe algún índice j para el cual $c_n v_{n,j} \neq 0$ y

$$\frac{u_{k+1,j}}{u_{k,j}} \to \frac{1}{\lambda_n} \quad \text{para } k \to \infty$$

Cabe, desde luego, hacer para el método de la potencia inversa las mismas observaciones que se expusieron sobre el método de la potencia (referidas a la conveniencia de la normalización de los vectores de la sucesión y a las posibles dificultades vinculadas a la elección de $\vec{u}_0$ o la inexistencia de un autovalor estrictamente dominante –que ahora deberán hacerse para la inexistencia de un autovalor con módulo estrictamente menor al de todos los demás–).

Es preciso, sin embargo, añadir una observación de tipo computacional. Se ha formulado el método empleando la matriz A^{-1} pero, como se ha comentado en muchas ocasiones, en muy raras ocasiones (si es que hay alguna) se calcula explícitamente la matriz inversa: en

su lugar, se calcula la acción de la matriz inversa sobre un vector resolviendo el sistema de ecuaciones lineales asociado. En particular, cada iteración del método de la potencia inversa se lleva a cabo en la práctica calculando $\vec{u}_{k+1}$ solución del sistema de ecuaciones lineales

$$A\vec{u}_{k+1} = \vec{u}_k$$

Debe observarse también, en relación con este último comentario, que en cada etapa del método de la potencia inversa se resuelve un sistema de ecuaciones asociado a la misma matriz (sólo cambia el término de segundo miembro). De este modo, habitualmente se procede a factorizar la matriz antes de iniciar el bucle de forma que, en cada etapa, simplemente se procede a llevar a cabo las etapas de descenso y remonte (que son, con diferencia, las etapas menos costosas desde el punto de vista computacional).

Método de la potencia inversa con decalado

Los métodos presentados hasta el momento permiten entonces aproximar (bajo ciertas condiciones) el autovalor de mayor módulo y el autovalor de menor módulo. Se plantea ahora la cuestión de si será posible adaptar esos métodos para calcular los autovalores más cercanos a un cierto número (posiblemente complejo). La respuesta es relativamente fácil: basta construir una matriz con todos los autovalores de la matriz original desplazados de modo que los autovalores que originalmente se encontraban cerca de ese número ahora lo estarán cerca del origen y podrá entonces emplearse el método de la potencia inversa para poder aproximarlos.

Así, si deseamos calcular el autovalor de la matriz A más próximo al escalar α bastaría con emplear el método de la potencia inversa para la matriz $M = A - \alpha I$ (que tendrá como autovalores $\mu_i = \lambda_i - \alpha$) de forma que se construiría, para un cierto $\vec{u}_0$ dado, una sucesión generada mediante

$$\vec{u}_k = (A - \alpha I)^{-1}\vec{u}_{k-1}\,, \quad k = 1, 2, \ldots$$

que, en la práctica, se calcula mediante la resolución de los sistemas de ecuaciones lineales

$$(A - \alpha I)\vec{u}_k = \vec{u}_{k-1}\,, \quad k = 1, 2, \ldots$$

Supongamos ahora que $A \in \mathcal{M}_{n\times n}$ (de elementos reales) es diagonalizable, verifica

$$|\lambda_1 - \alpha| \geq |\lambda_2 - \alpha| \geq \cdots \geq |\lambda_{n-2} - \alpha| \geq |\lambda_{n-1} - \alpha| > |\lambda_n - \alpha| > 0$$

y $\vec{u}_0$ se toma de modo que al hacer su descomposición en la base de autovectores

$$\vec{u}_0 = c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_{n-1}\vec{v}_{n-1} + c_n\vec{v}_n$$

se tiene $c_n \neq 0$.

En tal caso, rehaciendo de nuevo el análisis se garantiza que

$$(\lambda_n - \alpha)^k\vec{u}_k \to c_n\vec{v}_n \quad \text{para } k \to \infty$$

así como que existe algún índice j para el cual $c_n v_{n,j} \neq 0$ y

$$\frac{u_{k+1,j}}{u_{k,j}} \to \frac{1}{\lambda_n - \alpha} \quad \text{para } k \to \infty$$

4.4. Técnicas de deflación

El método de la potencia y sus variantes permiten (bajo determinadas circunstancias) calcular un autovalor con una cierta característica: ser el de mayor módulo, el de menor módulo o el más próximo a un cierto escalar dado. En la práctica puede (y suele) darse el caso de ser necesario calcular unos pocos autovalores con esa misma característica. Aunque, tal y como se ha visto previamente, este cálculo podría hacerse en teoría eligiendo cuidadosamente $\vec{u}_0$ (por ejemplo, podría calcularse el segundo autovalor con mayor módulo si $\vec{u}_0$ se toma de modo que su descomposición en la base de autovectores no contenga ninguna componente en $\vec{v_1}$), en la práctica los errores de redondeo harán imposible la convergencia (de forma que las sucesiones convergerán, aunque sea lentamente, al primer autovalor).

Alternativamente, se va a considerar a continuación una técnica que permite, dada una cierta matriz A y un autovalor de esta matriz, construir una segunda matriz B que contenga el resto de los autovalores de A. Esta técnica se conoce como deflación y es claro que los métodos estudiados aplicados a la matriz B nos devolverán ahora los autovalores buscados (por ejemplo, si se construye B empleando el autovalor dominante, el método de la potencia aplicado a la matriz B nos devolverá el segundo autovalor en módulo).

Puesto que la técnica de deflación que se va a presentar se apoya en la construcción de unas matrices con ciertas propiedades interesantes y que volverán a aparecer más tarde, se comenzará por su definición y el estudio de sus principales propiedades.

Definición 4.1 *Dado un vector $\vec{v} \in \mathbf{R}^n$ se denomina matriz de Householder asociada a $\vec{v}$ (y se nota mediante $H(\vec{v})$, o H cuando no exista ambigüedad posible) a la matriz definida por*

$$H(\vec{v}) = I - 2\frac{\vec{v}\vec{v}^t}{\|\vec{v}\|_2^2}$$

Lema 4.1 *Sea H la matriz de Householder asociada a un cierto vector $\vec{v}$. Entonces, se verifica*

a) $H = H^t = H^{-1}$ (esto es, H es simétrica y ortogonal)

b) $H\vec{v} = -\vec{v}$

Demostración: El apartado a) es bastante sencillo de demostrar ya que, por un lado,

$$H_{ij} = \delta_{ij} - 2\frac{v_i v_j}{\|\vec{v}\|_2^2}$$

lo que demuestra que la matriz es simétrica, y

$$H^2 = \left(I - 2\frac{\vec{v}\vec{v}^t}{\|\vec{v}\|_2^2}\right)^2 = I - 4\frac{\vec{v}\vec{v}^t}{\|\vec{v}\|_2^2} + 4\frac{\vec{v}\vec{v}^t\vec{v}\vec{v}^t}{\|\vec{v}\|_2^4} = I$$

de modo que $H = H^{-1}$.

También el apartado b) se obtiene de modo inmediato pues

$$H\vec{v} = \left(I - 2\frac{\vec{v}\vec{v}^t}{\|\vec{v}\|_2^2}\right)\vec{v} = \vec{v} - 2\vec{v}$$

□

Empleando ahora matrices de Householder se tiene el siguiente resultado:

Proposición 4.1 *Sea A una matriz y λ_i un autovalor de dicha matriz con autovector asociado $\vec{v}_i$ normalizado de modo que $\|\vec{v}_i\|_2 = 1$.*

Sea asimismo H la matriz de Householder asociada al vector $\vec{v} = \vec{v}_i - \vec{e}_1$ (donde $\vec{e}_1$ representa el primer vector de la base canónica).

Entonces se obtiene

$$H\ A\ H = \begin{pmatrix} \lambda_i & \vec{c} \\ 0 & B \end{pmatrix}$$

Demostración: En primer lugar comprobaremos (ésta es la parte tediosa de la demostración) que

$$H\vec{v}_i = \vec{e}_1$$

Así, sea

$$H\vec{v}_i = \left(I - 2\frac{(\vec{v}_i - \vec{e}_1)(\vec{v}_i - \vec{e}_1)^t}{(\vec{v}_i - \vec{e}_1)^t(\vec{v}_i - \vec{e}_1)}\right)\vec{v}_i = \vec{v}_i - 2\frac{(\vec{v}_i - \vec{e}_1)(\vec{v}_i - \vec{e}_1)^t\vec{v}_i}{(\vec{v}_i - \vec{e}_1)^t(\vec{v}_i - \vec{e}_1)}$$

donde, examinando el numerador y denominador del segundo término, se concluye (notando mediante $v_{i,1}$ la primera componente de $\vec{v}_i$)

$$(\vec{v}_i - \vec{e}_1)\ (\vec{v}_i - \vec{e}_1)^t\ \vec{v}_i = (\vec{v}_i - \vec{e}_1)\ (1 - v_{i,1})$$

$$(\vec{v}_i - \vec{e}_1)^t\ (\vec{v}_i - \vec{e}_1) = 2 - 2v_{i,1}$$

y, por lo tanto, se tiene en efecto

$$H\vec{v}_i = \vec{v}_i - 2\frac{1}{2}(\vec{v}_i - \vec{e}_1) = \vec{e}_1$$

Ahora, basta con examinar la primera columna de la matriz $H\ A\ H$ (que vendrá dada por $H\ A\ H\ \vec{e}_1$) y emplear la simetría de H junto con la propiedad que se acaba de probar para obtener

$$H\ A\ H\ \vec{e}_1 = H\ A\ \underbrace{H^{-1}\ \vec{e}_1} = H\ A\ \vec{v}_i = \lambda_i H \vec{v}_i = \lambda_i \vec{e}_1$$

mostrándose así que la matriz $H\ A\ H$ tiene la estructura anunciada.

□

Como consecuencia de esta proposición se ha logrado una transformación de semejanza (recuérdese que $H = H^{-1}$) que lleva a la matriz A a una forma triangular superior por bloques, de modo que los bloques en la diagonal contendrán los autovalores de A. Puesto que el bloque superior está formado exclusivamente por λ_1, es claro que el bloque inferior (la matriz B de acuerdo con la notación de la proposición) contiene el resto de los autovalores.

Así las cosas, si por ejemplo se desease calcular un cierto número de autovalores de la matriz A con los mayores módulos, se procedería del modo siguiente (suponiendo que todos los autovalores de A tienen módulos distintos):

1. empleando el método de la potencia sobre la matriz A se obtendría el autovalor dominante, λ_1
2. empleando la técnica de deflación descrita, se construye una matriz B que contiene el resto de los autovalores (todos los autovalores de A salvo λ_1)
3. empleando el método de la potencia sobre la matriz B se obtendría el segundo autovalor dominante, λ_2
4. empleando la técnica de deflación descrita, ahora sobre la matriz B, se construye una matriz C que contiene el resto de los autovalores (todos los autovalores de A salvo λ_1 y λ_2)
5. empleando el método de la potencia sobre la matriz C se obtendría el tercer autovalor dominante, λ_3

   ...

Cabe, no obstante, hacer algunas precisiones sobre el empleo de las técnicas de deflación para el cálculo de los autovalores de una matriz.

- No se trata de una técnica diseñada para calcular todos los autovalores de una matriz. La razón está no sólo en que existen técnicas mucho más eficientes para obtener todos los autovalores de la matriz (la siguiente sección presenta una de estas técnicas), sino que además el método de deflacción presenta un grave inconveniente ligado al deterioro de las aproximaciones de los autovalores restantes. Obsérvese, en el ejemplo citado para el cálculo de varios autovalores con mayor módulo, que la primera etapa (aplicación

del método de la potencia a la matriz original A) no proporcionará más que una aproximación de λ_1 (y no su valor exacto) por lo que la matriz B obtenida en la práctica no contiene el resto de los autovalores sino una aproximación de éstos (de hecho, la estructura de la matriz $H\ A\ H$ no será estrictamente triangular superior por bloques sino que la primera columna contendrá por debajo de la diagonal valores no nulos aunque muy reducidos: del orden del error de redondeo en el sistema sobre el cual se trabaje). A continuación se obtendrá, de nuevo, una aproximación numérica del autovalor dominante de B (el cual es, a su vez, una aproximación de λ_2) y el error asociado será trasladado a la siguiente etapa de deflación. Parece claro entonces que, tras un cierto número de etapas de deflación, la acumulación de errores puede deteriorar notablemente la aproximación.

- En la práctica existe un modo de *corregir* el deterioro de las aproximaciones tras algunas etapas de deflación. Así, retomando el ejemplo anterior, en la etapa 3 se ha obtenido una aproximación del autovalor λ_2 empleando el método de la potencia sobre la matriz B. Como se ha visto, este cálculo puede originar alguna dificultad ya que los autovalores (exactos) de B no son sino una aproximación de los autovalores de A (debido a que la deflación se ha hecho necesariamente con lo único de lo que se dispone: una aproximación numérica de λ_1). Puede corregirse este error si ahora se emplea el método de la potencia inversa con decalado (empleando como decalado la aproximación de λ_2 obtenida de aplicar el método de la potencia sobre B) sobre la matriz original A para buscar el autovalor (de A) más cercano a la aproximación de λ_2 obtenida previamente. De este modo se logra una aproximación de λ_2 que contiene el error asociado al empleo del método de la potencia inversa, pero no contiene ya una contribución adicional al error originada por emplear el método de deflación con una aproximación de λ_1. Obsérvese que esta corrección no será muy costosa ya que el método de la potencia inversa con decalado al arrancar muy cerca del valor buscado sólo requerirá unas pocas iteraciones.

4.5. Método basado en la factorización QR

En esta sección se expone un método para el cálculo de todos los autovalores de una matriz que se basa en el cálculo de una cierta factorización de una matriz, por lo que se comenzará presentando esta factorización junto con algunas ideas sobre su cálculo para, a continuación, exponer el modo de utilizar esta factorización para la aproximación de los autovalores de una matriz.

Factorización QR de matrices

Se comienza con la definición de factorización QR de una matriz cuadrada.

Definición 4.2 *Sea $A \in \mathcal{M}_{n\times n}$ (de elementos reales). Se denomina factorización QR de la matriz A a la búsqueda de una pareja de matrices Q y R tales que*

Q *es ortogonal:* $Q^{-1} = Q^t$

R *es triangular superior*

$A = QR$

Si tal factorización existiese (a continuación se verá que así es para toda matriz cuadrada), proporcionaría un método sencillo para resolver sistemas de ecuaciones lineales asociados a la matriz A. En efecto, dado un sistema de ecuaciones lineales

$$Ax = b$$

basta con emplear la factorización para escribir $QRx = b$ y descomponer en los sistemas

$$Qy = b \quad Rx = y$$

que pueden ser resueltos de forma sencilla, pues para el primero basta con hacer

$$y = Q^t b$$

y, conocido y, se calcula x mediante un proceso de remonte en el sistema $Rx = y$.

En todo caso, el interés de la factorización QR no está (al menos con carácter general) en la resolución de sistemas de ecuaciones, sino como se va a ver en la construcción de transformaciones de semejanza que permitan aproximar los autovalores de una matriz.

Proposición 4.2 *Dada $A \in \mathcal{M}_{n\times n}$ (de elementos reales), existe al menos una factorización QR de la matriz A.*

Demostración: La demostración que se va a exponer es de tipo constructivo y corresponde al modo en que se calcula en la práctica esta factorización. La idea básica en la construcción de la factorización QR es la búsqueda de unas matrices H_j ortogonales tales que

$$H_{n-1}H_{n-2}\cdots H_2H_1A = R$$

con R una matriz triangular superior. Obsérvese que, de modo parecido a la factorización LU, lo que se busca es convertir la matriz A en una matriz triangular operando por etapas (de nuevo aquí en cada etapa se busca *eliminar* los términos por debajo de la diagonal en una columna de la matriz). Si tales matrices H_j existen, bastará entonces con formar Q mediante (recuérdese que estas matrices son ortogonales)

$$Q = H_1^{-1}H_2^{-1}\cdots H_{n-2}^{-1}H_{n-1}^{-1} = H_1^tH_2^t\cdots H_{n-2}^tH_{n-1}^t$$

siendo sencillo probar que Q es, a su vez, ortogonal pues

$$QQ^t = H_1^tH_2^t\cdots H_{n-2}^tH_{n-1}^tH_{n-1}H_{n-2}\cdots H_2H_1 = I$$

El resto de la demostración contiene los detalles de la construcción de las matrices H_j (que se llevará a cabo mediante matrices de Householder) y resulta bastante técnica, por lo que puede ser obviada si no se está interesado en el algoritmo de cálculo.

Emplearemos, en lo que sigue, las siguientes notaciones

$$A_1 = A$$

$$A_{k+1} = H_k A_k \quad \text{para } k = 1, 2, \ldots, n-1$$

donde la matriz A_{k+1} representa el estado de la matriz original tras k operaciones de eliminación de términos por debajo de la diagonal (como se verá, cada etapa elimina los términos por debajo de la diagonal correspondiente), de modo que tras $n-1$ etapas se obtendrá una natriz triangular superior y así $A_n = R$.

El esquema procede, en una etapa k genérica, del siguiente modo para construir la matriz H_k:

- se extrae la $k-$ésima columna de A_k descartando los elementos por encima de la diagonal, formando así un vector $\vec{a}_k \in \mathbf{R}^{n-k+1}$ (este vector contiene los elementos de A_k que se desean eliminar)

- se forma, a partir de $\vec{a}_k$, un segundo vector $\vec{v}_k$ definido mediante

  $$\vec{v}_k = \vec{a}_k \pm \|\vec{a}_k\|_2 \, \vec{e_1}$$

  donde el signo se elige de modo que el valor absoluto de la primera componente de $\vec{v}_k$ sea el mayor posible si se quieren minimizar los efectos de redondeo asociados a divisiones por números pequeños (en todo caso, debe tomarse forzosamente un signo menos si lo que se desea es obtener elementos positivos en la diagonal de la matriz R final)

- se construye la matriz de Householder asociada a $\vec{v}_k$, que denotamos mediante $H(\vec{v}_k)$

  $$H(\vec{v}_k) = I - 2\frac{\vec{v}_k \vec{v}_k^t}{\vec{v}_k^t \vec{v}_k}$$

  y es de tamaño $(n-k+1)\times(n-k+1)$ (en caso de que $\vec{v}_k$ fuese nulo, que correspondería al caso en que no es preciso eliminar términos por debajo de la diagonal porque éstos ya son nulos, se tomaría $H(\vec{v}_k) = I$)

- se forma finalmente H_k mediante

  $$H_k = \begin{pmatrix} I & 0 \\ 0 & H(\vec{v}_k) \end{pmatrix}$$

  donde I representa la matriz identidad de tamaño $(k-1)\times(k-1)$

Resulta preciso entonces, para completar la demostración, probar que

a) H_k es ortogonal

b) la matriz $A_{k+1} = H_k A_k$ preserva los ceros por debajo de la diagonal en las $k-1$ primeras columnas ya presentes en la matriz A_k

c) la matriz $A_{k+1} = H_k A_k$ tiene ceros en la columna $k-$ésima por debajo de la diagonal

Comenzando por el apartado a), se tiene

$$H_k H_k^t = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H(\vec{v}_k) \end{pmatrix} \begin{pmatrix} I_{k-1} & 0 \\ 0 & (H(\vec{v}_k))^t \end{pmatrix} = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H(\vec{v}_k)\ (H(\vec{v}_k))^t \end{pmatrix}$$

donde

$$H(\vec{v}_k)\ (H(\vec{v}_k))^t = \left(I_{n-k+1} - 2\frac{\vec{v}_k \vec{v}_k^t}{\vec{v}_k^t \vec{v}_k}\right)\left(I_{n-k+1} - 2\frac{\vec{v}_k \vec{v}_k^t}{\vec{v}_k^t \vec{v}_k}\right) = I_{n-k+1}$$

de modo que, en efecto, $H_k H_k^t = I_n$.

Para la propiedad b), obsérvese que, notando mediante $A_{k,ij}$ los bloques de la matriz A_k y observando que el bloque $A_{k,21}$ tiene todos sus elementos nulos (ya que los elementos bajo de la diagonal de las primeras $k-1$ columnas ya han sido eliminados) se tiene

$$A_{k+1} = H_k A_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H(\vec{v}_k) \end{pmatrix} \begin{pmatrix} A_{k,11} & A_{k,12} \\ 0 & A_{k,22} \end{pmatrix} = \begin{pmatrix} A_{k,11} & A_{k,12} \\ 0 & H(\vec{v}_k) A_{k,22} \end{pmatrix}$$

de modo que, en efecto, la matriz $A_{k+1} = H_k A_k$ preserva los ceros por debajo de la diagonal en las $k-1$ primeras columnas ya presentes en la matriz A_k.

Finalmente, resta ya sólo probar que además en la k-ésima columna también se han eliminado los elementos por debajo de la diagonal, tal y como asegura la propiedad c). Así, obsérvese que la columna k-ésima de la matriz $A_{k+1} = H_k A_k$ viene dada por (notando mediante $\vec{b}_k$ los elementos de la k-ésima columna de A_k que se sitúan por encima de la diagonal)

$$H_k \begin{pmatrix} \vec{b}_k \\ \vec{a}_k \end{pmatrix} = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H(\vec{v}_k) \end{pmatrix} \begin{pmatrix} \vec{b}_k \\ \vec{a}_k \end{pmatrix} = \begin{pmatrix} \vec{b}_k \\ H(\vec{v}_k)\vec{a}_k \end{pmatrix}$$

donde, por otro lado, se tendrá (repitiendo el cálculo visto en el estudio de las propiedades de las matrices de Householder)

$$H(\vec{v}_k)\vec{a}_k = \left(I - 2\frac{\vec{v}_k \vec{v}_k^t}{\vec{v}_k^t \vec{v}_k}\right)\vec{a}_k = \vec{a}_k - 2\frac{\vec{v}_k \vec{v}_k^t}{\vec{v}_k^t \vec{v}_k}\vec{a}_k = \vec{a}_k - 2\frac{\vec{v}_k^t \vec{a}_k}{\vec{v}_k^t \vec{v}_k}\vec{v}_k$$

con

$$\vec{v}_k^t \vec{a}_k = (\vec{a}_k \pm \|\vec{a}_k\|_2 \vec{e}_1)^t \vec{a}_k = \|\vec{a}_k\|_2^2 \pm \|\vec{a}_k\|_2 a_{k,1}$$

$$\vec{v}_k^t \vec{v}_k = (\vec{a}_k \pm \|\vec{a}_k\|_2 \vec{e}_1)^t (\vec{a}_k \pm \|\vec{a}_k\|_2 \, \vec{e}_1) = 2\|\vec{a}_k\|_2^2 \pm 2\|\vec{a}_k\|_2 \, a_{k,1}$$

(donde $a_{k,1}$ representa la primera componente del vector $\vec{a}_k$), de forma que

$$H(\vec{v}_k) \, \vec{a}_k = \vec{a}_k - \vec{v}_k$$

y de la definición de $\vec{v}_k$ (que coincide con $\vec{a}_k$ en todas sus componentes salvo la primera) se obtiene finalmente la propiedad b)

$$H(\vec{v}_k) \, \vec{a}_k = (\mp\|\vec{a}_k\|_2 \; 0 \; 0 \; \cdots \; 0)^t$$

□

Aproximación de autovalores basada en la factorización QR

Dada una cierta matriz $A \in \mathcal{M}_{n\times n}$ (de elementos reales), se propone el siguiento método de aproximación de autovalores basado en la factorización QR (al que algunos textos se refieren con el nombre de *algoritmo de Francis* o, más explícitamente, *algoritmo QR de Francis*)

- `Se toma` $A_1 = A$
- `para` $k = 1, 2, \ldots$ `se hace`:

  `Dada` A_k `se calcula su factorización QR` : $A_k = Q_k R_k$

  `Se construye` $A_{k+1} = R_k Q_k$

que, como se ve, genera una sucesión de matrices calculando, en cada etapa, una factorización QR.

Obsérvese que, dado el modo de construir la sucesión, todas las matrices son semejantes. En efecto, si consideramos A_{k+1} y A_k, es fácil ver que al tomar $A_{k+1} = R_k Q_k$ se tiene (recuérdese que Q_k es ortogonal):

$$Q_k A_{k+1} Q_k^t = Q_k R_k = A_k$$

De este modo, cualquier matriz de la sucesión contiene los mismos autovalores que A y el método sería interesante si éstos fuesen más fáciles de obtener en las matrices A_k (en realidad, en el límite cuando $k \to \infty$) que en la matriz original A. Esta propiedad es cierta (al menos bajo algunas hipótesis) tal y como asegura el siguiente resultado:

Proposición 4.3 *Sea una matriz $A \in \mathcal{M}_{n\times n}$ (de elementos reales) tal que sus autovalores verifiquen*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$$

Sea P tal que $A = P\Lambda P^{-1}$ donde Λ representa la matriz diagonal formada con los n autovalores. Supongamos que P admite una factorización LU.

Entonces, se tiene (denotando mediante $A_{k,ij}$ el elemento (i,j) de la matriz A_k):

- $\lim_{k\to\infty} A_{k,ii} = \lambda_i$ *para* $1 \leq i \leq n$
- $\lim_{k\to\infty} A_{k,ij} = 0$ *para* $1 \leq j < i \leq n$

Demostración: Obsérvese que la segunda parte del resultado muestra que, bajo ciertas condiciones, la sucesión de matrices A_k presenta elementos por debajo de la diagonal que tienden a cero. Así, puesto que cada matriz A_k es semejante a A (y los autovalores de una matriz triangular superior son los elementos de la diagonal) tendrá que ocurrir que los elementos sobre la diagonal en la sucesión convergerán a los autovalores. En suma, es la segunda parte de la propiedad la que debe ser probada puesto que la primera es una consecuencia de la segunda.

La demostración es bastante tediosa y no resulta fácil explicar las líneas generales de ésta. Puede en todo caso consultarse dicha demostración en el texto *Introduction to numerical linear algebra and optimisation* citado al final de este tema.

□

Cabe, en este punto, hacer algunas observaciones sobre el método y el resultado que se acaba de enunciar.

- Existen algunos resultados de convergencia con condiciones algo más débiles. Además existen algunas situaciones donde la segunda hipótesis, sobre la existencia de factorización LU para la matriz P, se verifica automáticamente (un ejemplo son las matrices de Hessenberg, lo cual resulta relevante en la práctica como se verá más tarde).
- La condición sobre la inexistencia de autovalores con iguales módulos no puede, sin embargo, ser obviada. De hecho, cuando aparecen autovalores con igual módulo (lo cual será frecuente ya que, por ejemplo, así ocurrirá cuando existan autovalores complejos) no se puede asegurar que la estructura límite de la sucesión de matrices sea triangular superior sino triangular superior por bloques, con bloques cuyo tamaño viene dado por el número de autovalores (con su multiplicidad) con igual módulo. Obsérvese que para autovalores complejos este resultado era completamente esperable ya que el método genera una sucesión de matrices con elementos reales, de modo que los elementos de la diagonal no podrían en ningún caso converger a estos autovalores. En la práctica, se observa en efecto la aparición de bloques 2×2 cuando se aplica el algoritmo (tal y como se ha descrito) a una matriz con autovalores complejos.
- Cuando, aun verificándose la hipótesis sobre la inexistencia de autovalores con iguales módulos, existen autovalores con módulos muy próximos la convergencia puede hacerse extremadamente lenta (especialmente sobre los correspondientes bloques).

Parece claro, a partir de los comentarios anteriores, que es preciso reformular el método si se desea contar con un esquema capaz de calcular todos los autovalores de una matriz con carácter general. En particular, la implementación práctica del algoritmo se hace con dos modificaciones fundamentales que se describen a continuación.

- Antes de arrancar el algoritmo se transforma la matriz A, por semejanza empleando matrices de Householder, en una matriz Hessemberg superior (matriz con elementos nulos por debajo de la primera subdiagonal inferior). El motivo de esta transformación es que el carácter Hessemberg se conserva en las matrices de la sucesión, reduciéndose considerablemente el coste computacional de cada etapa. Adicionalmente, para una matriz Hessemberg siempre puede asegurarse la existencia de una factorización LU para la matriz P, de modo que esta hipótesis deja de ser restrictiva.

- En cada etapa del método se emplea un decalado, reemplazando la matriz A_k por $A_k - s_k I$ antes de calcular la factorización y eligiendo s_k de modo adecuado para acelerar la convergencia (lo cual pasa por que dicho decalado separe los autovalores lo más posible). Pueden consultarse los detalles, por ejemplo, en el texto *Introduction to numerical linear algebra and optimisation*.

El algoritmo resultante es muy eficiente (aunque su análisis en el caso general es muy complicado) y constituye la base de la mayor parte de los códigos de aproximación de todos los autovalores de una matriz general (es el algoritmo empleado en LAPACK y, en consecuencia, en multitud de entornos, comenzando por Octave y MATLAB). Obsérvese, finalmente, que si se desea calcular los autovectores una vez que se tiene una aproximación de los autovalores, basta con una iteración del método de la potencia inversa con decalado (por cada autovalor) y se cuenta además con una factorización de la matriz para resolver con un coste reducido el sistema de ecuaciones asociado.

4.6. Códigos disponibles

En primer lugar, en lo que se refiere a códigos para el cálculo de todos los autovalores de una matriz de tamaño moderado (en el caso de grandes matrices huecas el cálculo de todos los autovalores no sólo implica unos costes computacionales desorbitados sino que, lo que es más importante, no suele presentar ningún interés práctico, estando centrado éste en el cálculo de unos pocos autovalores con alguna propiedad) los códigos más empleados son las subrutinas de la biblioteca LAPACK.

La biblioteca de funciones de LAPACK para el cálculo de autovalores tiene su origen en la biblioteca EISPACK, con unas muy notables mejoras desde el punto de vista computacional (LAPACK emplea el nivel BLAS 3 para las operaciones de bajo nivel, en tanto que EISPACK sólo emplea el nivel BLAS 1, lo que hace que sea poco eficiente especialmente en modernas arquitecturas de supercomputación). Las versiones más actuales de LAPACK (junto con una amplia documentación) pueden encontrarse en la dirección

`http://www.netlib.org/lapack/`

Por otro lado, muchos entornos de programación emplean las funciones de LAPACK. Por ejemplo, las funciones relativas al cálculo de autovalores (de hecho, todas las relativas al Álgebra Lineal Numérica) de Scilab, Octave o MATLAB están tomadas de LAPACK (aún

más, MATLAB surgió como un entorno de acceso fácil a las subrutinas de LINPACK y EISPACK).

En lo que se refiere al cálculo de (unos pocos) autovalores para grandes matrices huecas, existen multitud de códigos disponibles. Debe observarse que, como ya se ha mencionado, las técnicas para este tipo de problemas son, por muchas razones, específicas y notablemente más sofisticadas que los métodos presentados aquí. El texto de J. Dongarra citado al final del tema constituye un introducción a estas técnicas. El propio J. Dongarra mantiene una lista de referencias sobre códigos disponibles, centrada en códigos libres para problemas de autovalores asociados a grandes matrices huecas en la siguiente dirección:

`http://www.netlib.org/utk/people/JackDongarra/la-sw.html`

Un código bastante popular dentro de esta categoría es ARPACK, que puede encontrarse en la siguiente dirección

`http://www.caam.rice.edu/software/ARPACK/`

junto con una extensa documentación. Se trata de una biblioteca de funciones, escritas en FORTRAN77, que cuentan con una versión para máquinas paralelas y que actualmente se están reescribiendo en C++.

4.7. Referencias

- **P.G. Ciarlet; *Introduction to numerical linear algebra and optimisation.* Cambridge University Press, 1989.**
- **J.J. Dongarra; I.S. Duff; D.C. Sorensen; H.A. van der Vost; *Numerical Linear Algebra for High-Performance Computers.* SIAM, 1999.**
- **C. Fernández; F.J. Vázquez; J.M. Vegas; *Ecuaciones diferenciales y en diferencias. Sistemas dinámicos.* Thomson, 2003.**

Capítulo 5

Interpolación numérica

5.1. Motivación

Ya se ha abordado en la titulación (y volverá a hacerse posteriormente) el problema de la caracterización de una cierta señal a partir de un muestreo de ésta (formalmente, este problema corresponde a la aproximación de una función a partir de sus valores en unos ciertos puntos). Hay muchas razones para llevar a cabo esta operación en el dominio de la frecuencia. Se emplea así el denominado análisis armónico para caracterizar la señal mediante su contenido en frecuencias (habitualmente a través del análisis de Fourier que utiliza funciones trigonométricas).

En algunas situaciones, sin embargo, puede resultar interesante llevar a cabo la caracterización de la señal (o, con carácter más general, la aproximación de la función) directamente en el dominio temporal. Éste es el caso cuando la señal transporta poca información (que además no tiene un espectro estructurado) o, más habitualmente, cuando se trata de aproximar una función que representa alguna propiedad de un sistema que varía de forma relativamente *suave* con un cierto parámetro. Es habitual entonces usar polinomios para llevar a cabo dicha aproximación. Este tema abordará justamente las técnicas para la obtención de estos polinomios y las correspondientes propiedades de aproximación.

Cabe mencionar, finalmente, que existen técnicas que pueden considerarse *mixtas* en lo que se refiere al empleo del dominio de frecuencia y el dominio temporal. Un ejemplo de dichas técnicas son las técnicas basadas en *wavelets* que usan una descripción armónica que podríamos considerar *focalizada* en la variable temporal.

Antes de comenzar el estudio de la aproximación de funciones mediante polinomios, se tratará de justificar la razón del empleo de polinomios en dicha aproximación. Dicha razón se encuentra en el siguiente resultado

Teorema 5.1 *(Teorema de aproximación de Weierstrass)*
Sea $f \in C([a,b])$. Para cualquier $\epsilon > 0$ existe un polinomio P tal que

$$|f(x) - P(x)| \leq \epsilon \qquad \forall x \in [a, b]$$

Demostración: Consúltese el texto de Isaacson y Keller para una demostración constructiva debida a Bernstein.

□

Como se ve, basta con que la función que tratemos sea mínimamente regular (que sea, al menos, continua) para que ésta pueda ser aproximada con una precisión arbitraria. Queda por comprobar, no obstante, la eficiencia desde el punto de vista práctico de esta aproximación.

5.2. Aproximación de funciones mediante polinomios

Dada una cierta función $f : [a, b] \to \mathbf{R}$ regular nos planteamos la búsqueda de un polinomio P de un cierto grado, que suponemos prefijado, que aproxime de algún modo la función f sobre el intervalo $[a, b]$.

La forma de construir dicho polinomio (y, por lo tanto, el propio polinomio resultante) dependerá lógicamente del sentido que queramos dar al término *aproximación*. Para comenzar, hay muchas formas de definir la distancia entre f y P (algo que ya ocurría, por ejemplo, al medir la distancia entre dos puntos de $\mathbf{R}^n$) que, además, no son equivalentes (esto sí representa una notable diferencia con respecto a otras situaciones, como las distancias entre puntos de $\mathbf{R}^n$). Así, la elección de una u otra distancia llevaría a elegir un polinomio con distintas propiedades.

En la práctica, por otro lado, resulta conveniente formular de un modo lo más sencillo posible la búsqueda del polinomio de aproximación. Suponiendo que el grado de este polinomio es n, desearíamos obtener $n + 1$ condiciones a fin de fijar (a ser posible de forma única) los $n + 1$ coeficientes de dicho polinomio.

A continuación se muestran las principales alternativas en la elección del polinomio.

- Interpolación de Lagrange:

  Se eligen $n + 1$ puntos sobre el intervalo $[a, b]$, $\{x_j\}_{j=0}^n$, y se busca un polinomio $P_n(x)$ que verifique

  $$P_n(x_j) = f(x_j) \qquad \text{para } j = 0, 1, 2, \ldots, n$$

- Interpolación de Hermite:

  Se eligen m puntos sobre el intervalo $[a, b]$, $\{x_j\}_{j=0}^m$, y se busca un polinomio $P_n(x)$ que verifique

  $$P_n(x_j) = f(x_j), \qquad P_n'(x_j) = f'(x_j), \quad \ldots, \quad P_n^{(k_j}(x_j) = f^{(k_j}(x_j) \qquad \text{para } j = 0, 1, 2, \ldots, n$$

donde los índices k_j se toman de forma que

$$\sum_{j=0}^{m} k_j = n - m$$

a fin de imponer, en total, $n + 1$ condiciones.

- Interpolación de Taylor:

  Se elige un punto sobre el intervalo $[a, b]$, x_0, y se busca un polinomio $P_n(x)$ que verifique

  $$P_n(x_0) = f(x_0), \qquad P_n'(x_0) = f'(x_0), \quad \ldots, \quad P_n^{(n}(x_0) = f^{(n}(x_0)$$

- Aproximación mediante mínimos cuadrados:

  Se eligen $m+1$ puntos sobre el intervalo $[a, b]$, $\{x_j\}_{j=0}^m$, y se busca un polinomio $P_n(x)$ que haga mínima la cantidad

  $$\sum_{j=0}^{m} (P_n(x_j) - f(x_j))^2$$

Obsérvese que en los tres primeros ejemplos se impone, de modo directo, un número de condiciones igual al número de incógnitas (cada una de estas condiciones devuelve además una ecuación lineal en las incógnitas, que son los coeficientes del polinomio). En el último ejemplo, las condiciones sobre los coeficientes se obtienen caracterizando el vector de coeficientes como el punto donde se hace mínima la función (de los coeficientes)

$$\sum_{j=0}^{m} (P_n(x_j) - f(x_j))^2$$

lo que devuelve también un sistema de ecuaciones lineales en los coeficientes.

La interpolación de Taylor, a diferencia de los otros ejemplos, devuelve una aproximación *local*. Así, no cabe esperar que dicha aproximación se comporte bien en cuanto nos separemos del punto x_0. Además, queda claro que este tipo de aproximación sólo devolverá buenos resultados si la función es suficientemente regular (lo cual requiere no sólo que admita derivadas hasta un orden elevado sino además que éstas no sean grandes en valor absoluto) ya que en otro caso el término de error no será reducido.

Por lo general, la interpolación de Hermite y de Taylor tiene un reducido interés práctico. La razón (además del carácter *local* de la aproximación en el caso de Taylor) está en que suele ser raro en la práctica contar con valores de las derivadas de la función que se trata de aproximar. De hecho, como se verá en el siguiente tema, el problema suele ser más bien el contrario: dados los valores de una función sobre ciertos puntos, nos planteamos el problema de aproximar sus derivadas.

La formulación del problema de mínimos cuadrados engloba, desde luego, al problema de interpolación de Lagrange. Así, si se toma $m \leq n$ se puede conseguir que el mínimo de la función

$$\sum_{j=0}^{m} (P_n(x_j) - f(x_j))^2$$

sea exactamente cero. Para ello basta con tomar precisamente el polinomio de interpolación de Lagrange sobre esos puntos. Obsérvese que si $m < n$ pueden, de hecho, encontrarse varios polinomios con la propiedad de hacer cero la función que se trata de minimizar (fijando, por ejemplo, el valor del polinomio en otros puntos hasta completar un número de ecuaciones igual al de coeficientes que se han de determinar). En la práctica, no obstante, se tratará normalmente con casos donde $m > n$ ya que el proceso de ajuste mediante mínimos cuadrados busca un cierto polinomio que aproxime los datos medidos pero no espera que recupere exactamente esos valores en los puntos donde se toman las medidas y se tomará un número relativamente alto de medidas (en comparación con el grado del polinomio que se va a calcular) a fin de compensar los errores de medida.

Cabe mencionar, por último, que existe un enfoque alternativo de la aproximación mediante polinomios que consiste en buscar funciones que no son polinomios sobre todo el intervalo considerado sino que constituyen polinomios sobre subintervalos, prescribiendo además una cierta regularidad para la función. Denominaremos aproximación polinomial a trozos a dicha técnica.

En este tema se abordarán exclusivamente dos técnicas de aproximación de funciones mediante polinomios

- interpolación de Lagrange
- aproximación polinomial (mediante interpolación de Lagrange) a trozos

excluyendo de la presentación el resto. La razón es que algunas técnicas (como la interpolación de Hermite y de Taylor) tienen un interés práctico limitado y otras (como la aproximación mediante mínimos cuadrados) ya se ha estudiado previamente.

5.3. Interpolación de Lagrange

En esta sección se abordará en primer lugar la existencia y unicidad del polinomio de interpolación de Lagrange, junto con algunas alternativas en la construcción de dicho polinomio. Posteriormente se obtendrá una expresión del error de interpolación (como se verá, esta expresión resultará útil también para otras técnicas numéricas basadas en la interpolación, como la derivación y la integración numéricas que se estudiarán posteriormente) y se ilustrará el comportamiento del polinomio de interpolación de Lagrange en diferentes situaciones. Este último apartado servirá para motivar la siguiente sección, dedicada al estudio de la interpolación polinomial a trozos.

5.3.1. Existencia y construcción del polinomio de interpolación de Lagrange

En la sección anterior ya se observó que la interpolación de Lagrange prescribe un número de condiciones (que, de hecho, toman la forma de ecuaciones lineales sobre los coeficientes)

igual al número de coeficientes que se deben determinar. Queda, no obstante, por demostrar que dicho problema admite solución y ésta es única. En particular se tiene el siguiente resultado.

Teorema 5.2 *(Existencia y unicidad del polinomio de interpolación de Lagrange) Sea una tabla de puntos* $\{(x_k, y_k)\}_{k=0}^n$ *con* $x_0 < x_1 < x_2 < \cdots < x_n$. *Entonces existe un y sólo un polinomio de grado n,* $P_n(x)$, *tal que*

$$P_n(x_k) = y_k \qquad \text{para } \; k = 0, 1, 2, \ldots, n$$

Demostración: Se prueba, en primer lugar, que existe dicho polinomio. Para ello, se consideran las siguientes funciones (para $i = 0, 1, 2, \ldots, n$):

$$L_i(x) = \prod_{k=0\,,\,k\neq i}^{n} \frac{x - x_k}{x_i - x_k}$$

que se denominan (la razón se verá a continuación) *funciones de base de interpolación de Lagrange*.

Es claro que dichas funciones son polinomios de grado n y que verifican además

$$L_i(x_j) = \begin{cases} 1 & \text{si} \quad i = j \\ 0 & \text{si} \quad i \neq j \end{cases}$$

Por lo tanto, el polinomio de grado n definido mediante

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

constituye el polinomio de interpolación de Lagrange buscado.

En cuanto a la unicidad de dicho polinomio, es fácil concluir que forzosamente ha de tener lugar mediante reducción al absurdo. Así, supóngase que existen dos polinomios de grado n que resuelven el problema de interpolación de Lagrange, $P_n^1(x)$ y $P_n^2(x)$. Es claro que $P(x) = P_n^1(x) - P_n^2(x)$ también es un polinomio de grado n, que además se anula sobre los puntos $\{x_i\}_{i=0}^n$ puesto que

$$P(x_i) = P_n^1(x_i) - P_n^2(x_i) = y_i - y_i = 0 \qquad \text{para } i = 1, 2, \ldots, n$$

En consecuencia P es idénticamente nulo (ya que se trata de un polinomio de grado n con $n + 1$ raíces) por lo que P_n^1 y P_n^2 han de ser iguales.

□

Nota 5.1 *Existe en todo caso un modo más directo de obtener el polinomio de interpolación de Lagrange resolviendo el sistema de ecuaciones planteado en los coeficientes del polinomio. Así, notando*

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

se deberá resolver

$$\begin{pmatrix} x_0^n & x_0^{n-1} & ... & x_0 & 1 \\ x_1^n & x_1^{n-1} & ... & x_1 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & ... & x_n & 1 \end{pmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

cuya matriz recibe el nombre de matriz de Vandermonde.

Este modo de construir el polinomio de interpolación de Lagrange ya permite poner de manifiesto una de las dificultades de la interpolación de Lagrange con polinomios de grado elevado: el condicionamiento de la matriz empeora notablemente conforme aumenta el número de puntos. De esta forma, pequeñas perturbaciones en los valores $\{y_i\}_{i=0}^n$ *pueden conducir a grandes variaciones en los coeficientes* $\{a_i\}_{i=0}^n$ *del polinomio (y, en consecuencia, obtener polinomios muy distintos). Obsérvese que, en la práctica, existen numerosas razones por las cuales podrían producirse perturbaciones en los valores* $\{y_i\}_{i=0}^n$ *(por ejemplo, errores en las medidas o el truncamiento de los valores al ser representados), de modo que esta dificultad asociada a la interpolación con polinomios de grado elevado debe ser tenida en cuenta. Puede ampliarse el estudio del condicionamiento del problema de interpolación de Lagrange, por ejemplo, en el texto de A. Quarteroni* et al.

□

Existe aún otra alternativa para la construcción del polinomio de interpolación de Lagrange conocida como fórmula o esquema de Newton, que presenta dos ventajas fundamentales desde el punto de vista computacional

- su coste computacional es más reducido (crece con el cuadrado del número de puntos en tanto que la resolución del sistema de ecuaciones mediante un método directo del sistema asociado a la matriz de Vandermonde lo haría con el cubo de este número)
- la adición de nuevos nodos de interpolación no obliga a rehacer todos los cálculos, sino un número reducido de ellos

Desde luego, la primera ventaja tiene un interés muy reducido. Como se ha comentado previamente y se volverá a examinar con posterioridad, la interpolación con un número elevado de puntos presenta graves dificultades y rara vez se empleará en la práctica. No ocurre lo mismo con la segunda ventaja pues es claro que, en determinadas situaciones, puede resultar interesante incluir puntos adicionales para mejorar la aproximación.

La idea básica del esquema o fórmula de Newton es la siguiente: supóngase que se dispone del polinomio de interpolación de Lagrange $P_{n-1}(x)$ asociado a la tabla $\{(x_i, y_i)\}_{i=0}^{n-1}$ y se desea añadir un nuevo punto a esa tabla, (x_n, y_n) y construir un nuevo polinomio de interpolación, $P_n(x)$, de la tabla ampliada $\{(x_i, y_i)\}_{i=0}^n$. Puede entonces pensarse en formar $P_n(x)$ de la forma siguiente

$$P_n(x) = P_{n-1}(x) + c_n \prod_{j=0}^{n-1}(x - x_j)$$

donde es inmediato comprobar que

$$P_n(x_i) = P_{n-1}(x_i) = y_i \qquad \text{para } i = 0, 1, 2, \ldots, n-1$$

de modo que sólo será preciso imponer

$$P_n(x_n) = y_n$$

lo que conduce a tomar c_n como

$$c_n = \frac{y_n - P_{n-1}(x_n)}{\prod_{j=0}^{n-1}(x_n - x_j)}$$

La idea de las fórmulas de Newton es entonces, dada una cierta tabla $\{(x_i, y_i)\}_{i=0}^n$, construir $P_n(x)$ de forma recursiva tomando (x_0, y_0) y añadiendo puntos de uno en uno, en forma ordenada, hasta completar la tabla. Si los puntos $\{x_i\}_{i=0}^n$ están ordenados de forma creciente, dicha técnica suele denominarse *fórmula de Newton progresiva* y se denomina, como resulta esperable, *fórmula de Newton regresiva* a aquella que toma en primer lugar (x_n, y_n) y añade puntos de forma ordenada hasta completar la tabla.

Nota 5.2 *Si se considera la denominada* fórmula de Newton progresiva *y se construyen los primeros términos se obtiene:*

$$c_0 = y_0$$

$$c_1 = \frac{y_1 - y_0}{x_1 - x_0}$$

$$c_2 = \frac{y_2 - c_0 - c_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} = \frac{\dfrac{y_1 - y_0}{x_1 - x_0} - \dfrac{y_2 - y_1}{x_2 - x_1}}{x_2 - x_0}$$

$$\ldots$$

lo que sugiere generar los coeficientes a partir de un cálculo recurrente de las denominadas diferencias divididas. *Consúltese en cualquier texto de métodos numéricos, por ejemplo en el texto de Kincaid y Cheney, la definición recurrente de las diferencias divididas y el modo de organizar los cálculos para obtener los coeficientes* $\{c_i\}_{i=0}^n$.

□

5.3.2. Error de aproximación en la interpolación de Lagrange

Se va a obtener, en primer lugar, una expresión del error de aproximación cometido en la interpolación de Lagrange.

Teorema 5.3 *(Expresión del error de aproximación)*
Sea $f \in C^{n+1}([a,b])$ y sea P_n el polinomio de interpolación de Lagrange de f sobre los nodos $\{x_j\}_{j=0}^n$ en $[a,b]$. Entonces, para cada $x \in [a,b]$ existe $\xi_x \in [a,b]$ tal que

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1}(\xi_x) \prod_{j=0}^{n} (x - x_j)$$

Demostración: Se supondrá en lo que sigue que $x \neq x_j \ \ \forall j \in \{0,1,2,\ldots,n\}$ pues, en otro caso, la expresión del error se verifica de modo trivial.

Se define una función w de la forma

$$w(t) = \prod_{j=0}^{n} (t - x_j)$$

y (con x ya fijado) un número c mediante

$$c = \frac{f(x) - P_n(x)}{w(x)}$$

Obérvese que, puesto que $x \neq x_j \ \ \forall j \in \{0,1,2,\ldots,n\}$, se tiene que $w(x) \neq 0$.

Definimos además una segunda función ϕ mediante

$$\phi(t) = f(t) - P_n(t) - c\, w(t)$$

donde, puesto que f es de clase $C^{n+1}([a,b])$ y que P_n y w son (como polinomios) funciones de clase $C^{\infty}([a,b])$, es claro ver que ϕ es de clase $C^{n+1}([a,b])$.

Por otro lado, es fácil probar que ϕ presenta $n+2$ raíces puesto que:

$$\phi(x_j) = (f(x_j) - P_n(x_j)) - c\, w(x_j) = 0 - 0 = 0 \qquad \forall j \in \{0,1,2,\ldots,n\}$$

$$\phi(x) = (f(x) - P_n(x)) - c\, w(x) = 0$$

de modo que, por aplicación del teorema de Rolle, ϕ' tiene $n+1$ raíces, ϕ'' tiene n raíces ... y, finalmente, $\phi^{(n+1}$ tiene 1 raiz.

Sea entonces ξ_x la raiz de $\phi^{(n+1}$ cuya existencia se acaba de asegurar. Es claro que

$$\phi^{(n+1}(\xi_x) = f^{(n+1}(\xi_x) - P_n^{(n+1}(\xi_x) - c\, w^{(n+1}(\xi_x) = f^{(n+1}(\xi_x) - 0 - c\,(n+1)!$$

de donde se concluye

$$c = \frac{1}{(n+1)!} f^{(n+1}(\xi_x)$$

y llevando dicha igualdad a la defínción de c se deduce el resultado enunciado.

□

Del resultado anterior, que hace depender el error de aproximación del producto de tres factores de acuerdo con la expresión

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1}(\xi_x) \prod_{j=0}^{n} (x - x_j)$$

se desprenden varias conclusiones de gran relevancia desde el punto de vista práctico. Las principales son las siguientes:

- Si consideramos fijos los nodos de interpolación, $\{x_j\}_{j=0}^n$, (de modo que el primero y el último de los factores quedan también fijados) y abordamos la interpolación sobre dichos nodos de diferentes funciones es claro que la bondad de la aproximación que se consiga depende de la regularidad de la función que se desea aproximar. Así, si la función es regular, entendiendo como función regular aquella para la cual los valores absolutos de sus derivadas de orden superior no son muy grandes, el segundo factor de la expresión del error de interpolación no será grande. Si, por el contrario, la función no es regular, porque no tiene derivadas de un orden elevado, o porque éstas toman valores absolutos muy grandes, la expresión anterior no será aplicable (ya que exige que la función admita derivadas de orden superior) o proporcionará una cota tan elevada que no garantiza una buena aproximación.

- Suponiendo ahora que la función es regular (en el sentido anterior) y admitiendo que el último factor no puede ser grande (como se verá puede, de hecho, hacerse pequeño), es también claro que cabe esperar una mejora de la aproximación de f mediante un polinomio de interpolación de Lagrange P_n aumentando el número de nodos de interpolación pues el primer factor decrece rápidamente conforme aumenta n.

- Si ahora suponemos que el número de nodos de interpolación es fijo pero no su posición parece natural plantearse cuál debe ser la adecuada elección de la posición de estos nodos puesto que de ella depende el último de los factores del error de aproximación. Aunque podría también plantearse la búsqueda de una adecuada distribución de los nodos para lograr una buena aproximación de una determinada función, nos plantearemos posteriormente la búsqueda de una distribución de nodos que haga mínimo (para n fijo) el tercer factor de la expresión del error.

Como ilustración de las observaciones anteriores se van a considerar a continuación varios ejemplos. El primero de ellos ilustra la interpolación de una función (muy) regular.

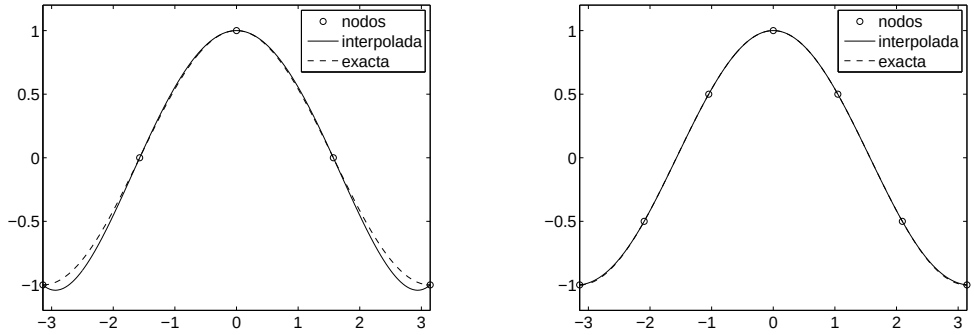

Figura 5.1: Interpolación de Lagrange de la función $f(x) = \cos(x)$ sobre el intervalo $[-\pi, \pi]$ con 5 (izquierda) y 7 (derecha) nodos equiespaciados

Ejemplo 5.1 *Se considera la aproximación mediante un polinomio de interpolación de Lagrange de la función* $f : [-\pi, \pi] \to \mathbf{R}$ *definida mediante* $f(x) = \cos(x)$*. Es claro que dicha función es regular (en el sentido definido anteriormente) y cabe por lo tanto esperar una rápida convergencia (conforme aumenta el número de nodos de interpolación) del polinomio de interpolación de Lagrange.*

La figura 5.1 muestra los polinomios de interpolación obtenidos para dos conjuntos de nodos equiespaciados, correspondientes a $n = 4$ *y* $n = 6$*. Cabe observar que, como se esperaba, basta un número reducido de nodos para lograr una buena aproximación de la función (con* $n = 6$ *apenas se perciben diferencias entre las representaciones de la función y su interpolada).*

Además, en la figura 5.1 puede también observarse que para $n = 4$ *las diferencias entre la función y su interpolada son mayores cerca de los extremos. Como se verá en otros ejemplos, esta situación es frecuente y, en cierto modo, aparece recogida en la expresión de error deducida anteriormente ya que el último factor contiene el producto de las distancias desde el punto donde se evalúa el error hasta cada uno de los nodos de interpolación, de modo que tomará valores más elevados cerca de los extremos del intervalo al ser mayores las distancias desde éstos (lo que resultará más marcado cuanto mayor sea el número de puntos). La figura 5.2 muestra precisamente el valor de ese factor para* $n = 4$ *y* $n = 6$*.*

□

A continuación se considera la interpolación de una función no regular (en el sentido anterior).

Ejemplo 5.2 *Sea la función* $f : [-3, 3] \to \mathbf{R}$ *definida mediante* $f(x) = \dfrac{1}{1 + x^2}$*. Obsérvese que para esta función no se tiene una acotación uniforme de sus derivadas de orden superior y, en consecuencia, no es tan claro que el aumento del número de nodos de interpolación*

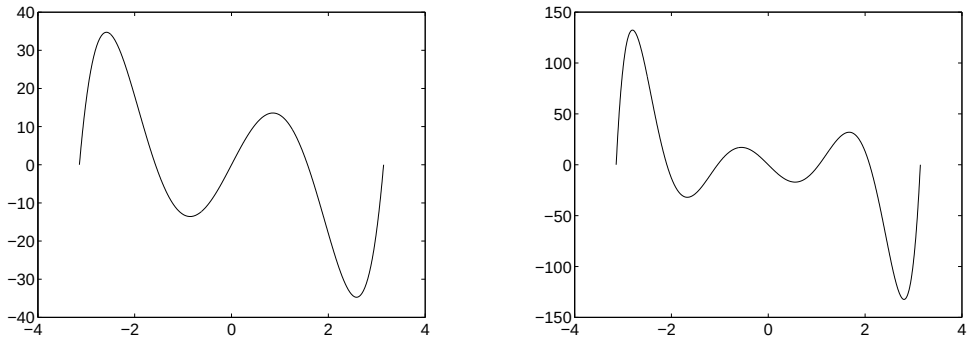

Figura 5.2: Representación de la función $\prod_{j=0}^{n}(x-x_j)$ sobre el intervalo $[-\pi,\pi]$ con 5 (izquierda) y 7 (derecha) nodos equiespaciados

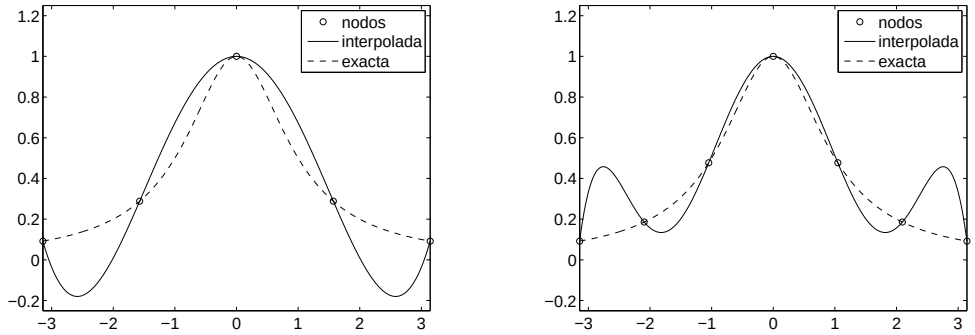

Figura 5.3: Interpolación de Lagrange de la función $f(x)=\dfrac{1}{1+x^2}$ sobre el intervalo $[-3,3]$ con 5 (izquierda) y 7 (derecha) nodos equiespaciados

lleve a una rápida mejora de la aproximación.

La figura 5.3 muestra la función y su interpolación de Lagrange (sobre nodos equiespaciados) para $n=4$ y $n=6$. Obsérvese que la aproximación es notablemente peor que la correspondiente a la función $f(x)=\cos(x)$ descrita en el ejemplo anterior. Puede, por otro lado, comprobarse que el aumento del número de nodos de interpolación mejora la aproximación en el centro del intervalo pero no ocurre lo mismo cerca de los extremos del intervalo, como se oberva en la figura 5.4, que muestra los resultados correspondientes a $n=10$ y $n=20$. Recuérdese además que el aumento del número de nodos provoca un deterioro del condicionamiento del problema de cálculo de los coeficientes del polinomio de interpolación.

□

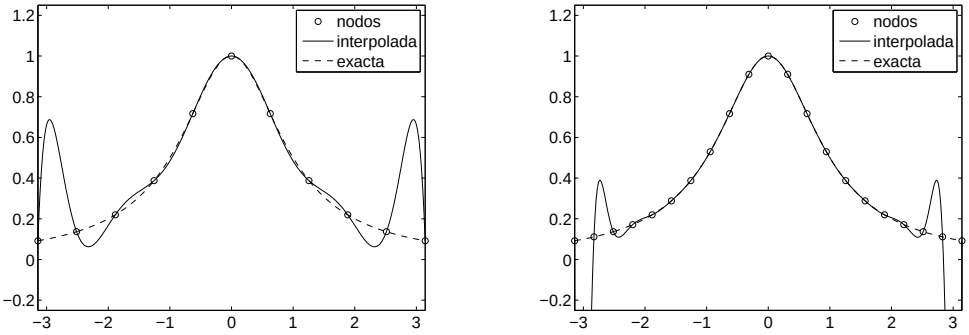

Figura 5.4: Interpolación de Lagrange de la función $f(x) = \dfrac{1}{1+x^2}$ sobre el intervalo $[-\pi, \pi]$ con 11 (izquierda) y 21 (derecha) nodos equiespaciados (el valor mínimo del polinomio de interpolación con 21 nodos es aproximadamente -3.76)

Si consideramos la aproximación mediante interpolación de Lagrange de una cierta función, sólo cabe controlar la calidad de la aproximación mediante el número de nodos de interpolación y, una vez fijado dicho número, su localización sobre el intervalo. En los ejemplos anteriores se ha considerado siempre un conjunto de nodos equiespaciados. Aunque ciertamente hay situaciones donde dicho reparto de nodos es prácticamente obligatorio (por ejemplo, si los nodos de interpolación corresponden a un muestreo temporal de una señal y resulta muy difícil configurar los equipos para muestrear sobre instantes de tiempo arbitrariamente elegidos) en muchas otras no lo es y en esos casos, como se va a ver, no hay ninguna razón para elegir los nodos de forma equiespaciada.

A la vista de la estimación de error obtenida en el teorema 5.3 parece claro que, si fuese posible elegir los nodos de interpolación sin ninguna restricción, éstos deberían tomarse de modo que se minimizase el último factor de la expresión de la cota de error:

$$\prod_{j=0}^{n}(x - x_j)$$

Así, suponiendo prefijado el grado del polinomio de interpolación ,n, se plantea la búsqueda de $n+1$ puntos, $\{x_j\}_{j=0}^{n}$, para los cuáles la función

$$\Phi(x) = \prod_{j=0}^{n}(x - x_j)$$

sea lo más pequeña posible sobre el intervalo $[a, b]$. La respuesta a esta cuestión, cuando para medir el *tamaño* de la función se emplea la norma infinito (el mayor valor absoluto alcanzado por Φ), son los denominados *nodos de Chebyshev*.

Los *nodos de Chebyshev* (que representan los ceros de los polinomios del mismo nombre) para el intervalo $[-1, 1]$ son los puntos

$$\hat{x}_j = \cos\left(\frac{2j+1}{n+1}\,\frac{\pi}{2}\right) \qquad j = 0, 1, 2, \ldots, n$$

en tanto que para un intervalo $[a, b]$ cualquiera se obtienen a partir de éstos mediante

$$x_j = \frac{a+b}{2} + \frac{b-a}{2}\,\hat{x}_j \qquad j = 0, 1, 2, \ldots, n$$

Nota 5.3 *Para los nodos de Chebyshev se obtiene la siguiente expresión para la norma infinito de la función* Φ

$$\max_{a\le x\le b} |\Phi(x)| = \frac{1}{2^n}\left(\frac{b-a}{2}\right)^{n+1}$$

lo que permite completar, cuando se emplean estos nodos, la estimación del error de interpolación dada por el teorema 5.3.

En la figura 5.5 se muestra la función $\Phi(x) = \prod_{j=0}^{n}(x - x_j)$ para los nodos de Chebyshev, con $n = 4$ y $n = 6$, sobre el intervalo $[-\pi, \pi]$. Obsérvese que si se compara esta figura con la correspondiente para conjuntos de nodos equiespaciados (figura 5.2) se comprueba que

- la función toma, en efecto, valores más reducidos para el caso de los nodos de Chebyshev, asegurando de este modo una cota de error más pequeña y, en consecuencia, se esperan mejores resultados de interpolación
- a diferencia de lo que ocurre con los nodos equiespaciados, la función Φ asociada a los nodos de Chebyshev mantiene órdenes de magnitud semejantes en todo el intervalo (en el caso de los nodos equiespaciados se obervaba que esta función crecía sensiblemente cerca de los extremos lo que conducía a peores resultados de interpolación en esta zona), por lo que ahora no cabe esperar un grave deterioro de la interpolación cerca de los extremos

Ejemplo 5.3 *Se retoma ahora la función de Runge,* $f(x) = \frac{1}{1+x^2}$*, para comprobar sobre ella las mejoras a las que conduce el empleo de nodos de Chebyshev en la interpolación de Lagrange.*

En la figura 5.6 se muestran los resultados de la interpolación de la funciónde Runge sobre el intervalo $[-\pi, \pi]$ *para* $n = 4$*,* 6*,* 10 *y* 20*. Como se puede comprobar, los resultados son notablemente mejores que los correspondientes a la interpolación con nodos equiespaciados (representados en las figuras 5.3 y 5.4). En particular, las gráficas sugieren que puede obtenerse una buena aproximación en todo el intervalo aumentando suficientemente el número de nodos de integración.*

En cualquier caso, el aumento del número de nodos genera (al igual que ocurre, de hecho con mayor gravedad, en el caso de emplear nodos equiespaciados) una dificultad adicional que

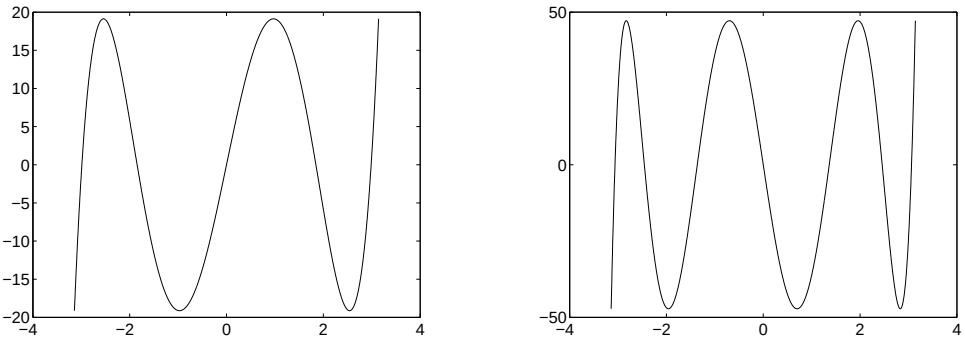

Figura 5.5: Representación de la función $\prod_{j=0}^{n}(x - x_j)$ sobre el intervalo $[-\pi, \pi]$ con 5 (izquierda) y 7 (derecha) nodos de Chebyshev

es el problema del mal condicionamiento del sistema de ecuaciones que se ha de resolver para obtener el polinomio de interpolación, reflejando el mal condicionamiento del propio problema de interpolación (pueden consultarse más detalles sobre el condicionamiento del problema de interpolación en los textos de Quarteroni et al. o de Isaacson y Keller). Esta dificultad debe ser tenida muy en cuenta porque, de este modo, pequeños errores en la evaluación de la función (o muestreo de los datos, que pueden estar además contaminados con ruido) puede conducir a interpolantes que se separen mucho de la función.

Nota 5.4 *El ejemplo anterior sugiere que, al menos en ausencia de errores de medida y redondeo, la interpolación de funciones con derivadas continuas hasta algún cierto orden puede hacerse mediante interpolación de Lagrange con nodos de Chebyshev garantizándose la convergencia conforme aumenta el número de nodos. En efecto se dispone de un resultado en ese sentido aunque problemas como el mal condicionamiento aconsejen no emplear este tipo de aproximación. En particular se tiene (consultar, por ejemplo, el texto de Issacson y Keller) que, dada un cierta función f de clase $C^2([-1, +1])$ y P_n su polinomio de interpolación de Lagrange empleando los nodos de Chebyshev,*

$$|f(x) - P_n(x)| = \mathcal{O}(\frac{1}{\sqrt{n}})$$

conforme n tiende a infinito.

A modo de conclusión del breve estudio de la interpolación de Lagrange aquí realizado cabe mencionar que dicha técnica de aproximación resulta eficiente sólo en la medida en que trata de aproximar funciones bastante regulares sobre intervalos reducidos (en el sentido en que la función no varía de forma muy drástica sobre dicho intervalo, de modo que las derivadas no pueden alcanzar valores absolutos muy grandes). En otros casos esta técnica se encontrará con diversas dificultades; así, por ejemplo, será preciso considerar un número elevado de nodos para mejorar la aproximación y esto conducirá a un mal condicionamiento

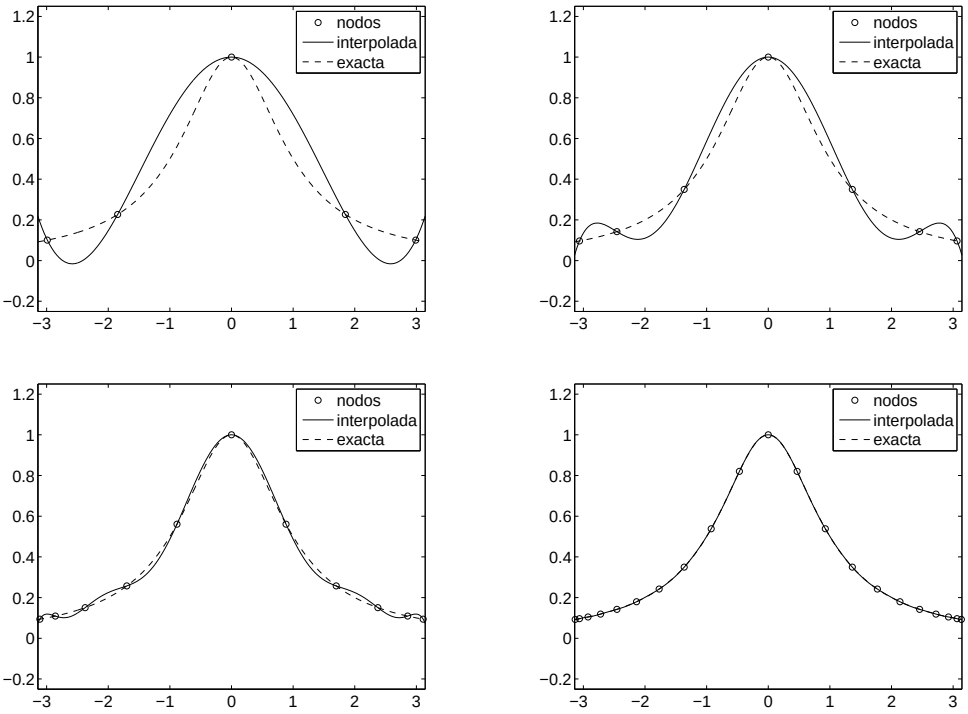

Figura 5.6: Interpolación de Lagrange de la función $f(x) = \frac{1}{1+x^2}$ con 5, 7, 11 y 21 (de izquierda a derecha y de arriba a abajo) nodos de Chebyshev

de los problemas de cálculo. La siguiente sección muestra una técnica más eficiente para la aproximación de funciones en esos casos (que serán, por otra parte, los más habituales).

5.4. Aproximación polinómica a trozos

Como se acaba de ver, la interpolación polinómica de funciones mediante polinomios globales no ofrece, por lo general, buenos resultados cuando se trata de interpolar una función sobre un intervalo amplio, pues un aumento del número de nodos de interpolación conduce (al margen de los problemas de condicionamiento) a polinomios de interpolación oscilantes.

La respuesta a estos problemas pasa por *controlar* el grado de los polinomios, manteniendo éste bajo a fin de evitar las oscilaciones del polinomio de interpolación. Así, se dividirá el intervalo completo donde se desea interpolar una cierta función en subintervalos, interpolando la función con un polinomio de grado relativamente bajo sobre cada subintervalo. Adicionalmente, se pedirá que el interpolante (definido como una función a trozos) verifique unas ciertas condiciones de regularidad globales (de modo que al *pegar* los trozos se tenga la continuidad de la función y, quizás, de alguna de sus derivadas).

En principio existen muchas posibilidades de interpolación polinómica a trozos, pero en la práctica sólo se emplean dos

- interpolación lineal a trozos
- interpolación cúbica a trozos (también denominado interpolación mediante *splines* cúbicos)

5.4.1. Interpolación lineal a trozos

Se trata de la aproximación polinómica a trozos más elemental. Dada una tabla de valores $\{(x_k, y_k)\}_{k=0}^{n}$ a interpolar, se toma sobre cada intervalo (x_{k-1}, x_k) el interpolante lineal asociado a los valores $\{y_{k-1}, y_k\}$. Es claro que dicho interpolante está bien definido (sobre cada subintervalo se plantea un problema de interpolación de Lagrange con un polinomio de grado uno, que como ya sabemos admite una y sólo una solución) y genera una función continua (pues los valores en los nodos intermedios $\{x_k\}_{k=1}^{n-1}$ son los mismos para los dos subintervalos que tiene a cada lado) pero no se garantizará la continuidad de las derivadas primeras en los nodos intermedios (salvo en el raro caso donde los interpolantes lineales coincidan exactamente en ambos subintervalos). Es claro también que al manejar polinomios de primer grado, no hay ninguna posibilidad de que la función de interpolación contenga oscilaciones (salvo, lógicamente, las que presenten los propios datos).

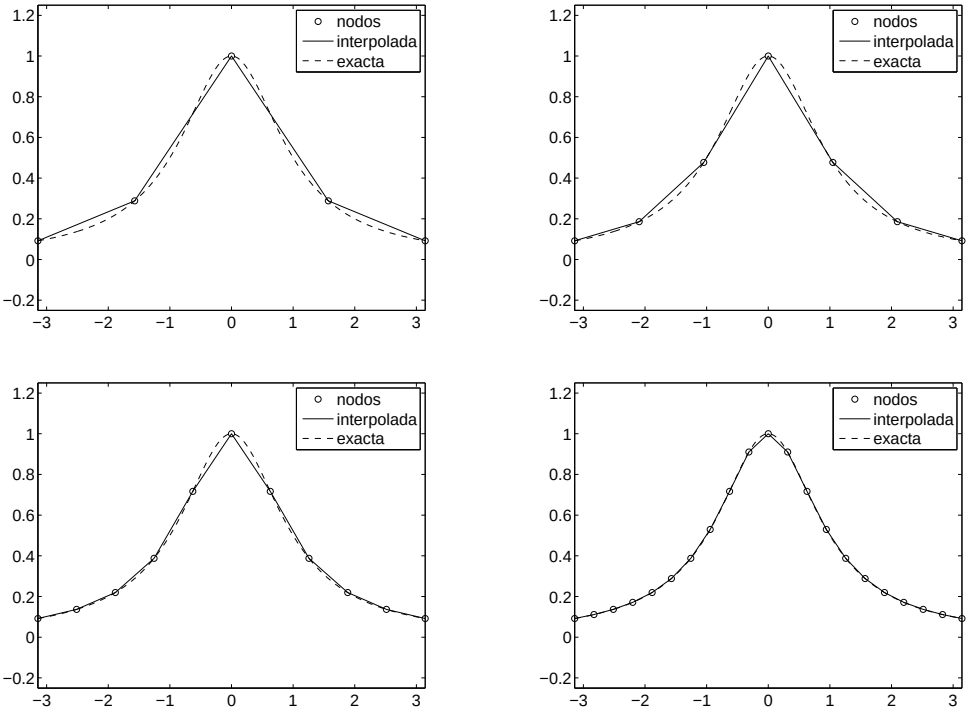

Figura 5.7: Interpolación lineal a trozos de la función $f(x) = \frac{1}{1+x^2}$ con 5, 7, 11 y 21 (de izquierda a derecha y de arriba a abajo) nodos

Ejemplo 5.4 *En la figura 5.7 se muestra los resultados de la interpolación lineal a trozos de la función de Runge,* $f(x) = \frac{1}{1+x^2}$*, sobre el intervalo* $[-\pi, \pi]$ *empleando* $n = 4$*,* 6*,* 10 *y* 20*.*

El ejemplo anterior ya ilustra la necesidad de emplear un número relativamente elevado de nodos para tener una adecuada representación de la función que se trata de interpolar. Resulta, en todo caso, sencillo obtener una estimación del error cometido en la interpolación lineal a trozos ya que, como se ha comentado, el problema se reduce a una interpolación lineal (independiente) sobre cada subintervalo. Así, retomando los resultados obtenidos anteriormente, se llega a la siguiente propiedad.

Teorema 5.4 *(Error de aproximación en la interpolación lineal a trozos)*
Sea $f \in C^2([a,b])$ *y sea* $P : [a,b] \to \mathbf{R}$ *la función de interpolación lineal a trozos asociada a* f *sobre los nodos (ordenados)* $\{x_k\}_{k=0}^n$*. Entonces se tiene*

$$|f(x) - P(x)| \leq \frac{1}{8} \max_{a \leq \eta \leq b} |f''(\eta)|\, h^2$$

donde $h = \max_{1 \leq i \leq n} (x_i - x_{i-1})$*.*

Demostración: Inmediata a partir del resultado de estimación del error de aproximación en la interpolación de Lagrange (teorema 5.3) aplicado a cada subintervalo.

□

La interpolación lineal a trozos verifica entonces uno de los requerimientos prácticos de la función de interpolación (la ausencia de oscilaciones espúreas). Sin embargo, también se ha comprobado que presenta un grave inconveniente: aunque la función que se trate de interpolar sea muy regular, no se conseguirá ninguna regularidad adicional de la función de interpolación, que simplemente será continua pero presentará derivadas discontinuas. Se hace necesario entonces contar con alguna alternativa (que pasará necesariamente por emplear polinomios de grado más elevado) para obtener funciones de interpolación más regulares (globalmente).

Finalmente, cabe observar una limitación adicional de la interpolación lineal a trozos: aún cuando dicha interpolación pueda resultar adecuada (a condición de emplear un número suficientemente elevado de nodos) cuando se pretende aproximar los valores de la función que se interpola, es claro que la interpolación lineal a trozos no produce buenos resultados cuando se desea recuperar además los valores de las derivadas de la función interpolada (obsérvese que las derivadas primeras de la función de interpolación devuelven una función constante sobre cada intervalo, lo cual constituye una aproximación bastante pobre).

5.4.2. Interpolación cúbica a trozos (spline cúbicos)

Como ya se ha mencionado en la sección anterior, si se desea recuperar una función de interpolación con cierta regularidad se hace preciso considerar la interpolación sobre cada subintervalo con un polinomio de grado más elevado. Una alternativa, muy empleada en la práctica, es considerar polinomios de grado tres sobre cada subintervalo. Las funciones de interpolación resultantes se denominan también *spline* cúbicos (que tienen su origen en el proceso mecánico empleado desde antiguo para lograr piezas con formas suaves).

Así, dada una tabla de valores $\{(x_k, y_k)\}_{k=0}^n$ (donde los nodos $\{x_k\}_{k=0}^n$ se suponen ordenados) se busca una función $S : [a, b] \to \mathbf{R}$ con las propiedades siguientes

(a) S es un polinomio de grado 3 sobre cada subintervalo $[x_i, x_{i+1}]$

(b) $S(x_i) = y_i$ sobre cada nodo x_i

(c) $S \in C^2([x_0, x_n])$

En primer lugar, es preciso saber si el número de condiciones impuestas por las propiedades no supera el número de coeficientes que es preciso fijar para determinar por completo la función S. En particular, dado que es preciso definir un polinomio de grado 3 sobre cada uno de los n subintervalos, se tienen $4n$ coeficientes libres. Por otro lado, se han de fijar

- $2n$ condiciones de la propiedad (b) (condiciones de interpolación sobre los dos nodos de cada uno de los n subintervalos)

- $2(n-1)$ condiciones de la propiedad (c) (al imponer la continuidad de las derivadas primera y segunda de la función S sobre cada uno de los $n-1$ nodos intermedios)

En suma, se obtiene que resultan $4n-2$ condiciones para fijar $4n$ coeficientes, por lo que cabe esperar que no sólo sea posible determinar una función S con las propiedades mencionadas, sino que además quedarán dos parámetros libres que podrán ser fijados mediante condiciones adicionales.

Resulta habitual emplear, como pareja de condiciones adicionales, alguna de las siguientes

- $S''(x_0) = S''(x_n) = 0$
- $S'(x_0) = y'_0$ y $S'(x_n) = y'_n$ para y'_0 e y'_n dados
- S''' continua en x_1 y x_{n-1}

La tercera de las alternativas expuestas (conocida como *not–a–knot*) es la empleada con más frecuencia y conduce, como es fácil comprobar, a que los polinomios de interpolación coincidan sobre los intervalos $[x_0, x_1]$ y $[x_2, x_3]$ por un lado y los intervalos $[x_{n-2}, x_{n-1}]$ y $[x_{n-1}, x_n]$ por otro (de ahí el nombre que reciben estas condiciones).

En particular, si se emplea la condición *not–a–knot* para determinar la función de interpolación cúbica a trozos $S(x)$ se obtiene el siguiente resultado

Teorema 5.5 *(Error de aproximación en interpolación mediante spline cúbico)*
Sea $f \in C^4([a,b])$ y sea $\{x_k\}_{k=0}^n$ un conjunto (ordenado) de nodos contenidos en el intervalo $[a,b]$.
Entonces, existe un spline cúbico que interpola la tabla $\{(x_k, f(x_k))\}_{k=0}^n$ y verifica la condición not–a–knot.
Además, denotando $h = \max_{1\le i\le n}(x_i - x_{i-1})$, existe una constante C_0 tal que

$$|f(x) - S(x)| \le C_0\, h^4 \max_{x_0\le\eta\le x_n} |f^{(iv}(\eta)|\,, \quad \forall x \in [x_0, x_n]$$

y unas constantes C_1 y C_2 tales que

$$|f^{(i}(x) - S^{(i}(x)| \le C_i\, h^{4-i} \max_{x_0\le\eta\le x_n} |f^{(iv}(\eta)|\,, \quad \forall x \in [x_0, x_n]$$

Demostración: Consúltese, por ejemplo, la obra de C. de Boor.

□

El resultado anterior muestra las buenas propiedades de aproximación de los spline cúbicos, no sólo en relación con los valores de la función sino también de sus derivadas primeras y segundas (esta cuestión será retomada en el próximo tema). En particular, obsérvese que (si f es suficientemente regular) el error depende de la potencia cuarta de h, en tanto que con interpolantes lineales a trozos se tenía una dependencia de la potencia segunda de h. Puesto que h depende a su vez del inverso del número de nodos de interpolación, es claro que el empleo de la aproximación mediante spline no exige (a diferencia de lo que ocurría con la interpolación lineal a trozos) un número muy elevado de nodos para obtener una buena aproximación (que será, además, siempre regular).

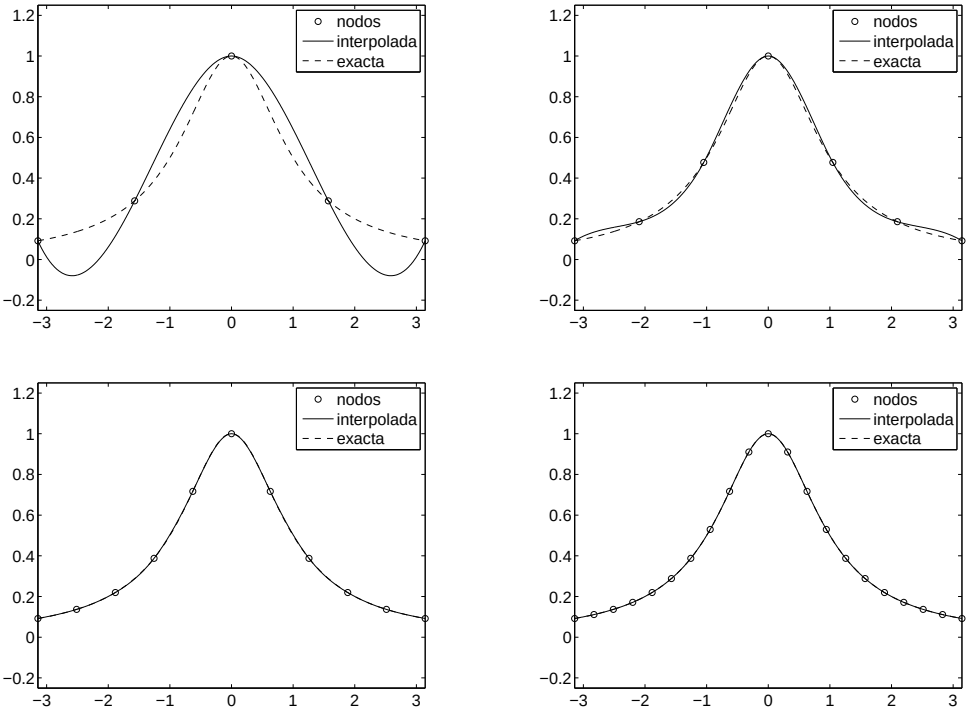

Figura 5.8: Interpolación mediante spline cúbico de la función $f(x) = \frac{1}{1+x^2}$ con 5, 7, 11 y 21 (de izquierda a derecha y de arriba a abajo) nodos

Ejemplo 5.5 *Se considera, una vez más, el ejemplo de Runge empleando ahora interpolación mediante splines cúbicos con los mismos valores de n que en la interpolación lineal a trozos (n = 4, 6, 10 y 20). En la figura 5.8 se muestran los resultados, que confirman las propiedades que se acaban de enunciar.*

Nota 5.5 *Es preciso aclarar, no obstante, que el cálculo de los interpolantes cúbicos a trozos es más complicado que el correspondiente a los interpolantes lineales a trozos (donde además el cálculo sobre cada subintervalo era, de forma inmediata, independiente del resto). El cálculo eficiente de los interpolantes cúbicos a trozos pasa por construir unas ciertas bases de funciones denominadas B-splines, que permitan simplificar las operaciones. Puede consultarse, para los detalles, el texto de Quarteroni et al. o la monografía de C. de Boor.*

5.5. Códigos disponibles

Existen multitud de paquetes que incorporan los métodos descritos (y, desde luego, muchos más). A continuación se señalan dos de ellos (que son, en realidad, bibliotecas de subrutinas escritas en fortran 90).

- PPPACK. de Boor's Cubic Spline Routines

PPPACK constituye el paquete más difundido. Se trata de una biblioteca de subrutinas escritas en fortran 90 (empleando aritmética de doble precisión) que contiene una versión de los códigos programados originalmente por C. de Boor.

La biblioteca PPPACK está disponible, por ejemplo, en la dirección
`http://www.netlib.org/pppack/`
o también (en este caso, con algunos comentarios) en
`http://www.scs.fsu.edu/~burkardt/f_src/pppack/pppack.html`

En todo caso, el propio texto de de Boor es una magnífica referencia para los códigos de esta biblioteca.

- SPLINE. Interpolation and Approximation of Data

Esta biblioteca (más limitada que la anterior) está disponible (con comentarios) en las direcciones
`nttp://www.scs.fsu.edu/~burkardt/m_src/spline/spline.html`
`http://orion.math.iastate.edu/burkardt/f_src/spline/spline.html`

5.6. Referencias

- Isaacson-Keller; *Analysis of Numerical Methods*. Wiley, 1966 (existe una reimpresión, en Dover, de 1994).
- Quarteroni et al; *Numerical Mathematics*. Springer, 2000.
- Carl de Boor; *A Practical Guide to Splines*. Springer Verlag, 1978 (existen numerosas reimpresiones, la última de ellas (en paperback) en Springer de 1994).

Capítulo 6

Derivación numérica

En este tema se consideran algunos esquemas para la aproximación de las derivadas (de funciones de una variable) mediante métodos numéricos. En particular, se recuperan en primer lugar las técnicas de interpolación mediante splines cúbicos estudiadas previamente para recordar sus propiedades en la aproximación de derivadas. Posteriormente se exponen los fundamentos de las técnicas de diferencias finitas para la aproximación numérica de las derivadas de una función.

6.1. Motivación

Si bien es cierto que el cálculo de derivadas de funciones (durante todo este tema se considerarán funciones de una única variable, aunque muchos de los conceptos que aquí se exponen pueden extenderse de forma inmediata a funciones de más variables) puede ser, en principio, abordado mediante técnicas analíticas y, en consecuencia, abordable ya sea manualmente o mediante programas de cálculo simbólico, existen diversas situaciones donde se precisa el empleo de métodos numéricos para el cálculo de las derivadas.

Algunos ejemplos de estas situaciones son las siguientes:

- estimación de las derivadas de una señal muestreada (o, con carácter más general, de una función muestreada en ciertos puntos)
- métodos de diferencias finitas para la aproximación numérica de problemas de contorno (o mixtos) asociados a ecuaciones diferenciales ordinarias o ecuaciones en derivadas parciales
- métodos basados en la derivación numérica para la aproximación de problemas de valor inicial asociados a ecuaciones diferenciales ordinarias (o a ecuaciones en derivadas parciales una vez discretizadas en la variable espacial)

Existe, en todo caso, una diferencia fundamental entre el primer ejemplo y los otros dos.

Así, en el primer ejemplo podemos suponer que contamos con los valores muestreados

en el momento en que se desean aproximar las derivadas. De este modo, el problema tiene muchos elementos comunes con el problema de interpolación y, como se verá, se pueden utilizar de hecho alguna de las técnicas ya consideradas en la resolución del problema de interpolación.

Por el contrario, en los otros dos ejemplos se plantea la aproximación de las derivadas a partir de los valores (aproximados) de la solución sobre ciertos puntos, que constituyen precisamente los valores que se buscan. Las fórmulas de derivación numérica sirven entonces para deducir las ecuaciones que han de verificar los valores (aproximados) de la solución.

6.2. Interpolación y derivación numérica

La interpolación de funciones mediante polinomios estudiada previamente ya devuelve una aproximación de las derivadas de una función (a partir de los valores de la función sobre unos ciertos puntos). Así, en efecto, si consideramos, por ejemplo, la interpolación de Lagrange global de una tabla de n puntos, obtenemos un polinomio (de grado $n+1$) que, siendo una función de clase C^∞, devuelve aproximaciones de las derivadas de cualquier orden de la función que se interpola.

Obsérvese que, de entrada, no puede esperarse que las derivadas de orden muy elevado del polinomio de interpolación constituyan buenas aproximaciones de las derivadas de la función que se interpola. Esto es así porque las derivadas de orden superior del polinomio de interpolación devolverán polinomios de orden reducido (a partir de un cierto orden de derivación serán de hecho nulas) que difícilmente constituirán una buena aproximación salvo para funciones muy particulares.

La interpolación global de Lagrange presenta, además, otro grave problema. Como se ha visto, la interpolación de funciones *no muy regulares* mediante polinomios de grado elevado origina comportamientos oscilantes del polinomio de interpolación (recuérdese, por ejemplo, la interpolación de la función de Runge: $1/(1+x^2)$). En tal caso, es obvio que ya la aproximación de la derivada primera es absolutamente deficiente.

Como se vió en el tema anterior, la interpolación de funciones *no muy regulares* (y por lo tanto, la estrategia general de interpolación empleará esta técnica) debe realizarse a través de polinomios a trozos. Se han estudiado dos alternativas para dicha interpolación mediante polinomios a trozos

- interpolación lineal a trozos
- interpolación spline cúbica

Es claro que la primera alternativa no es útil para la aproximación de las derivadas puesto que sólo admite una derivada primera (y tan sólo fuera de los nodos de integración, pues en éstos la función de interpolación no es derivable) que devuelve además una aproximación

muy pobre: es constante sobre cada intervalo comprendido entre dos nodos de interpolación consecutivos y discontinua en los nodos de interpolación.

Queda así como única posibilidad el empleo de la interpolación spline cúbica. Se va a mostrar además que dicha interpolación sí constituye una técnica eficiente para la aproximación de las derivadas de orden bajo de una función.

En primer lugar, recuérdese que la interpolación mediante una función spline cúbica de una tabla de valores $\{(x_k, f(x_k))\}_{k=0}^n$ conduce a una fución de clase $C^2([x_0, x_n])$ definida por un polinomio de grado 3 sobre cada intervalo $[x_k, x_{k+1}]$. En particular esto implica que

- se retienen aproximaciones continuas de las derivadas hasta el orden 2 pero, en general, no para las derivadas de orden superior
- la derivada tercera de la función spline cúbica es una constante sobre cada intervalo (x_k, x_{k+1}) y no está definida en los nodos $\{x_k\}_{k=1}^{n-1}$
- la función de interpolación no aporta ninguna información sobre las derivadas de orden igual o mayor que cuatro

Obsérvese que estas propiedades ya reflejan un principio observado antes: la aproximación de las derivadas va deteriorándose conforme se consideran órdenes de derivación cada vez más elevados. Este principio se verá claramente reflejado en la siguiente estimación de error.

Teorema 6.1 *(Error de aproximación de derivadas con spline cúbica)*
Sea $f : [a, b] \to \mathbf{R}$ de clase $C^4([a, b])$. Sea asimismo $\{x_k\}_{k=0}^n$ un conjunto equiespaciado de puntos con $x_0 = a$ y $x_0 = b$, y $S(x)$ la función de interpolación spline cúbica (con la condición not-a-knot *para determinar los dos grados de libertad adicionales).*

Entonces, para todo x en el intervalo $[a, b]$ se verifican las siguientes acotaciones (donde h representa el paso):

$$|f'(x) - S'(x)| \leq \frac{1}{24} h^3 \max_{a \leq \eta \leq b} |f^{(iv}(\eta)|$$

$$|f''(x) - S''(x)| \leq \frac{3}{8} h^2 \max_{a \leq \eta \leq b} |f^{(iv}(\eta)|$$

$$|f'''(x) - S'''(x)| \leq h \max_{a \leq \eta \leq b} |f^{(iv}(\eta)|$$

Demostración: Consúltese el texto de C. de Boor.

Así, siempre y cuando la función cuyas derivadas se tratan de aproximar sea suficientemente regular (de clase C^4), la interpolación mediante splines cúbicas devuelve, en efecto, aproximaciones continuas de las derivadas primeras y segundas cuyos errores pueden ser fácilmente controlables con tal de refinar la malla de puntos empleados en la interpolación.

Las cotas de error sugieren que el doblamiento del número de nodos (o, de forma más precisa, la reducción del paso a la mitad) conduce a una reducción del error de aproximación

de derivadas primeras y segundas a un octavo y un cuarto, respectivamente, de los errores con los nodos originales. Esta reducción sería, de nuevo, consistente con la mayor dificultad de aproximación de las derivaras de orden superior.

Como se ve en el resultado anterior, la interpolación con splines cúbicas es también útil para la aproximación de las derivadas terceras de la función muestreada. Obsérvese, en cualquier caso, que la aproximación devuelta es, en general, discontinua y la convergencia es relativamente lenta (al reducir el paso a la mitad, no cabe esperar que el error se reduzca más que a la mitad).

Nota 6.1 *Los resultados anteriores pueden extenderse a distribuciones no equiespaciadas de puntos. En tal caso, denotando ahora*

$$h_i \;=\; x_i - x_{i-1} \quad \text{para } i = 1, 2, \ldots, n \qquad\qquad h \;=\; \max_{1\le i\le n} h_i \qquad\qquad \beta \;=\; h \,/\, \min_{1\le i\le n} h_i$$

las acotaciones para los errores en las derivadas primera y segunda se escriben del mismo modo (con h definida ahora tal y como se acaba de hacer), en tanto que, definiendo $C_3 = (\beta + \beta^{-1})/2$, la acotación de error para la derivada tercera se escribe ahora

$$|f'''(x) - S'''(x)| \;\le\; C_3\, h \max_{a\le\eta\le b} |f^{(iv)}(\eta)|$$

Ejemplo 6.1 *Se considera la aproximación de las derivadas primeras de la función $f : [-3, +3] \to \mathbf{R}$ definida mediante*

$$f(x) = \frac{1}{1 + x^2}$$

empleando interpolación spline cúbica.

Tomando 9 nodos equiespaciados sobre dicho intervalo y calculando la interpolante spline cúbica con la condición not-a-knot *en ambos extremos, se logran las aproximaciones (para la función y su derivada primera) que se muestran en la figura 6.1. Obsérvese que se obtiene una buena aproximación de la derivada primera a pesar de emplear pocos nodos.*

6.3. Derivación numérica mediante esquemas de diferencias finitas

Como se ha mencionado, en ocasiones resulta preciso obtener esquemas para la aproximación de las derivadas de una función desconocida pero de la cual se sabe que satisface una cierta ecuación que hace intervenir sus derivadas (esto es, una ecuación diferencial).

Un ejemplo sería la búsqueda de una función $y(x)$ que verifique la siguiente ecuación diferencial

$$-\frac{d}{dx}\left(\alpha(x)\frac{dy}{dx}\right) + \beta(x)y = f(x) \qquad \text{para } x \in (a, b)$$

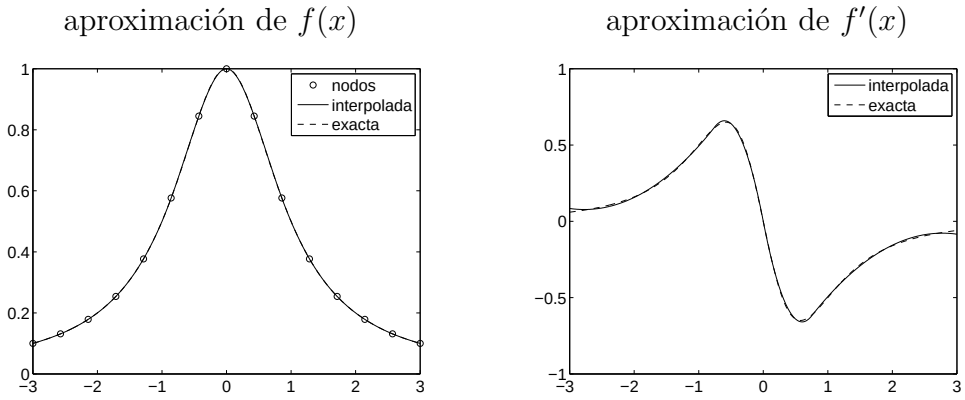

Figura 6.1: Aproximaciones de $f(x)$ y $f'(x)$ mediante interpolación spline cúbica con 9 nodos equiespaciados, para $f(x) = 1/(1+x^2)$

A pesar de su sencillez, este problema tiene un claro interés práctico pues surge, entre otros casos, en el cálculo del potencial eléctrico en un condensador (y representaría dicho potencial, α la constante dieléctrica y f la densidad de carga en el medio, si éste lo permitiese) o el cálculo de la intensidad de sonido asociada a la propagación de un frente plano (la ecuación correspondería a la versión unidimensional de la ecuación de Helmholtz; ahora y representaría la intensidad, α y β estarían relacionadas con la velocidad de propagación del sonido y la frecuencia de éste, en tanto que f representaría una cierta medida de la amplitud de las fuentes de sonido)

A la ecuación anterior es preciso añadirle unas ciertas condiciones de contorno que, por sencillez, tomaremos como

$$y(a) = y_a \qquad\qquad y(b) = y_b$$

Siempre y cuando α, β y f sean funciones continuas, se conoce del estudio de las ecuaciones lineales de orden dos (y ésta, desde luego, lo es) que existe una familia de funciones dependiente de dos parámetros arbitrarios (cuya forma será además la suma de una combinación lineal de dos soluciones de la ecuación diferencial homogénea asociada y una solución de la ecuación diferencial completa) que es solución de la ecuación diferencial. Aún más, se sabe que todas las soluciones de la ecuación diferencial están contenidas en dicha familia.

En tal caso, la resolución del problema parece sencilla: basta con obtener dicha familia e imponer las dos condiciones de contorno para buscar los valores de las dos constantes arbitrarias que corresponden a la solución del problema de contorno planteado. Dejando al margen la discusión de existencia y unicidad de solución (que, en todo caso, puede reducirse, como se ve, a la discusión de un sistema de dos ecuaciones lineales) se plantean dos claras dificultades:

- sólo se sabe buscar soluciones de una ecuación diferencial lineal homogénea con carácter

general si ésta es de coeficientes constantes; para las ecuaciones con coeficientes variables, sólo se saben resolver casos muy especiales

- aún cuando se encontrasen las dos soluciones de la ecuación diferencial homogénea asociada queda el problema de la búsqueda de la solución particular (esto es, una solución de la ecuación diferencial completa); ésta puede buscarse con un método sencillo que sólo requiere ajustar unos coeficientes (denominado precisamente método de los coeficientes indeterminados) cuando el término no homogéneo (la función $f(x)$ en el ejemplo) presenta una forma muy especial, en otro caso es preciso acudir a un método más general (conocido como método de variación de parámetros) que obliga a calcular dos integrales que podrían (y con frecuencia lo harán) no tener primitiva

Así nos enfrentamos con un problema donde parece claro que no es posible contar con las técnicas analíticas conocidas para resolver todos los casos de interés práctico sino sólo un número muy limitado de casos. Se plantea entonces la necesidad de resolverlo de forma aproximada mediante métodos numéricos.

Una de las técnicas más empleadas en la resolución de estos problemas son las denominadas técnicas de diferencias finitas, que se plantean buscar los valores aproximados de la función y sobre $n+1$ puntos distribuidos sobre el intervalo $[a,b]$

$$a \;=\; x_0 \;<\; x_1 \;<\; \cdots \;<\; x_{n-1} \;<\; x_n \;=\; b$$

que denotaremos mediante $\{y_i\}_{i=0}^{n}$, planteando para ello un sistema de $n+1$ ecuaciones formado del modo siguiente

- una ecuación para imponer la condición de contorno en $x = a$: $y_0 = y_a$
- una ecuación para imponer la condición de contorno en $x = b$: $y_n = y_b$
- una ecuación por cada *nodo interior*, donde se imponga que se satisface la ecuación diferencial (de forma aproximada, pues no se dispone de los valores de las derivadas)

Se plantea entonces la cuestión de cómo aproximar los valores de las derivadas primera y segunda de una cierta función de la cual sólo conoceremos sus valores en ciertos puntos. Obsérvese que el probema es muy semejante al tratado en la sección anterior (donde, a partir de los valores muestreados de la función sobre unos ciertos nodos, se interpolaba mediante una función spline cúbica para lograr aproximaciones de las derivadas de la función) y la única diferencia es que ahora, en el momento de aproximar la derivada, no conocemos todavía esa tabla de valores. Recuérdese que, justamente, queremos obtener la expresión de esas derivadas (a partir de los valores aproximados de la función en los nodos) precisamente para construir el sistema de ecuaciones que nos permita hallar esos valores. Así, las técnicas empleadas en la sección anterior no resultan demasiado útiles en este nuevo contexto.

6.3.1. Idea general de las fórmulas en diferencias finitas

Se plantea entonces la cuestión de cómo aproximar las derivadas de una cierta función a partir de una combinación sencilla de los valores de la función (o, más bien, de una aproximación de éstos). Obsérvese que en los ejemplos mencionados debe lograrse que el sistema de ecuaciones que posteriormente se habrá de resolver (para conocer los valores aproximados de la función) sea lo más sencillo posible.

Así, si fuese posible, podría buscarse que se trate de un sistema de ecuaciones lineales asociado a una matriz hueca con términos no nulos sólo sobre la diagonal y unas pocas subdiagonales. La linealidad del sistema impone entonces que la aproximación de las derivadas debe hacerse mediante combinaciones lineales de los valores de la función en los distintos puntos (o, para ser más precisos, de las aproximaciones de estos valores). La estructura *en banda* de la matriz impone que dicha combinación incorpore solamente los valores correspondientes a nodos muy próximos al punto donde se quiere aproximar la derivada.

Con todo esto esto en mente, supongamos que se cuenta con un conjunto de puntos equiespaciados

$$a \;=\; x_0 \;<\; x_1 \;<\; \cdots \;<\; x_{n-1} \;<\; x_n \;=\; b$$

y los valores de una cierta función f (que supondremos regular) sobre dichos puntos $\{f(x_i)\}_{i=0}^{n}$.

Dado un nodo genérico, x_i, se plantea entonces la cuestión de cómo combinar linealmente el valor $f(x_i)$ con algunos valores prómixos a fin de lograr una aproximación de alguna derivada (generalmente de primer o segundo orden) de f en x_i.

La búsqueda de dicha combinación pasará por reescribir los valores de f en los nodos cercanos en forma de desarrollo de Taylor y ajustar los coeficientes de la combinación lineal a fin de lograr aproximar el valor deseado. Este proceso permite además controlar la expresión del error, que se obtendrá de los términos que no se cancelen en la combinación de los desarrollos de Taylor.

A continuación se consideran varios ejemplos de fórmulas de este tipo para la aproximación de las derivadas de órdenes uno y dos.

6.3.2. Fórmulas en diferencias finitas para derivadas de primer orden

Una expresión inmediata para la aproximación de $f'(x_i)$ resultaría de la siguiente combinación lineal del valor $f(x_i)$ con el valor $f(x_{i+1})$ (denotando $h = x_{i+1} - x_i$):

$$f'(x_i) \;\simeq\; \frac{1}{h} f(x_{i+1}) - \frac{1}{h} f(x_i)$$

Desde luego, ésta es una fórmula que devuelve una aproximación arbitrariamente precisa de $f'(x_i)$ con tal de tomar h suficientemente pequeño, puesto que el límite cuando h tiende

a cero corresponde precisamente a la definición de derivada de f en el punto x_i.

Nos planteamos ahora el error cometido por dicha fórmula cuando se emplea (como se hará, obviamente, en la práctica) con h reducido pero finito. Para ello, tal y como se ha descrito previamente, se construyen los desarrollos de Taylor de los valores de f empleados

$$\begin{aligned} f(x_{i+1}) &= f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(\eta) \\ f(x_i) &= f(x_i) \end{aligned}$$

(donde η corresponde a un cierto punto en el intervalo $[x_i, x_{i+1}]$) y se combinan de acuerdo con la fórmula propuesta para obtener

$$\frac{1}{h}f(x_{i+1}) - \frac{1}{h}f(x_i) = f'(x_i) + \frac{h}{2}f''(\eta)$$

Así, se tiene una expresión para el error

$$e_i = f'(x_i) - \left(\frac{1}{h}f(x_{i+1}) - \frac{1}{h}f(x_i)\right) = -\frac{h}{2}f''(\eta)$$

que asegura, entre otras cosas, que el error se reducirá de forma lineal conforme se refina la malla de puntos (así, dividiendo h a la mitad es esperable que el error también se reduzca a la mitad) y que el error depende de la *dificultad* de la función, medida en términos de su derivada segunda.

Es fácil comprobar que también puede aproximarse $f'(x_i)$, empleando un punto situado a la izquierda, mediante la fórmula

$$f'(x_i) \simeq \frac{1}{h}f(x_i) - \frac{1}{h}f(x_{i-1})$$

con idénticas propiedades de convergencia (salvo la modificación del signo en la expresión del error).

Se plantea ahora la cuestión acerca de la posibilidad de lograr una mejor aproximación de $f'(x_i)$ empleando, junto con $f(x_i)$, simultáneamente los valores $f(x_{i-1})$ y $f(x_{i+1})$. Para ello, se buscan unos ciertos coeficientes α, β y γ tales que la combinación

$$\alpha\frac{1}{h}f(x_{i+1}) + \beta\frac{1}{h}f(x_i) + \gamma\frac{1}{h}f(x_{i-1})$$

devuelva una aproximación lo mejor posible de $f'(x_i)$. Obsérvese que se ha incluido un factor $\frac{1}{h}$ para garantizar que los coeficientes que buscamos sean independientes de h y así identificar más fácilmente los órdenes de cada término.

Desde luego, en la expresión anterior, debemos entender *lo mejor posible* en el sentido siguiente: la expresión del error de aproximación deberá decrecer tan rápido como se pueda cuando h se acerca a cero.

Escribiendo los desarrollos de Taylor

$$\begin{aligned} f(x_{i+1}) &= f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(\eta) \\ f(x_i) &= f(x_i) \\ f(x_{i-1}) &= f(x_i) - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(\xi) \end{aligned}$$

y agrupando los términos en la misma potencia de h se tiene:

$$\begin{aligned} \alpha\frac{1}{h}f(x_{i+1}) + \beta\frac{1}{h}f(x_i) + \gamma\frac{1}{h}f(x_{i-1}) &= (\alpha+\beta-\gamma)h^{-1}f(x_i) \\ &\quad +(\alpha+\gamma)f'(x_i) \\ &\quad +(\alpha-\gamma)\frac{h}{2}f''(x_i) \\ &\quad +(\alpha f'''(\eta)+\gamma f'''(\xi))\frac{h^2}{6} \end{aligned}$$

Así, a fin de lograr que con h tendiendo a cero dicha combinación recupere el valor de $f'(x_i)$ es preciso imponer

$$\alpha+\beta-\gamma=0$$

$$\alpha+\gamma=1$$

Sin embargo, puesto que disponemos de tres parámetros, podemos imponer una condición adicional que garantice que el siguiente término de los desarrollos también se cancele (si no fuese así, el error tendría de nuevo un comportamiento lineal con h) haciendo

$$\alpha-\gamma=0$$

Las tres ecuaciones devuelven entonces como solución

$$\alpha = \gamma = \frac{1}{2} \qquad \beta = 0$$

y la fórmula se escribe entonces

$$f'(x_i) \simeq \frac{1}{2h}f(x_{i+1}) - \frac{1}{2h}f(x_{i-1})$$

Por su parte el error de aproximación resulta ahora

$$e_i = f'(x_i) - \left(\frac{1}{2h}f(x_{i+1}) - \frac{1}{2h}f(x_{i-1})\right) = -\frac{1}{2}(f'''(\eta)+f'''(\xi))\frac{h^2}{6}$$

que puede reescribirse (empleando la regularidad de la función)

$$e_i \;=\; -f'''(\zeta)\frac{h^2}{6}$$

para algún ζ en el intervalo $[\eta, \xi]$ (y, por lo tanto, también en $[x_{i-1}, x_{i+1}]$).

La expresión de error obtenida muestra entonces que se ha logrado mejorar un orden la aproximación (se tiene ahora una aproximación de orden dos) y, por ejemplo, es esperable que el error se reduzca a la cuarta parte cuando se divide h a la mitad.

A continuación se resumen las tres fórmulas obtenidas junto con el nombre habitual y la expresión del error (para una función suficientemente regular)

$\frac{1}{h}(f(x_{i+1}) - f(x_i))$	$-\frac{h}{2}f''(\eta)$	fórmula en diferencias de dos puntos progresiva
$\frac{1}{h}(f(x_i) - f(x_{i-1}))$	$+\frac{h}{2}f''(\eta)$	fórmula en diferencias de dos puntos regresiva
$\frac{1}{2h}(f(x_{i+1}) - f(x_{i-1}))$	$-\frac{h^2}{6}f'''(\eta)$	fórmula en diferencias de dos puntos centrada

Ejemplo 6.2 *Se considera la aproximación mediante las fórmulas anteriores de la derivada de la función $f(x) = \exp(x)$ en $x = 0$ para diferentes valores de h.*

Se denotarán mediante e_h^+, e_h^- y e_h^0 los errores cometidos en la aproximación mediante las fórmulas de dos puntos progresiva, regresiva y centrada, respectivamente. De los resultados anteriores se tiene

$$e_h^+ = 1 - \frac{\exp(h) - 1}{h} = -\frac{h}{2}f''(\eta^+) \simeq -\frac{h}{2}$$

$$e_h^- = 1 - \frac{1 - \exp(-h)}{h} = +\frac{h}{2}f''(\eta^-) \simeq +\frac{h}{2}$$

$$e_h^0 = 1 - \frac{\exp(h) - \exp(-h)}{2h} = -\frac{h^2}{6}f'''(\eta) \simeq -\frac{h^2}{6}$$

Se obtiene así la siguiente tabla de errores para los valores indicados de h

h	e_h^+	e_h^-	e_h^0
1	-7.1828×10^{-1}	$+3.6788 \times 10^{-1}$	-1.7520×10^{-1}
10^{-1}	-5.1709×10^{-2}	$+4.8374 \times 10^{-2}$	-1.6675×10^{-3}
10^{-2}	-5.0167×10^{-3}	$+4.9834 \times 10^{-3}$	-1.6667×10^{-5}
10^{-3}	-5.0017×10^{-4}	$+4.9983 \times 10^{-4}$	-1.6667×10^{-7}
10^{-4}	-5.0002×10^{-5}	$+4.9998 \times 10^{-5}$	-1.6669×10^{-9}
10^{-5}	-5.0000×10^{-6}	$+5.0000 \times 10^{-6}$	-1.2102×10^{-11}

Obsérvese que las estimaciones de error ofrecen, cuando se emplean valores reducidos de h, una idea muy precisa del error cometido por cada fórmula.

□

Ejemplo 6.3 *Se considera ahora la aproximación numérica de la derivada primera de una función no tan regular:*

$$f(x) = x + \omega \, \mathrm{sinc}(\omega x)$$

Esta función presenta derivadas con valores absolutos grandes si ω es, a su vez, grande. Obérvese que en el entorno del origen, se tiene

$$f(x) \sim x + \omega - \frac{1}{6} \, \omega^3 x^2 + \mathcal{O}(x^4)$$

y por lo tanto, para sus dos primeras derivadas se espera

$$f'(x) \sim 1 - \frac{1}{3} \, \omega^3 x + \mathcal{O}(x^3)$$

$$f''(x) \sim -\frac{1}{3} \, \omega^3 + \mathcal{O}(x^2)$$

Así, con $\omega = 100$, si empleamos la fórmula de dos puntos progresiva para aproximar la derivada primera en el origen

$$f'(0) \simeq \frac{f(h) - f(0)}{h}$$

se tiene un error de aproximación

$$e_h \; = \; -\frac{h}{2} f''(\eta) \; \simeq \; \frac{h\omega^3}{6} \; = \; \frac{1}{6} 10^6 h$$

y no cabe esperar buenas aproximaciones salvo que h sea muy reducido.

A continuación se muestra el valor de la aproximación mediante la fórmula de dos puntos progresiva y el error de aproximación asociado para los valores de h indicados

h	aproximación	e_h
1	$-9.9506 \times 10^{+1}$	$1.0051 \times 10^{+2}$
10^{-1}	$-1.0534 \times 10^{+3}$	$1.0544 \times 10^{+3}$
10^{-2}	$-1.5843 \times 10^{+3}$	$1.5853 \times 10^{+3}$
10^{-3}	$-1.6558 \times 10^{+2}$	$1.6658 \times 10^{+2}$
10^{-4}	$-1.5667 \times 10^{+1}$	$1.6667 \times 10^{+1}$
10^{-5}	-6.6667×10^{-1}	1.6667
10^{-6}	$+8.3333 \times 10^{-1}$	1.6667×10^{-1}
10^{-7}	$+9.8333 \times 10^{-1}$	1.6667×10^{-2}
10^{-8}	$+9.9833 \times 10^{-1}$	1.6676×10^{-3}
10^{-9}	$+9.9983 \times 10^{-1}$	1.6689×10^{-4}

□

En la aproximación efectiva de las derivadas mediante fórmulas de diferencias finitas existe una fuente adicional de error, asociada a los redondeos efectuados al operar en aritmética finita.

Así, mientras en aritmética exacta el error de aproximación de la derivada $f'(x)$ empleando la denóminada *fórmula progresiva*, e_h, se define como

$$e_h = f'(x) - \frac{f(x+h) - f(x)}{h}$$

en aritmética finita el error de aproximación, $\hat{e}_h$, será

$$\hat{e}_h = f'(x) - \frac{\hat{f}(x+h) - \hat{f}(x)}{h}$$

donde se denota $\hat{f}(x)$ la evaluación efectiva de la función en aritmética finita.

Un análisis elemental (pero suficientemente ilustrativo) de este error puede suponer que se tiene

$$\frac{\hat{f}(x+h) - \hat{f}(x)}{h} = \frac{f(x+h) - f(x) + \varphi}{h}$$

donde $\varphi \sim \epsilon$ (con ϵ la precisión relativa empleada en la aritmética finita). Obsérvese que este análisis elemental no tiene en consideración los problemas de pérdida de cifras significativas en la sustracción de dos cantidades muy próximas.

Con la suposición anterior, el error de aproximación en aritmética finita viene dado por

$$\hat{e}_h = f'(x) - \frac{f(x+h) - f(x)}{h} - \frac{\varphi}{h} = e_h - \frac{\varphi}{h}$$

Empleando un almacenamiento en coma flotante con la norma IEEE para representación con doble precisión, los redondeos tienen un orden de magnitud de 10^{-16} (el denominado *epsilon de la máquina* es 2.2204×10^{-16}). Así, se tendrá que el error final tiene una contribución e_h que, como se ha visto, viene dada por

$$e_h \;=\; \frac{h}{2} f''(\eta)$$

y una segunda contribución, debida a los redondeos, de orden de magnitud

$$\frac{\varphi}{h} \;\sim\; \frac{1}{h}\, 10^{-16}$$

Así, la principal contribución al error será e_h para valores moderadamente pequeños de h. En general, denominando ϵ a la precisión relativa, esto ocurre si $h << \epsilon^{1/2}$. Empleando doble precisión (de acuerdo con la norma IEEE) el error de redondeo será despreciable siempre que h sea más grande que 10^{-8}. Sin embargo, si h toma valores más reducidos los errores de redondeo comenzarán a ser apreciables.

Ejemplo 6.4 *Considérese, de nuevo, la aproximación de* $f'(x)$ *para la función* $\exp(x)$ *en* $x = 0$ *empleando la fórmula de dos puntos* progresiva

$$\frac{exp(h) - 1}{h}$$

El error de aproximación de la derivada operando con aritmética exacta, e_h*, es aproximadamente* $-h/2$*. Trabajando con doble precisión, éste es el único error apreciable que cabe esperar con* $h < 10^{-8}$*. Sin embargo, cuando* h *toma valores más reducidos, se debe esperar que el error comience a crecer debido a la contribución de los errores de redondeo.*

De este modo, si se toma $h = 10^{-12}$ *aunque el error* e_h *sea muy reducido (aproximadamente* -5×10^{-13}*) se esperan unos errores de redondeo del orden de* $\frac{\epsilon}{h} \sim 10^{-4}$*, que dominarán por completo el error final. Por el contrario, con un paso de* $h = 1e-6$ *el error* e_h *es algo mayor (en torno a* -5×10^{-7}*) pero la contribución del error de redondeo será ahora desprecible (del orden de* 10^{-10}*) y el error total será más reducido que en el caso anterior.*

A continuación se muestra una tabla con los errores totales (resultantes al programar la fórmula empleando doble precisión) para los valores indicados de h*, que ilustra las ideas expuestas.*

h	$\hat{e}_h$
1	-7.1828×10^{-1}
10^{-1}	-5.1709×10^{-2}
10^{-2}	-5.0167×10^{-3}
10^{-3}	-5.0017×10^{-4}
10^{-4}	-5.0002×10^{-5}
10^{-5}	-5.0000×10^{-6}
10^{-6}	-4.9996×10^{-7}
10^{-7}	-4.9434×10^{-8}
10^{-8}	$+6.0775 \times 10^{-9}$
10^{-9}	-8.2740×10^{-8}
10^{-10}	-8.2740×10^{-8}
10^{-11}	-8.2740×10^{-8}
10^{-12}	-8.8901×10^{-5}
10^{-13}	$+7.9928 \times 10^{-4}$
10^{-14}	$+7.9928 \times 10^{-4}$
10^{-15}	-1.1022×10^{-1}
10^{-16}	-1.0000

Obsérvese que del análisis anterior (y esta tabla lo respalda) la elección óptima del paso h es aquella que iguala el orden de ambos errores, que es $h = \sqrt{\epsilon}$. En doble precisión (con la norma IEEE), corresponde a tomar $h = 10^{-8}$. Es habitual encontrar esta recomendación, por ejemplo, cuando se aproxima numéricamente la matriz jacobiana de una cierta función vectorial con la fórmula de diferencias finitas descrita, al implementar un método de cuasi-Newton con derivación numérica para resolver un sistema de ecuaciones no lineales (se denomina, de forma genérica, métodos de cuasi-Newton a los métodos basados en el método de Newton que aproximan de un modo u otro la matriz jacobiana).

□

6.3.3. Fórmulas en diferencias finitas para derivadas de segundo orden

Se busca ahora aproximar la derivada segunda de una cierta función (regular) en un punto mediante una combinación de los valores de esa función en varios puntos cercanos.

De acuerdo con las ideas generales expuestas anteriormente, dicha combinación se buscará de modo que logre la mejor aproximación posible (para funciones regulares), estudiando los desarrollos de Taylor de la función en torno al punto donde se quiere aproximar la derivada.

Parece claro, en todo caso, que serán necesarios al menos tres puntos para lograr, mediante la adecuada combinación de valores de la función en dichos puntos, cancelar los términos en $f(x_i)$ y $f'(x_i)$ de los desarollos de Taylor y hacer uno el coeficiente que afecta a $f''(x_i)$

(puesto que se exigen tres condiciones será preciso disponer de tres grados de libertad en la combinación lineal).

Se consideran entonces los valores de f en el punto x_i (donde se quiere aproximar la derivada segunda de f) y en los puntos $x_{i+1} = x_i + h$ y $x_{i-1} = x_i - h$. Los correspondientes desarrollos de Taylor devuelven

$$\begin{aligned} f(x_{i+1}) &= f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(x_i) + \frac{h^4}{24}f^{(iv}(\eta) \\ f(x_i) &= f(x_i) \\ f(x_{i-1}) &= f(x_i) - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f'''(\xi) + \frac{h^4}{24}f^{(iv}(\xi) \end{aligned}$$

y se busca una combinación

$$\alpha\frac{1}{h^2}f(x_{i+1}) + \beta\frac{1}{h^2}f(x_i) + \gamma\frac{1}{h^2}f(x_{i-1})$$

para aproximar la derivada segunda de f en x_i (obsérvese que ya se ha incluido un factor $\frac{1}{h^2}$ en los coeficientes que necesariamente habría de aparecer en ellos).

La combinación anterior da lugar, al sustituir los desarrollos y agrupar los términos con idénticas potencias de h, a

$$\begin{aligned} \alpha\frac{1}{h^2}f(x_{i+1}) + \beta\frac{1}{h^2}f(x_i) + \gamma\frac{1}{h^2}f(x_{i-1}) &= (\alpha+\beta+\gamma)h^{-2}f(x_i) \\ &\quad +(\alpha-\gamma)h^{-1}f'(x_i) \\ &\quad +(\frac{1}{2}\alpha+\frac{1}{2}\gamma)f''(x_i) \\ &\quad +(\frac{1}{6}\alpha-\frac{1}{6}\gamma)hf'''(x_i) \\ &\quad +(\alpha f^{(iv}(\eta)+\gamma f^{iv}(\xi))\frac{h^2}{24} \end{aligned}$$

Si imponemos que, con h tendiendo a cero, la combinación anterior debe converger a $f''(x_i)$ es preciso que se cumpla

$$\alpha+\beta+\gamma = 0$$

$$\alpha-\gamma = 0$$

$$\frac{1}{2}\alpha+\frac{1}{2}\gamma = 1$$

obteniéndose entonces, como única solución

$$\alpha = 1 \qquad \beta = -2 \qquad \gamma = 1$$

En suma, la aproximación buscada mediante una fórmula de diferencias finitas de tres puntos (que denominaremos también centrada) es

$$f''(x_i) \simeq \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$$

En cuanto al error de aproximación, es fácil ver que al ser $\alpha = \gamma$ el término en $f'''(x_i)$, que debería devolver la primera contribución al error, se cancela y se tiene

$$e_h = f''(x_i) - \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} = -(f^{(iv}(\eta) + f^{(iv}(\xi))\frac{h^2}{24}$$

de modo que, empleando el teorema del valor medio, puede escribirse

$$e_h = -f^{(iv}(\zeta)\frac{h^2}{12}$$

para algún $\zeta \in [x_{i-1}, x_{i+1}]$.

Ejemplo 6.5 *Retomamos la aproximación de las derivadas de la función $f(x) = \exp(x)$ y consideramos ahora la aproximación de la derivada segunda en $x = 0$ mediante la fórmula de diferencias finitas que se acaba de obtener.*

Se tiene entonces la aproximación

$$\frac{\exp(h) - 2 + \exp(-h)}{h^2}$$

para el valor de dicha derivada. A continuación se muestra el error de aproximación, e_h, para los valores indicados de h

h	e_h
1	-8.6161×10^{-2}
10^{-1}	-8.3361×10^{-4}
10^{-2}	-8.3334×10^{-6}
10^{-3}	-8.3407×10^{-8}

Puede comprobarse que, al igual que ocurría en las fórmulas para las derivadas de primer orden, el empleo de valores muy reducidos de h puede empeorar la aproximación efectiva al trabajar en aritmética finita. En particular, si se emplea la representación con doble precisión de la norma IEEE, retomando el análisis expuesto para las derivadas de primer orden, la contribución del error de redondeo es del orden $\frac{\epsilon}{h^2}$ y por tanto comenzará a ser apreciable para $h < \epsilon^{1/4} \sim 10^{-4}$. A continuación se muestra el error de aproximación para valores reducidos de h.

h	e_h
10^{-3}	-8.3407×10^{-8}
10^{-4}	-5.0248×10^{-9}
10^{-5}	$+1.0275 \times 10^{-6}$
10^{-6}	$+2.2122 \times 10^{-5}$
10^{-7}	$+7.9928 \times 10^{-4}$
10^{-8}	$+1.0000$
10^{-9}	$-1.1002 \times 10^{+2}$

□

Nota 6.2 *Se ha considerado aquí como construir de modo directo fórmulas de diferencias finitas para la aproximación de las derivadas de orden dos. Alternativamente, también pueden deducirse esquemas en diferencias finitas para la aproximación de las derivadas de orden dos empleando, de forma sucesiva, las fórmulas estudiadas para la aproximación de las derivadas de primer orden.*

Por ejemplo, la aproximación de la derivada segunda en un cierto punto x_i perteneciente a una malla de nodos equiespaciados $\{x_j\}_{j=0}^n$ puede hacerse empleando la fórmula de dos puntos progresiva para la aproximación de todas las derivadas de primer orden:

$$\begin{aligned} f''(x_i) &\simeq \frac{f'(x_{i+1}) - f'(x_i)}{h} \\ &\simeq \frac{\dfrac{f(x_{i+2}) - f(x_{i+1})}{h} - \dfrac{f(x_{i+1}) - f(x_i)}{h}}{h} \\ &= \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} \end{aligned}$$

El análisis del error de esta fórmula permite comprobar que, cuando h tiende a cero, la fórmula converge (siempre que f sea regular) al valor $f''(x_i)$. No obstante, este análisis permite ver que la convergencia es simplemente lineal; esto es, el error e_h se comporta como $\mathcal{O}(h)$. De hecho, como sabemos, esta fórmula devuelve una aproximación de orden dos de $f''(x_{i+1})$.

□

6.3.4. Aplicación de las fórmulas en diferencias finitas

Se considera, para terminar, la aplicación de las fórmulas en diferencias finitas expuestas a la resolución numérica de un problema de contorno asociado a una ecuación diferencial ordinaria.

En particular, se busca $f(x)$ solución de la ecuación diferencial

$$f''(x) + f'(x) + (2\pi x)^2 f = -2\pi(x+1)\text{sen}(\pi x^2) \qquad x \in (0,1)$$

con las condiciones (de contorno)

$$f(0) = 1 \qquad f(1) = -1$$

Para dicho problema se conoce que existe una única solución. Sin embargo, puesto que no existe ninguna técnica analítica que permita resolver, con carácter general, ecuaciones diferenciales ordinarias lineales con coeficientes variables, no es posible obtenerla de forma analítica. Se plantea entonces su resolución numérica.

Así, se considera un conjunto de puntos equiespaciados sobre el intervalo $[0,1]$:

$$0 = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = 1$$

y se buscan los valores aproximados de f sobre cada uno de esos puntos. Denotaremos mediante f_i el valor aproximado de $f(x_i)$.

Como se mencionó anteriormente, es preciso formar ahora un sistema de ecuaciones que permita encontrar los valores de f_i. Las ecuaciones para y_0 e y_n son evidentemente

$$y_0 = 1$$

$$y_n = -1$$

en tanto que para los nodos interiores se obtendrá una ecuación al forzar que se verifique, en dicho nodo, la ecuación diferencial sustituyendo las derivadas por una fórmulas en diferencias finitas adecuadas.

De acuerdo con lo expuesto anteriormente pueden emplearse, para un nodo genérico x_i, las fórmulas

$$f'(x_i) \simeq \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

$$f''(x_i) \simeq \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$$

que devuelven aproximaciones de segundo orden (esto es, el error de aproximación se comporta como $\mathcal{O}(h^2)$).

Se tiene entonces que, para ese nodo genérico x_i, se impondrá la ecuación

$$\frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + \frac{f_{i+1} - f_{i-1}}{2h} + (2\pi x_i)^2 f_i = 2\pi(1+x_i)\text{sen}(\pi x_i^2)$$

donde se han empleado los valores aproximados de f que se buscan.

Con el procedimiento descrito se ha construido entonces un sistema de $n+1$ ecuaciones para encontrar las $n+1$ incógnitas $\{x_j\}_{j=0}^n$, de la forma

$$f_0 = 1$$

$$\frac{f_2 - 2f_1 + f_0}{h^2} + \frac{f_2 - f_0}{2h} + (2\pi x_1)^2 f_1 = 2\pi(1 + x_1)\text{sen}(\pi x_1^2)$$

$$\frac{f_3 - 2f_2 + f_1}{h^2} + \frac{f_3 - f_1}{2h} + (2\pi x_2)^2 f_2 = 2\pi(1 + x_2)\text{sen}(\pi x_2^2)$$

...

$$\frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + \frac{f_{i+1} - f_{i-1}}{2h} + (2\pi x_i)^2 f_i = 2\pi(1 + x_i)\text{sen}(\pi x_i^2)$$

...

$$\frac{f_n - 2f_{n-1} + f_{n-2}}{h^2} + \frac{f_n - f_{n-2}}{2h} + (2\pi x_{n-1})^2 f_{n-1} = 2\pi(1 + x_{n-1})\text{sen}(\pi x_{n-1}^2)$$

$$f_n = -1$$

que, llevado a forma matricial, corresponde a un sistema asociado a una matriz tridiagonal, fácil de resolver mediante eliminación de Gauss para obtener los valores aproximados de f en cada nodo.

Puesto que cuanto menor sea h más reducido es el error de aproximación de la derivadas que proporcionan las fórmulas de diferencias finitas, cabe esperar que también será mejor la aproximación que devuelve el método anterior para la solución de la ecuación diferencial.

A continuación se muestran las soluciones obtenidas para los valores indicados de h donde en efecto se observa la mejora de la aproximación con la reducción del paso. Obsérvese que se ha representado con puntos los valores de las aproximaciones $\{f_i\}_{i=0}^n$ y en trazo continuo la solución exacta $f(x) = \cos(\pi x^2)$.

6.4. Referencias

- **C. de Boor; *A practical guide to splines* Revised edition. Springer, 2001.**
- referencia a [Kincaid-Cheney] o [Burden-Faires] para MDF en edo

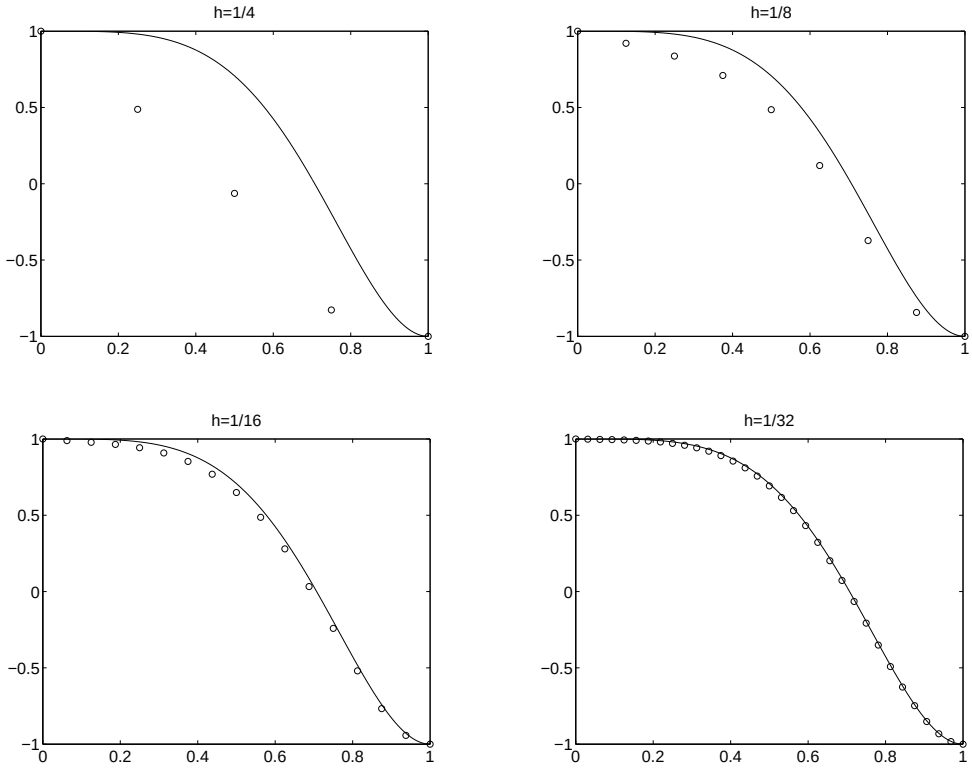

Figura 6.2: Solución exacta y aproximada del problema de contorno para los valores indicados del paso de malla h

Capítulo 7

Integración numérica

En este tema se consideran las principales familias de métodos para la evaluación numérica de integrales de funciones de una variable. Se exponen en particular las principales fórmulas basadas en interpolación numérica, tanto simples como compuestas y se exponen las ideas de las fórmulas adaptativas.

7.1. Motivación

La integración de funciones de una variable (y, obviamente, también la integración de funciones de varias variables) conduce frecuentemente a cálculos que no pueden ser resueltos analíticamente. De hecho, sólo un grupo limitado de funciones (aunque relevantes, en parte precisamente por esta propiedad) aceptan funciones primitivas. En el resto de los casos, el cálculo de sus integrales sólo puede llevarse a cabo de forma aproximada mediante métodos numéricos.

Algunos problemas específicos que conducen al cálculo de integrales mediante métodos numéricos son

- tratamiento de señales o, con carácter más general, postprocesado de funciones muestreadas, donde es preciso calcular la integral de la función muestreada (por ejemplo, cálculo de una carga eléctrica a partir de muestreo de la intensidad de corriente) o alguna integral que involucre a dicha función (como, por ejemplo, el cálculo de convoluciones)

- métodos basados en la integración numérica para la resolución de problemas de valor inicial asociados a ecuaciones diferenciales ordinarias (como es el caso de los métodos de Runge-Kutta y los métodos de Adams, que constituyen la base de la mayor parte de los códigos para la integración de problemas generales -o no rígidos-)

7.2. Interpolación e integración numérica

Dada una función $f(x)$ y su evaluación sobre un cierto conjunto de puntos $\{x_i\}_{i=0}^n$, se ha estudiado previamente cómo aproximar dicha función mediante un polinomio de interpolación. Al margen de las posibles dificultades que implica este proceso en lo que se refiere a la consecución de una buena aproximación (que constituye una cuestión ya examinada previamente y sobre la que se volverá más tarde) es claro que dicho polinomio permite también obtener aproximaciones del valor de la integral de f.

Integración numérica mediante interpolación

Sea $f : [a, b] \to \mathbf{R}$ y supóngase que deseamos calcular

$$\int_a^b f(x)\, dx$$

y nos planteamos su cálculo aproximado en dos pasos:

- tomamos unos ciertos puntos $x_0, x_1, \ldots, x_n$ sobre el intervalo $[a, b]$ y calculamos el polinomio de interpolación $P_n(x)$ de los valores de f sobre esos puntos
- calculamos una aproximación de la integral de f mediante

  $$\int_a^b P_n(x)\, dx$$

  puesto que $P_n(x)$ devuelve una aproximación de $f(x)$ y la integral puede calcularse de forma exacta ya que los polinomios sí admiten función primitiva.

En la práctica, no resulta preciso realizar explícitamente cada uno de los dos pasos. Recordando la formulación del problema de interpolación a partir de las funciones de base de Lagrange, $L_i(x)$, puede escribirse el polinomio de interpolación de f como

$$P_n(x) = \sum_{i=0}^{n} f(x_i) L_i(x)$$

de modo que la integral resulta

$$\int_a^b P_n(x)\, dx = \sum_{i=0}^{n} f(x_i) \int_a^b L_i(x)\, dx$$

y denominando

$$w_i = \int_a^b L_i(x)\, dx \qquad i = 0, 1, 2, \ldots, n$$

basta con conocer los $n + 1$ coeficientes w_i (independientes de la función f) junto con los valores de la función f sobre los correspondientes puntos para poder calcular la aproximación de la integral.

Así, los esquemas de integración numérica basados en la interpolación (que se denominan habitualmente *fórmulas de cuadratura*) se escriben genéricamente de la forma

$$\int_a^b f(x)\, dx \simeq \sum_{i=0}^{n} f(x_i)\, w_i$$

Por su parte, los puntos $\{x_i\}_{i=0}^n$ reciben el nombre de *nodos de integración numérica* (o *nodos de cuadratura*) y los coeficientes w_i se denominan *pesos de la fórmula de integración numérica* (o *pesos de la fórmula de cuadratura*).

Transformación del intervalo de integración

Obsérvese que los coeficientes $\{w_i\}_{i=0}^n$ sólo dependen, en principio, del intervalo $[a, b]$ y de la colocación de los nodos $\{x_i\}_{i=0}^n$ dentro de ese intervalo. Además, gracias a las propiedades de invarianza del operador de interpolación, es posible comprobar que no dependen de ambos datos por separado sino que dependen exclusivamente de la posición relativa de los nodos sobre el intervalo. En todo caso, como se va a ver a continuación, se eliminará la posible dependencia con respecto al intervalo de integración mediante la transformación a un intervalo de referencia (sobre el cual tabular los valores de los pesos).

Se considera el cálculo de la integral

$$\int_a^b f(x)\, dx$$

mediante un método de integración numérica de tipo interpolatorio diseñado para la integración sobre el intervalo de referencia $[-1, +1]$

$$\int_{-1}^{+1} g(\zeta)\, d\zeta \simeq \sum_{i=0}^{n} g(\zeta_i) w_i$$

Como se verá, basta para ello con realizar antes un cambio de variable que transforme el intervalo. Así, sea la función $h : [-1, +1] \to [a, b]$ definida mediante

$$h(\zeta) = \frac{a+b}{2} + \frac{b-a}{2}\zeta$$

que verifica las hipótesis del teorema de cambio de variable. Se tiene entonces

$$\int_a^b f(x)\, dx = \int_{-1}^{+1} f(h(\zeta)) h'(\zeta)\, d\zeta = \frac{b-a}{2} \int_{-1}^{+1} f(h(\zeta))\, d\zeta$$

y es suficiente, por lo tanto, con aplicar la fórmula de integración numérica a la función $f(h(\zeta))$

$$\int_a^b f(x)\, dx \simeq \frac{b-a}{2} \sum_{i=0}^{n} f(h(\zeta_i)) w_i$$

Estimación de error de integración numérica

En el tema dedicado a interpolación se estudió una expresión del error de interpolación para funciones regulares. En particular, dada una función $f \in C^{n+1}([x_0, x_n])$ y su función de interpolación de Lagrange, $P_n(x)$, sobre los nodos

$$x_0 < x_1 < x_2 < \cdots < x_n$$

se demostró que para todo x en el intervalo $[x_0, x_n]$ existe un punto ξ_x en ese mismo intervalo para el cual se verifica

$$f(x) - P_n(x) \;=\; \frac{1}{(n+1)!}\, f^{(n+1}(\xi_x)\, \Pi_{i=0}^{n}(x - x_i)$$

Es claro que este resultado permite obtener una estimación del error de integración puesto que

$$e = \int_a^b f(x)\, dx - \int_a^b P_n(x)\, dx = \frac{1}{(n+1)!}\, \int_a^b f^{(n+1}(\xi_x)\, \Pi_{i=0}^{n}(x - x_i)\, dx$$

y basta, entonces, con acotar la integral (a partir de una acotación de la derivada de f).

En la práctica, es conveniente manejar esta acotación de error con cuidado a fin de no sobrestimar el error de integración. Por ejemplo, los errores debidos a las oscilaciones de $P_n(x)$ (cuando se emplea un número elevado de nodos para una función cuyas derivadas de orden superior presentan valores absolutos grandes) se cancelan en cierta medida al integrar sobre el intervalo. Así, si empleamos una acotación *grosera* del error de interpolación, obtenida a partir de una cota superior de éste sobre todo el intervalo, estaremos obteniendo una cota muy elevada del error de integración que será poco útil.

Posteriormente se considerarán acotaciones *finas* (y, por lo tanto, ilustrativas del comportamiento de los métodos) de los errores de integración numérica.

7.3. Fórmulas de cuadratura simples

De acuerdo con las ideas expuestas en la sección anterior, una primera estrategia para la evaluación numérica de una integral pasa por la interpolación mediante un polinomio global sobre el intervalo de integración. Esta estrategia conduce a lo que se denominan las *fórmulas de cuadratura simples*.

Se dispondrá de una fórmula de cuadratura simple por cada elección del número de puntos (*nodos de integración*) y la localización de éstos sobre el intervalo. Se examinarán a continuación dos estrategias distintas para la localización de los puntos sobre el intervalo (dentro de cada estrategia aparecerá una familia de métodos, donde cada método corresponderá a un número de puntos)

- fórmulas de Newton-Cotes (cerradas)
- fórmulas de Gauss

7.3.1. Fórmulas de Newton-Cotes (cerradas)

Las fórmulas de Newton-Cotes (cerradas) corresponden a una elección relativamente natural de los nodos de integración: éstos se reparten de forma equiespaciada sobre el intervalo e incluyen a los extremos del intervalo.

Así, para el cálculo de la integral

$$\int_a^b f(x)\, dx$$

la fórmula de Newton-Cotes (cerrada) de $n+1$ puntos toma los *nodos de integración* $\{x_i\}_{i=0}^n$ de modo que

$$x_0 = a, \quad x_n = b \quad \text{y} \quad x_{j+1} - x_j = \frac{b-a}{n} \quad \text{para} \;\; j = 1, 2, \ldots, n-1$$

Existe una segunda familia de fórmulas, denominadas fórmulas de Newton-Cotes abiertas que también toman los nodos equiespaciados pero no incluyen ninguno de los dos extremos del intervalo.

Cálculo efectivo de los *pesos* de las fórmulas de Newton-Cotes

Aunque, como hemos visto, los pesos w_i se definen a partir de las funciones de base de interpolación de Lagrange, en la práctica se calculan de un modo distinto.

Se considera, en primer lugar, si existen algunas funciones, $f(x)$, para las cuales el error de integración sea nulo. Esto es, nos planteamos si existen algunas funciones para las cuales

$$\int_a^b f(x)\, dx \;=\; \sum_{i=0}^{n} f(x_i) w_i$$

Puesto que las fórmulas que consideramos son de tipo interpolatorio polinómico (es decir, aproximan la función por su interpolada y después integran de forma exacta el polinomio) la respuesta es fácil: se integrarán de forma exacta todos los polinomios de grado menor o igual que n.

Así las cosas tenemos $n+1$ funciones linealmente independientes (por ejemplo, los monomios de grado menor o igual que n) para las cuales la fórmula de integración numérica devuelve el valor exacto. Se puede entonces escribir un sistema de $n+1$ ecuaciones para determinar los $n+1$ pesos $\{w_i\}_{i=0}^n$.

Tomando entonces los monomios de grado menor o igual que n se obtiene

$$\int_a^b x^j\, dx \;=\; \sum_{i=0}^{n} x_i^j w_i \qquad \text{para } j = 0, 1, 2, \ldots, n$$

que permite determinar los $n+1$ pesos.

Ejemplo 7.1 *Considérese la primera fórmula de Newton-Cotes (cerrada) correspondiente a dos puntos situados en los extremos. La fórmula para este esquema (sobre el intervalo de referencia) tendrá la forma*

$$\int_{-1}^{+1} f(\zeta)\, d\zeta \;\simeq\; f(\zeta_0)\, w_0 \;+\; f(\zeta_1)\, w_1$$

con $\zeta_0 = -1$ *y* $\zeta_1 = +1$.

Se procederá a la deducción de los coeficientes w_0 *y* w_1 *imponiendo que la fórmula de integración sea exacta para los monomios* 1 *y* x*:*

$$\int_{-1}^{+1} 1\, d\zeta \;=\; 2 \;=\; w_0 \;+ w_1$$

$$\int_{-1}^{+1} \zeta\, d\zeta \;=\; 0 \;=\; -w_0 \;+ w_1$$

de donde $w_0 \;=\; w_1 \;=\; 1$.

Para calcular la integral sobre un intervalo arbitrario,

$$\int_a^b f(x)\, dx$$

la fórmula se escribe (empleando el cambio de variable $h(\zeta) = \frac{a+b}{2} + \frac{b-a}{2}\zeta$ *descrito previamente):*

$$\int_a^b f(x)\, dx \;=\; \frac{b-a}{2}\int_{-1}^{+1} f(h(\zeta))\, d\zeta \;\simeq\; \frac{b-a}{2}\left(f(h(-1)) + f(h(1))\right) \;=\; \frac{b-a}{2}\left(f(a) + f(b)\right)$$

□

En cualquier caso, los valores de los pesos para las fórmulas de Newton-Cotes (cerradas) aparecen tabuladas en la mayor parte de los textos de métodos numéricos de modo que en la práctica no es preciso calcularlos.

Algunas fórmulas de Newton-Cotes (cerradas)

A continuación se recogen las fórmulas de Newton-Cotes (cerradas) asociadas a un número reducido de nodos de integración, n, donde h representa la separación entre nodos ($h = \frac{b-a}{n}$):

- Fórmula del trapecio ($n = 1$):

$$\int_a^b f(x)\, dx \;\simeq\; \frac{h}{2}\left(f(a) + f(b)\right)$$

- Fórmula de Simpson ($n = 2$):

$$\int_a^b f(x)\, dx \;\simeq\; \frac{h}{3}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right)$$

- Fórmula 3/8 ($n = 3$):

$$\int_a^b f(x)\,dx \simeq \frac{3h}{8}\left(f(a) + 3f(\frac{2a+b}{3}) + 3f(\frac{a+2b}{3}) + f(b)\right)$$

- Fórmula de Boole ($n = 4$):

$$\int_a^b f(x)\,dx \simeq \frac{2h}{45}\left(7f(a) + 32f(\frac{3a+b}{4}) + 12f(\frac{a+b}{2}) + 32f(\frac{a+3b}{4}) + 7f(b)\right)$$

Ejemplo 7.2 *Se considera ahora la aplicación de estas fórmulas al cálculo de la siguiente integral*

$$\int_0^{\pi/2} \cos x\,dx$$

cuyo valor exacto es 1. *A continuación se muestran los resultados obtenidos (reteniendo sólo los cuatro primeros decimales):*

- *Fórmula del trapecio:*

$$I_1 = \frac{1}{2}\,\frac{\pi}{2}\,\left(\cos(0) + \cos(\frac{\pi}{2})\right) \simeq 0.7854$$

- *Fórmula de Simpson:*

$$I_2 = \frac{1}{3}\,\frac{\pi}{4}\,\left(\cos(0) + 4\cos(\frac{\pi}{4}) + \cos(\frac{\pi}{2})\right) \simeq 1.0023$$

- *Fórmula 3/8:*

$$I_2 = \frac{3}{8}\,\frac{\pi}{6}\,\left(\cos(0) + 3\cos(\frac{\pi}{6}) + 3\cos(\frac{\pi}{3}) + \cos(\frac{\pi}{2})\right) \simeq 1.0010$$

- *Fórmula de Boole:*

$$I_2 = \frac{2}{45}\,\frac{\pi}{8}\,\left(7\cos(0) + 32\cos(\frac{\pi}{8}) + 12\cos(\frac{\pi}{4}) + 32\cos(\frac{3\pi}{8}) + 7\cos(\frac{\pi}{2})\right) \simeq 1.0000$$

En la figura 7.1 se muestra la función de interpolación empleada por cada una de las cuatro fórmulas. El área sombreada representa el valor calculado para la aproximación de la integral mediante la correspondiente fórmula.

□

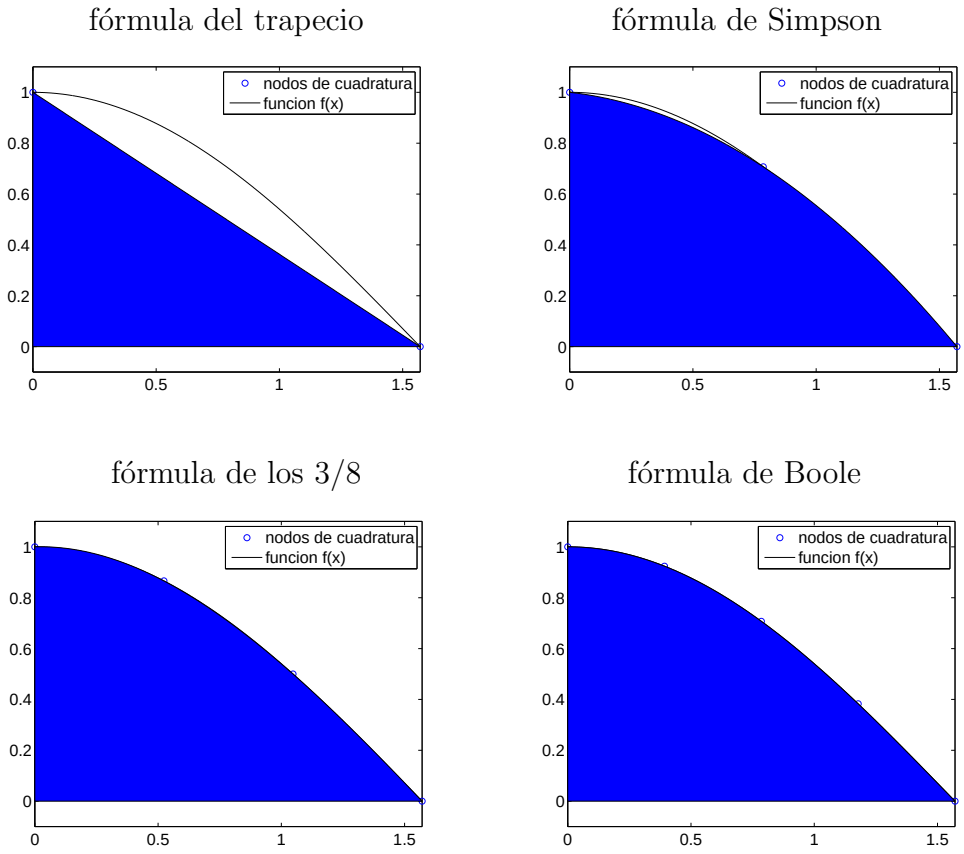

Figura 7.1: Aproximación de la integral $\int_0^{\pi/2} \cos(x)\, dx$ mediante las cuatro primeras fórmulas de Newton-Cotes (cerradas): fórmula del trapecio ($n = 1$), fórmula de Simpson ($n = 2$), fórmula 3/8 ($n = 3$) y fórmula de Boole ($n = 4$). El área sombreada corresponde al área comprendida entre la gráfica de la función de interpolación y el eje de abscisas (y, por lo tanto, a la integral numérica calculada). Obsérvese que la gráfica correspondiente a la fórmula del trapecio permite interpretar la expresión del error para esta fórmula en función de la longitud del intervalo y los valores de la derivada segunda de la función.

Estimación del error de integración

Retomando la estimación de error de integración de carácter general expuesta anteriormente, se plantea el cálculo del error de integración para las fórmulas de Newton-Cotes (cerradas).

a) Fórmula del trapecio.

La expresión general de error devuelve en este caso:

$$e = \int_a^b f(x)\,dx - \int_a^b P_1(x)\,dx = \frac{1}{2}\int_a^b f''(\xi_x)\,(x-a)(x-b)\,dx$$

A fin de estimar el valor de la integral, se recuerda que el teorema del valor medio (generalizado) del cálculo integral asegura que si F es una función continua sobre un cierto intervalo $[a,b]$ y G una función integrable que no cambia de signo sobre dicho intervalo, entonces existe $\zeta \in (a,b)$ tal que

$$\int_a^b F(x)G(x)\,dx = F(\zeta)\int_a^b G(x)\,dx$$

Así, empleando el teorema del valor medio (generalizado) del cálculo integral (suponiendo que f es de clase C^2 y empleando que $(x-a)(x-b)$ es una función no positiva sobre el intervalo), se tiene

$$e = \frac{1}{2}f''(\xi)\int_a^b (x-a)(x-b)\,dx = -\frac{1}{12}(b-a)^3 f''(\zeta)$$

Dicha estimación de error muestra la dependencia del error con respecto a la longitud del intervalo de integración y la regularidad de la función que se integra. Obsérvese que esta información ya podía esperarse de la interpretación gráfica del método (a la que debe lógicamente su nombre).

b) Caso general.

Con alguna modificación, la demostración anterior puede extenderse al caso general. Es preciso, no obstante, observar que existe una diferencia entre los casos con números pares e impares de puntos de integración. En particular, el signo de $\Pi_{i=0}^n(x-x_i)$ depende de la paridad de n. Además se tiene que $\int_a^b \Pi_{i=0}^n(x-x_i)\,dx$ es 0 si n es par, lo que no ocurre si n es impar.

Se tiene así el siguiente resultado:

Teorema 7.1 *(Error de integración de las fórmulas de Newton-Cotes (cerradas))*
Sea $f : [a,b] \to \mathbf{R}$ integrable y sea

$$E_n = \int_a^b f(x)\,dx - \sum_{i=0}^{n} f(x_i)w_i$$

el error de integración numérica mediante la fórmula de Newton-Cotes (cerrada) de $n+1$ puntos. Sea $h = \frac{b-a}{n}$ el equiespaciado entre los nodos.

a) Si n es par y $f \in C^{n+2}([a,b])$, entonces existen $C_n < 0$ y $\zeta_n \in [a,b]$ tales que:

$$E_n = \frac{C_n}{(n+2)!} h^{n+3} f^{(n+2}(\zeta_n)$$

b) Si n es impar y $f \in C^{n+1}([a,b])$, entonces existen $C_n < 0$ y $\zeta_n \in [a,b]$ tales que:

$$E_n = \frac{C_n}{(n+1)!} h^{n+2} f^{(n+1}(\zeta_n)$$

Demostración: *Consúltense las referencias de Isaacson-Keller o Quarteroni et al. para la demostración.*

□

Del resultado anterior se desprenden varias observaciones:

- Las fórmulas correspondientes a números impares de puntos (valores pares de n) presentan un mejor comportamiento que las asociadas a números pares de puntos (valores impares de n), habida cuenta de la potencia de h que aparece en las expresiones del error. Puesto que se dice que una fórmula de cuadratura es *de orden p* si p es el mayor número entero tal que, dada una cierta función regular f, existan unas constantes $C_f > 0$ y δ que verifiquen

  $$|E_n| \leq C_f h^p \qquad \forall\, h \in (0,\delta)$$

  el resultado anterior asegura que una fórmula de Newton-Cotes (cerrada) de $n+1$ puntos es de orden $n+3$ si n es par y solamente es de orden $n+2$ si n es impar.

- De las expresiones del error se deduce que las fórmulas asociadas a valores de impares del número de puntos (esto es, valores pares de n) no sólo son exactas para polinomios de grado menor o igual que n (puesto que en tal caso no hay error de interpolación) sino que también lo son para polinomios de grado $n+1$ (a pesar de que, en tal caso, sí hay error de interpolación) gracias a ciertas propiedades de simetría. Esta *exactitud adicional* es fácilmente reconocible en la primera fórmula de Newton-Cotes abierta (para la familia de fórmulas de Newton-Cotes abiertas se tiene un resultado equivalente al enunciado para las fórmulas cerradas) que corresponde a un esquema con un único nodo de integración en el punto medio

  $$\int_a^b f(x)\, dx \simeq (b-a) f(\frac{a+b}{2})$$

  capaz de integrar de forma exacta polinomios de grado uno. Al mismo tiempo, es sencillo probar que la fórmula del trapecio no integra de forma exacta polinomios de grado dos.

Teniendo en cuenta estas propiedades (y que los coeficientes C_n, con n par, y C_{n+1} presentan magnitudes semejantes), parece razonable emplear fundamentalmente fórmulas con un número impar de puntos de integración (y, por los tanto, con n par) ya que devuelven un error similar al siguiente método de la familia de las fórmulas de Newton-Cotes, *ahorrando* sin embargo una evaluación de f (debe tenerse en cuenta que el mayor coste computacional de las fórmulas de cuadratura es el coste de evaluación del integrando sobre los nodos).

Ejemplo 7.3 *Retomando el ejemplo anterior, se estimará el error cometido en la aproximación numérica de la integral* $I=\int_0^{\pi/2}\cos(x)\,dx$ *mediante las cuatro primeras fórmulas de Newton-Cotes (cerradas). Así, llamando de nuevo* $E_n = I - I_n$ *(con* I_n *el valor devuelto por la fórmula de* n *puntos), se obtiene*

$$\textit{fórmula del trapecio:} \quad E_1 \simeq +2.1460 \times 10^{-1}$$

$$\textit{fórmula de Simpson:} \quad E_2 \simeq -2.2799 \times 10^{-3}$$

$$\textit{fórmula de los 3/8:} \quad E_3 \simeq -1.0049 \times 10^{-3}$$

$$\textit{fórmula de Boole:} \quad E_4 \simeq 8.4345 \times 10^{-6}$$

Estos valores de error de integración reflejan fielmente la situación descrita por las estimaciones de error enunciadas:

- *como se ve, el aumento del número de nodos de integración numérica para la función coseno (para la cual todas las derivadas que aparecen en la expresión del error están acotadas, en valor absoluto, por la unidad) lleva a reducciones significativas del error*
- *además, se observa que la fórmula de Simpson (correspondiente a* $n = 2$*) y la fórmula de los 3/8 (correspondiente a* $n = 3$*) devuelven errores muy parecidos; visto de otro modo, conforme se aumenta el número de nodos de integración el error disminuye de forma muy sensible si se pasa a un número de nodos impar pero apenas lo hace si se pasa a un número de nodos par*

Por otro lado, las constantes C_n *que aparecen en el resultado de convergencia expuesto pueden, de hecho ser calculadas (véase cualquiera de las referencias mencionadas). Así, para las cuatro primeras fórmulas de Newton-Cotes (cerradas) se tiene, de forma más precisa*

$$\textit{fórmula del trapecio:} \quad E_1 = \frac{1}{12}h^3 f''(\zeta_1) \qquad \textit{con } h = b - a$$

$$\textit{fórmula de Simpson:} \quad E_2 = \frac{1}{90}h^5 f^{(4}(\zeta_2) \qquad \textit{con } h = (b-a)/2$$

$$\textit{fórmula de los 3/8:} \quad E_3 = \frac{3}{80}h^5 f^{(4}(\zeta_3) \qquad \textit{con } h = (b-a)/3$$

$$\textit{fórmula de Boole:} \quad E_4 = \frac{8}{945}h^7 f^{(6}(\zeta_4) \quad \textit{con } h = (b-a)/4$$

lo que permite obtener las siguientes acotaciones de $|E_n|$ *para el ejemplo considerado (empleando* $|f^{(k}(\zeta_n)| \leq 1$*):*

$$\textit{fórmula del trapecio:} \quad |E_1| \leq \frac{1}{12}h^3 \simeq 3.2298 \times 10^{-1}$$

$$\textit{fórmula de Simpson:} \quad |E_2| \leq \frac{1}{90}h^5 \simeq 3.3205 \times 10^{-3}$$

$$\textit{fórmula de los 3/8:} \quad |E_3| \leq \frac{3}{80}h^5 \simeq 1.4758 \times 10^{-3}$$

$$\textit{fórmula de Boole:} \quad |E_4| \leq \frac{1}{90}h^5 \simeq 1.2192 \times 10^{-5}$$

Como puede comprobarse, estas estimaciones devuelven cotas muy próximas a los errores realmente cometidos (gracias a que las derivadas de la función considerada no toma valores muy alejados de la cota superior empleada). No debe esperarse, en general, disponer de estimaciones del error a priori *tan precisas, pues en los casos de interés es raro que las derivadas de orden superior tengan tan buen comportamiento como en el ejemplo considerado.*

□

7.3.2. Fórmulas de Gauss

En algunos casos, la posición de los nodos de integración está obligada. Por ejemplo, en el muestreo de señales puede resultar complicado desde el punto de vista práctico muestrear sobre distribuciones no equiespaciadas de tiempos. En otros casos, sin embargo, puede resultar tan sencillo estimar sobre una distribución de nodos como sobre otra cualquiera. Éste último suele ser el caso cuando la variable sobre la que se integra representa algún parámetro de un cierto sistema.

En aquellos casos donde los nodos pueden distribuirse libremente sobre el intervalo, nos preguntamos si existe alguna razón para tomarlos así o si, por el contrario, sería preferible elegir alguna otra localización. La pregunta puede formularse entonces de la forma siguiente: una vez fijado el número de nodos de integración numérica que se va a emplear ¿cuál es la mejor disposición posible de estos nodos?

La respuesta a la pregunta anterior pasa por aclarar, en primer lugar, qué entendemos por *mejor disposición* de los nodos de integración. Desde luego, nuestra intención es que los errores sean lo más reducidos posible. Además, los resultados obtenidos para las fórmulas de Newton-Cotes (cerradas) dejan clara la relación entre el orden del esquema (la potencia de h que determina la magnitud del error y que condiciona la reducción de éste con el refinamiento

de la malla de nodos) y el grado de los polinomios que la fórmula es capaz de integrar de forma exacta.

Por otro lado, resulta natural esperar que la integración exacta de polinomios hasta un grado elevado por parte de una fórmula de cuadratura lleve a que los errores de integración para funciones regulares sea reducido. Esto es así porque, empleando un desarrollo truncado de Taylor de la función correspondiente a un polinomio con el máximo grado que se puede integrar de forma exacta, es claro que el error de integración proviene exclusivamente del error de integración del resto.

Nos planteamos así la búsqueda de $n+1$ nodos de integración numérica $\{x_i\}_{i=0}^n$ y $n+1$ pesos $\{w_i\}_{i=0}^n$, de forma que la fórmula de cuadratura

$$\int_{-1}^{+1} f(x)\, dx \simeq \sum_{i=0}^{n} f(x_i)\, w_i$$

sea exacta para polinomios del mayor grado posible.

Un simple recuento del número de incógnitas sugiere que el mayor grado posible será $2n+1$. El análisis riguroso del sistema de ecuaciones (no lineales) resultante demuestra que, en efecto, así es y caracteriza además los nodos como las raíces de un determinado polinomio. Pueden consultarse esta demostración en la referencia de Kincaid y Cheney.

Obsérvese que los nodos se calculan sobre el intervalo $[-1,+1]$. Para el cálculo de integrales sobre intervalos diferentes basta con aplicar el cambio de variable expuesto con anterioridad.

Finalmente, las coordenadas de los nodos y los valores de los pesos para las fórmulas de Gauss aparecen tabuladas en la mayor parte de los textos de métodos numéricos (con la precisión que corresponde a las representaciones en coma flotante con 64 bits de la norma IEEE) por lo que, en la práctica, no es necesario resolver el problema anterior para su obtención.

A continuación se muestran los valores de los nodos $\{x_i\}_{i=0}^n$ y los pesos $\{w_i\}_{i=0}^n$ para las cuatro primeras fórmulas cuadratura de Gauss

$$\int_{-1}^{+1} f(x)\, dx \simeq \sum_{i=0}^{n} f(x_i)\, w_i$$

- Fórmula de Gauss con un punto ($n=0$):

$$x_0 = 0 \quad w_0 = 2$$

- Fórmula de Gauss con dos puntos ($n=1$):

$$\begin{aligned} x_0 &= -0.577350269189626 \qquad w_0 = 1 \\ x_1 &= +0.577350269189626 \qquad w_1 = 1 \end{aligned}$$

- Fórmula de Gauss con tres puntos ($n = 2$):

$$\begin{array}{ll} x_0 = -0.774596669241483 & w_0 = 0.555555555555556 \\ x_1 = 0 & w_1 = 0.888888888888889 \\ x_2 = +0.774596669241483 & w_2 = 0.555555555555556 \end{array}$$

- Fórmula de Gauss con cuatro puntos ($n = 3$):

$$\begin{array}{ll} x_0 = -0.861136311594053 & w_0 = 0.347854845137454 \\ x_1 = -0.339981043584856 & w_1 = 0.652145154862546 \\ x_2 = +0.339981043584856 & w_2 = 0.652145154862546 \\ x_3 = +0.861136311594053 & w_3 = 0.347854845137454 \end{array}$$

Ejemplo 7.4 *Se considera de nuevo el cálculo de la integral* $\int_0^{\pi/2} \cos(x)\, dx$ *ahora mediante las fórmulas de integración numérica de Gauss. Obsérvese que el cambio de variable para transformar la integral al intervalo de referencia es* $h(\zeta) = \frac{\pi}{4}(1+\zeta)$ *de modo que*

$$\int_0^{\pi/2} \cos(x)\, dx = \frac{\pi}{4} \int_{-1}^{+1} \cos(h(\zeta))\, d\zeta$$

A continuación se muestran los resultados obtenidos con las cuatro primeras fórmulas (reteniendo ocho decimales):

- *Fórmula de Gauss con un punto:*

$$I_1 \;=\; \frac{\pi}{2} \cos(0.785398163397448) \;\simeq\; 1.11072073$$

- *Fórmula de Gauss con dos puntos:*

$$I_2 \;=\; \frac{\pi}{4} \left(\cos(0.331948322338894) + \cos(1.238848004456003) \right) \;\simeq\; 0.99847261$$

- *Fórmula de Gauss con tres puntos:*

$$\begin{aligned} I_3 \;=\; & \frac{\pi}{4} \Big(+0.555555555555556 \cos(0.177031362001407) \\ & +0.888888888888889 \cos(0.785398163397448) \\ & +0.555555555555556 \cos(1.393764964793490) \Big) \\ \simeq\; & 1.00000812 \end{aligned}$$

- *Fórmula de Gauss con cuatro puntos:*

$$\begin{aligned} I_4 \;=\; & \frac{\pi}{4}\,(\quad +0.347854845137454\,\cos(0.109063285836626) \\ & \qquad +0.652145154862546\,\cos(0.518377676175955) \\ & \qquad +0.652145154862546\,\cos(1.052418650618942) \\ & \qquad +0.347854845137454\,\cos(1.461733040958270)\;) \\ \simeq \;& 0.99999998 \end{aligned}$$

Con estos resultados los errores, $e_i = I - I_i$, cometidos con cada fórmula resultan:

- *Fórmula de Gauss con un punto:* $e_1 = -1.1072 \times 10^{-1}$
- *Fórmula de Gauss con dos puntos:* $e_2 = +1.5274 \times 10^{-3}$
- *Fórmula de Gauss con tres puntos:* $e_3 = -8.1216 \times 10^{-6}$
- *Fórmula de Gauss con cuatro puntos:* $e_4 = +2.2803 \times 10^{-8}$

Estos resultados muestran la rápida convergencia (esto es, la rápida mejora de la aproximación conforme se aumenta el número de puntos de integración numérica) de las fórmulas de integración numérica de Gauss para un caso regular. A fin de convencerse de las ventajas de las fórmulas de Gauss sobre las de Newton-Cotes, basta con comparar los errores cometidos para las funciones con el mismo número de puntos (y, por lo tanto, aproximadamente con el mismo coste computacional).

□

7.4. Fórmulas de cuadratura compuestas

La integración de funciones *no regulares* (entendiendo como tales aquellas para las cuales los valores absolutos de las derivadas de orden superior pueden ser muy grandes) dará lugar, en general, a dificultades en la convergencia de los esquemas de integración numérica. La razón es que, como ya se ha visto, los errores de interpolación pueden ser difíciles de controlar y posiblemente haya puntos sobre el intervalo donde éstos sean muy elevados. Así, aunque parte del error pueda cancelarse al calcular la integral (gracias al carácter oscilante del error de interpolación), resulta esperable que la integración numérica converja sólo lentamente y sea, por lo tanto preciso, emplear muchos nodos de integración para obtener una aproximación precisa del valor de la integral.

Ejemplo 7.5 *Se considera el cálculo de la integral*

$$\int_0^5 \frac{1}{1+x^2}\,dx$$

(cuyo valor exacto es $arctan(5) \simeq 1.3734$*) mediante fórmulas de integración numérica.*

Se toman, en primer lugar, las cinco primeras fórmulas de Newton-Cotes (cerradas) correspondientes a números de puntos de integración impares (de acuerdo con los resultados obtenidos previamente). Se obtienen, para dichas fórmulas (se representa el número de puntos mediante np), los siguientes resultados:

Fórmula de integración	*Aproximación*	*Error de integración*
n=2 (np=3)	1.3252	$+4.8246 \times 10^{-2}$
n=4 (np=5)	1.3076	$+6.5807 \times 10^{-2}$
n=6 (np=7)	1.3614	$+1.2032 \times 10^{-2}$
n=8 (np=9)	1.3762	-2.7572×10^{-3}
n=10 (np=11)	1.3759	-2.5160×10^{-3}

Obsérvese que, tal y como se esperaba, al no contar con demasiada regularidad (entendida ésta, como ya se ha hecho anteriormente, como una falta de acotaciones uniformes de los valores absolutos de las derivadas de órdenes elevados) la mejora en la aproximación de la integral no mejora sino lentamente con la incorporación de más puntos de integración. Esto es así porque la función de interpolación correspondiente no aproxima bien el integrando.

Sucesivos aumentos del número de puntos de integración podrían incluso empeorar la aproximación de la integral. Así, obsérvense en la figura ????? las funciones de interpolación con 11 *y* 21 *nodos equiespaciados (que corresponderían a las fórmulas de Newton-Cotes con* $n = 10$ *y* $n = 20$*, respectivamente).*

- *INCLUIR REPRESENTACION*

El empleo de fórmulas de Gauss puede mejorar algo los resultados, como se desprende de la siguiente tabla, pero en ningún caso proporciona un modo satisfactorio de abordar la integración numérica de funciones no regulares (entendidas éstas en el sentido varias veces referido).

Fórmula de integración	*Aproximación*	*Error de integración*
n=1 (np=2)	1.3323	$+4.1125 \times 10^{-2}$
n=2 (np=3)	1.4278	-5.4417×10^{-2}
n=3 (np=4)	1.3858	-1.2448×10^{-2}
n=4 (np=5)	1.3716	$+1.7933 \times 10^{-3}$
n=5 (np=6)	1.3721	$+1.2658 \times 10^{-3}$
n=6 (np=7)	1.3733	$+1.0180 \times 10^{-4}$
n=7 (np=8)	1.3735	-7.5357×10^{-5}
n=8 (np=9)	1.3734	-2.1626×10^{-5}

□

Así, cuando se trate con funciones *no regulares* en el sentido anterior (lo que constituirá la norma más que la excepción) la estrategia para obtener buenas aproximaciones de la integral no pasa por aumentar significativamente el número de puntos de interpolación global y, en consecuencia, el grado de los polinomios. Como ya se vió al estudiar la interpolación de funciones, esta estrategia está lejos de ser óptima. En su lugar se preferirá adoptar un enfoque similar al empleado en los problemas de interpolación: dividir el intervalo en intervalos más reducidos y considerar, sobre cada uno de ellos, una fórmula de integración numérica de tipo interpolatorio con un polinomio de grado limitado. De este modo, se confía en mejorar la aproximación mediante sucesivas subdivisiones de los intervalos, manteniendo fija la fórmula de integración numérica sobre cada subintervalo (esto es, manteniendo fijo el número de puntos de integración en cada subintervalo).

La estrategia que se acaba de describir (esto es, dividir el intervalo de integración en la unión de varios subintervalos y considerar fórmulas relativamente simples sobre cada uno de esos subintervalos) se denomina *integración numérica compuesta* y resulta aplicable, lógicamente, a cualquier método (o familia de métodos) de integración numérica.

Así, para el cálculo de

$$\int_a^b f(x)\,dx$$

se considera una distribución de puntos $\{z_i\}_{i=0}^{NI}$ con

$$a = z_0 < z_1 < \cdots < z_{NI-1} < z_{NI}$$

y una cierta fórmula de integración numérica (de orden relativamente bajo). Se denomina *fórmula de integración numérica compuesta asociada a la fórmula de integración considerada* al esquema que descompone

$$\int_a^b f(x)\,dx \;=\; \sum_{k=0}^{NI} \int_{z_{i-1}}^{z_i} f(x)\,dx$$

e integra, empleando la fórmula de integración numérica que se considere, sobre cada uno de los intervalos.

Se intenta entonces, de acuerdo con las observaciones ya hechas de la dificultad de convergencia de los polinomios de grado elevado en la interpolación de funciones no muy regulares, que la convergencia de la aproximación de la integral quede garantizada por la reducción de los intervalos de integración (subdividiendo éstos tanto como sea necesario) y no por el aumento del orden de los polinomios de interpolación (que puede generar malas aproximaciones de la función que se reflejen en una mala aproximación de la integral).

A continuación se considera la aplicación de esta técnica a las fórmulas de Newton-Cotes (cerradas) y a las fórmulas de Gauss.

7.4.1. Fórmulas de Newton-Cotes (cerradas) compuestas

De acuerdo con la idea general de las fórmulas compuestas expuesta anteriormente, se denominará *fórmula de Newton-Cotes (cerrada) compuesta de $n+1$ puntos y NI intervalos* para el cálculo de la integral

$$\int_a^b f(x)\,dx$$

a la fórmula que resulta de dividir el intervalo $[a,b]$ en NI intervalos mediante los puntos $\{z_i\}_{i=0}^{NI}$

$$a = z_0 < z_1 < \cdots < z_{NI-1} < z_{NI} = b$$

descomponer la integral en integrales sobre cada intervalo y emplear la fórmula de Newton-Cotes (cerrada) de $n+1$ puntos sobre cada subintervalo

$$\int_{z_{i-1}}^{z_i} f(x)\,dx \;\simeq\; \sum_{j=0}^{n} f(x_j^i)\,w_j$$

donde los puntos $\{x_j^i\}_{j=0}^{n}$ representan los nodos de la fórmula de cuadratura de Newton-Cotes (cerrada) de $n+1$ puntos sobre el intervalo $[z_{i-1}, z_i]$.

Aunque, en principio, los puntos $\{z_i\}_{i=0}^{NI}$ que subdividen los intervalos podrían tomarse de forma no equiespaciada, es habitual que se repartan sobre el intervalo con una separación constante entre cada pareja de puntos consecutivos (se hablará entonces de una fórmula compuesta con paso fijo). Obsérvese que ahora no hay ninguna colocación óptima de los nodos que sea independiente de la función f que se desea integrar. En cualquier caso, en las denominadas técnicas de integración numérica adaptativa (véase la siguiente sección) sí se considerará la posibilidad de emplear una distribución no homogénea de nodos que sea adecuada para la integración de una función particular.

A continuación se muestran, como ejemplo, las dos primeras fórmulas de Newton-Cotes (cerradas) compuestas con paso $h = \dfrac{x_{j+1}^i - x_j^i}{n}$ fijo (tanto para i como para j):

- Fórmula del trapecio ($n=1$) compuesta:

$$\int_a^b f(x)\,dx \;\simeq\; \frac{h}{2}\sum_{i=1}^{NI}\Big(f(x_0^i)+f(x_1^i)\Big) \;=\; \frac{h}{2}f(z_0) + h\sum_{i=1}^{NI-1} f(z_i) + \frac{h}{2}f(z_{NI})$$

- Fórmula de Simpson ($n=2$) compuesta:

$$\begin{aligned}\int_a^b f(x)\,dx \;&\simeq\; \frac{h}{3}\sum_{i=1}^{NI}\Big(f(x_0^i)+4f(x_1^i)+f(x_2^i)\Big)\\ &=\; \frac{h}{3}f(z_0) + \frac{2h}{3}\sum_{i=1}^{NI-1} f(z_i) + \frac{4h}{3}\sum_{i=1}^{NI-1} f\Big(\frac{z_{i-1}+z_i}{2}\Big) + \frac{h}{3}f(z_{NI})\end{aligned}$$

El análisis del error cometido por las fórmulas de integración numérica compuestas es inmediato pues se reduce, sobre cada intervalo, al análisis de las correspondientes fórmulas simples.

De este modo, entre los dos esquemas que se acaban de mostrar como ejemplo, queda claro que resulta preferible (al menos para funciones relativamente regulares, donde se tiene una acotación modesta de la derivada cuarta del integrando) el esquema basado en la fórmula de Simpson ya que, como se vió previamente, la fórmula del trapecio sobre cada intervalo presenta un orden 3 en tanto que la fórmula de Simpson tiene un orden 5. Así, comparando la estimación de error para dos esquemas basadas en trapecio compuesta y Simpson compuesta con el mismo número de evaluaciones (lo que impondrá que el paso empleado en la fórmula de Simpson ha de ser el doble del paso usado en la fórmula del trapecio) se comprueba que la cota de error global para el esquema del trapecio compuesto

$$|E_t| \;=\; \sum_{i=1}^{NI_t} |E_t^i| \;=\; \sum_{i=1}^{NI_t} \frac{h_t^3}{12} |f''(\zeta_t^i)| \;\leq\; \frac{h_t^2}{12}(b-a)M_2$$

(donde M_2 representa una cota para el valor absoluto de la derivada segunda sobre todo el intervalo de integración) será fácilmente mejorada por la correspondiente cota para la fórmula de Simpson compuesta

$$|E_S| \;=\; \sum_{i=1}^{NI_S} |E_S^i| \;=\; \sum_{i=1}^{NI_S} \frac{h_S^5}{90} |f^{(iv}(\zeta_S^i)| \;\leq\; \frac{h_S^4}{90}(b-a)M_4 \;=\; \frac{4h_t^4}{45}(b-a)M_4$$

(donde M_4 representa una cota para el valor absoluto de la derivada cuarta sobre todo el intervalo de integración) siempre que la función sea relativamente regular y la división en subintervalos suficientemente fina.

Obsérvese que un razonamiento parecido llevará a preferir la fórmula de Simpson compuesta a la fórmula de los 3/8 compuesta. En este caso, la evaluación adicional no permite mejorar el orden y puede ser empleada de forma más eficiente en reducir los intervalos en la fórmula de Simpson compuesta.

La fórmula de Simpson compuesta es (por las razones apuntadas aquí junto con el hecho de ser un esquema que no emplea interpolación con grados muy altos y evita en consecuencia las posibles oscilaciones del polinomio interpolador cuando se trate de integrar funciones no muy regulares) un esquema frecuentemente ampliado y constituye, con frecuencia, la base de los esquemas adaptativos que se verán más adelante.

Ejemplo 7.6 *Se retoma el cálculo de la integral*

$$\int_0^5 \frac{1}{1+x^2}\, dx$$

empleando ahora fórmulas de cuadratura compuestas.

Así, por ejemplo, el empleo de la fórmula de Simpson compuesta conduce a la siguiente tabla de aproximaciones

Número de intervalos	*Aproximación*	*Error de integración*
3	1.3509	$+2.2500 \times 10^{-2}$
4	1.3667	$+6.7412 \times 10^{-3}$
5	1.3715	$+1.9468 \times 10^{-3}$
6	1.3728	$+5.5672 \times 10^{-4}$
7	1.3732	$+1.5890 \times 10^{-4}$
8	1.3734	$+4.5408 \times 10^{-5}$

que pone de manifiesto el carácter convergente del refinamiento en el número de subintervalos en que se divide el intervalo de integración (a diferencia del aumento de puntos de integración en las fórmulas de cuadratura simples).

□

7.4.2. Fórmulas de Gauss compuestas

Del mismo modo que se ha procedido a la combinación de fórmulas de Newton-Cotes simples sobre subintervalos para obtener una fórmula compuesta, se hará para las fórmulas de Gauss.

Así, se denominará *fórmula de Gauss compuesta de* $n+1$ *puntos y* NI *intervalos* para el cálculo de la integral

$$\int_a^b f(x)\, dx$$

a la fórmula que resulta de dividir el intervalo $[a,b]$ en NI intervalos mediante los puntos $\{z_i\}_{i=0}^{NI}$

$$a = z_0 < z_1 < \cdots < z_{NI-1} < z_{NI} = b$$

descomponer la integral en integrales sobre cada intervalo y emplear la fórmula de Gauss de $n+1$ puntos sobre cada subintervalo

$$\int_{z_{i-1}}^{z_i} f(x)\, dx \;\simeq\; \sum_{j=0}^{n} f(x_j^i)\, w_j$$

donde los puntos $\{x_j^i\}_{j=0}^{n}$ representan los nodos de la fórmula de cuadratura de Gauss de $n+1$ puntos sobre el intervalo $[z_{i-1}, z_i]$.

Por ejemplo, las dos primeras fórmulas de Gauss compuestas con paso $h = \dfrac{z_{i+1} - z_i}{NI}$ fijo, resultan:

- Fórmula de Gauss de un punto ($n = 0$) compuesta:

$$\int_a^b f(x)\, dx \;\simeq\; h \sum_{i=1}^{NI} f(\frac{z_{i-1} + z_i}{2})$$

- Fórmula de Gauss de dos puntos ($n = 1$) compuesta:

$$\int_a^b f(x)\,dx \simeq \frac{h}{2}\sum_{i=1}^{NI}\left(f(\frac{z_{i-1}+z_i}{2}+\frac{z_i-z_{i-1}}{2}x_0)+f(\frac{z_{i-1}+z_i}{2}+\frac{z_i-z_{i-1}}{2}x_1)\right)$$

  para los nodos x_0 y x_1 definidos en la correspondiente fórmula de Gauss de dos puntos (simple)

Todos los comentarios hechos, en relación con el análisis del error de los esquemas compuestos, sobre las fórmulas de Newton-Cotes compuestas pueden extenderse al caso de las fórmulas de Gauss compuestas.

Por otro lado, las mejores propiedades de convergencia de las fórmulas de Gauss (sobre las fórmulas de Newton-Cotes) sugieren el empleo de las fórmulas compuestas de Gauss siempre que esto sea posible, aunque ahora las mejoras no son tan notables como en el caso de las fórmulas simples. A continuación se muestra un ejemplo que ilustra dicho comportamiento.

Ejemplo 7.7 *Se retoma, una vez más, el cálculo de la integral*

$$\int_0^5 \frac{1}{1+x^2}\,dx$$

empleando fórmulas de Gauss compuestas.

El uso de la fórmula de Gauss de tres puntos compuesta conduce, por ejemplo, a la siguiente tabla de aproximaciones (donde el error se mide, al igual que en anteriores ocasiones, como $I - I_{num}$):

Número de intervalos	*Aproximación*	*Error de integración*
1	1.4154	$+4.2028 \times 10^{-2}$
2	1.3774	$+3.9592 \times 10^{-3}$
3	1.3723	-1.1346×10^{-3}
4	1.3733	-6.3883×10^{-5}
5	1.3734	$+2.8339 \times 10^{-5}$

que, efectivamente, mejoran los correspondientes a la fórmula de Simpson compuesta. Sin embargo, debe tenerse en cuenta que las fórmulas de Gauss compuestas no permiten emplear las evaluaciones de la función para la aproximación de la integral en varios subintervalos (algo que sí hacen las fórmulas de Newton-Cotes cerradas con las evaluaciones en los extremos). Así, el último resultado de la tabla correspondiente a la fórmula de Gauss de tres puntos compuesta (que divide el intervalo en 5 *subintervalos) requiere* 15 *evaluaciones del integrando, mientras que el último resultado en la tabla para la fórmula del trapecio compuesta (que divide el intervalo en* 8 *subintervalos) necesita* 17 *evaluaciones.*

□

7.5. Fórmulas de cuadratura adaptativas

En la práctica, es necesario disponer de un control del error de integración numérica que permita estimar la precisión con la que se calcula la integral.

Por otro lado, en el empleo de las fórmulas de integración compuestas (que serán las que habitualmente se empleen, por las razones ya expuestas) no existe una subdivisión óptima general de los intervalos (ya que la integración sobre cada subintervalo es independiente del resto) pero sí es claro que la división uniforme del intervalo no constituye una buena elección para la mayor parte de las funciones. Por el contrario, deberían emplearse subintervalos reducidos donde la función que se integra varíe de forma apreciable y subintervalos más extensos donde ésta sea prácticamente constante. Sin embargo, quizás no se tenga una información precisa sobre este comportamiento que permita elegir la subdivisión del intervalo antes de proceder a la integración mediante una fórmula compuesta (incluso cuando se tenga esta información parece conveniente pensar en automatizar este proceso).

Las denominadas fórmulas de integración numérica adaptativa tratan de responder a ambas cuestiones. La idea general de este tipo de técnicas pasa por emplear una sucesión de fórmulas de cuadratura compuestas que permitan estimar el error cometido en la integración sobre cada subintervalo, refinando aquellos subintervalos donde los errores de integración no permiten alcanzar una cierta precisión global.

A continuación se presenta, para fijar las ideas, un ejemplo basado en el cálculo mediante la fórmula de Simpson adaptativa (pueden consultarse los detalles en el texto de Kincaid y Cheney) de la integral

$$\int_a^b f(x)\,dx$$

Supongamos que, en la etapa n-ésima del algoritmo adaptativo, se ha efectuado una integración numérica con la fórmula de Simpson compuesta sobre una cierta subdivisión del intervalo de integración $[a,b]$ que incluye un intervalo $[z_{i-1}^n, z_i^n]$. La contribución de dicho intervalo a la integral total será

$$I_i^n = \frac{h_i^n}{3}\left(f(z_{i-1}^n) + 4f(\frac{z_{i-1}^n + z_i^n}{2}) + f(z_i^n)\right)$$

donde $h_i^n = \dfrac{z_i^n - z_{i-1}^n}{2}$ y deseamos estimar el error cometido en dicha integración, a fin de saber si resulta o no aceptable y, en consecuencia, si debería o no refinarse este subintervalo.

Desde luego, contamos con una estimación *a priori* del error que asegura que:

$$e_i^n = \int_{z_{i-1}^n}^{z_i^n} f(x)\,dx - I_i^n = -\frac{(h_i^n)^5}{90} f^{(iv}(\zeta_i^n)$$

pero dicha estimación no resulta útil al no disponer, en general, de una buena estimación de $f^{(iv}(\zeta_n^i)$.

Supongamos que dividimos el intervalo $[z_{i-1}^n, z_i^n]$ en dos partes iguales y aplicamos en cada una de ellas la fórmula de Simpson para evaluar la integral. Denotaremos mediante

$I_{i,1}^{n+1}$ e $I_{i,2}^{n+1}$ las correspondientes aproximaciones de la integral. La estimación *a priori* del error de integración será ahora

$$e_i^{n+1} = \int_{z_{i-1}^n}^{z_i^n} f(x)\,dx - I_{i,1}^{n+1} - I_{i,2}^{n+1} = -\frac{(h_{i,1}^{n+1})^5}{90} f^{(iv}(\zeta_{i,1}^{n+1}) - \frac{(h_{i,2}^{n+1})^5}{90} f^{(iv}(\zeta_{i,2}^{n+1})$$

que, en principio, presenta la misma limitación que en la etapa n.

Sin embargo, si suponemos que el intervalo $[z_{i-1}^n, z_i^n]$ es reducido se tendrá que

$$f^{(iv}(\zeta_i^n) \simeq f^{(iv}(\zeta_{i,1}^{n+1}) \simeq f^{(iv}(\zeta_{i,1}^{n+1})$$

y empleando además que $h_{i,1}^{n+1} = h_{i,2}^{n+1} = \frac{1}{2} h_i^n$ se tiene

$$e_i^{n+1} \simeq \frac{1}{2^4} e_i^n$$

Del resultado anterior, junto con las expresiones que definen el error, se desprende

$$\int_{z_{i-1}^n}^{z_i^n} f(x)\,dx = I_i^n + e_i^n \simeq I_i^n + 16 e_i^{n+1}$$

$$\int_{z_{i-1}^n}^{z_i^n} f(x)\,dx = I_{i,1}^{n+1} + I_{i,2}^{n+1} + e_i^{n+1}$$

y basta restar ambas expresiones para obtener una estimación del error (para la evaluación de la integral correspondiente a la etapa $n+1$)

$$e_i^{n+1} \simeq \frac{1}{15}(I_{i,1}^{n+1} + I_{i,2}^{n+1} - I_i^n)$$

Dicha estimación servirá entonces (tras la comparación con la adecuada tolerancia) para determinar si es necesario o no dividir, en la etapa $n+2$, los dos subintervalos generados en la etapa $n+1$. Cuando, tras una cierta etapa, la estimación del error devuelva valores más reducidos que la tolerancia el proceso global se detendrá y devolverá una aproximación de la integral sumando las contribuciones de todos los subintervalos presentes tras esa última etapa.

7.6. Algunas extensiones

En este tema se han obviado dos cuestiones importantes relativas a los métodos de integración numérica, que son:

- métodos para el cálculo de integrales en varias variables

- métodos para el cálculo de integrales impropias correspondientes a integrandos no acotados
- métodos para el cálculo de integrales impropias correspondientes a intervalos no acotados

En todos los casos es preciso extender de algún modo las técnicas expuestas aquí para abordar su resolución, ya que la aplicación directa de las técnicas expuestas anteriormente no es posible o no conduce a resultados satisfactorios.

En primer lugar, la forma más simple de abordar la extensión al caso de integrales múltiples pasa (al igual que en el cálculo analítico) por aplicar el teorema de Fubini y reducir así el problema al cálculo (anidado) de varias integrales en una variable.

Por otro lado, en relación con las integrales impropias asociadas a integrandos no acotados, la idea es extraer la componente singular de la función que se integra calculando analíticamente la correspondiente contribución e integrando mediante las técnicas ya expuestas el resto (que ya es una función acotada) .

Finalmente, para la integración sobre intervalos no acotados, aunque es posible una alternativa elemental como es integrar sobre un intervalo suficientemente grande como para que la contribución que se desprecia sea muy reducido, la técnica habitual pasa por emplear un cambio de variable que convierta el intervalo de integración en un intervalo acotado (a costa de hacer que el integrando no lo sea) y entonces tratar la integral como en el caso anterior.

Puede consultarse una presentación introductoria de todas estas extensiones, por ejemplo, en la referencia de Burden y Faires o en la de Quarteroni et al.

7.7. Códigos disponibles

Aunque existe abundante *software* relativo a los métodos numéricos para la aproximación de integrales, merece especial consideración la biblioteca de programas `QUADPACK`.

`QUADPACK` (que constituye una parte del código `SLATEC`) es una colección de programas, escritos en en fortran 90, para la aproximación de integrales (propias o impropias) en una variable mediante fórmulas de Gauss. Los programas fuente pueden encontrarse, por ejemplo, en la dirección

`http://www.netlib.org/quadpack/`

o también, junto con unas breves explicaciones y algunos ejemplos, en

`http://www.csit.fsu.edu/~burkardt/f_src/quadpack/quadpack.html`

Las funciones de `QUADPACK` están también incorporados en la biblioteca GNU Scientific Library (`http://www.gnu.org/software/gsl/`) y alguna de ellas está disponible en Octave.

Pueden encontrarse asimismo otros programas que incorporan métodos numéricos para la aproximación de integrales mediante búsquedas en NETLIB (`http://www.netlib.org/`)

o GAMS (`http://gams.nist.gov/`).

7.8. Referencias

- **D. Kincaid; W. Cheney; *Análisis Numérico. Las Matemáticas del Cálculo Científico*. Addison-Wesley Iberoamericana, 1994.**

  Existe una edición actualizada de este texto titulada *Numerical Analysis: Mathematics of Scientific Computing. 3rd ed.* publicada por Brooks/Cole en 2002, de la que no existe aún traducción.

- **E. Isaacson; H.B. Keller; *Analysis of Numerical Methods*. Wiley, 1966.**

- **A. Quarteroni; R. Sacco; F. Saleri; *Numerical Mathematics*. Springer, 2000.**

- **R.L. Burden; J.D. Faires; *Análisis Numérico*. 7a ed. International Thomson Editores, 2003.**

Licencia Creative Commons

http://creativecommons.org

Reconocimiento-CompartirIgual 2.5 España

Licencia

La obra (según se define más adelante) se proporciona bajo los términos de esta licencia pública de Creative Commons ("ccpl" o "licencia"). La obra se encuentra protegida por la ley española de propiedad intelectual y/o cualesquiera otras normas resulten de aplicación. Queda prohibido cualquier uso de la obra diferente a lo autorizado bajo esta licencia o lo dispuesto en las leyes de propiedad intelectual.

Mediante el ejercicio de cualquier derecho sobre la obra, usted acepta y consiente las limitaciones y obligaciones de esta licencia. El licenciador le cede los derechos contenidos en esta licencia, siempre que usted acepte los presentes términos y condiciones.

1. Definiciones

a. La "obra" es la creación literaria, artística o científica ofrecida bajo los términos de esta licencia.

b. El "autor" es la persona o la entidad que creó la obra.

c. Se considerará "obra conjunta" aquella susceptible de ser incluida en alguna de las siguientes categorías:

   i. "Obra en colaboración", entendiendo por tal aquella que sea resultado unitario de la colaboración de varios autores.

   ii. "Obra colectiva", entendiendo por tal la creada por la iniciativa y bajo la coordinación de una persona natural o jurídica que la edite y divulgue bajo su nombre y que esté constituida por la reunión de aportaciones de diferentes autores cuya

contribución personal se funde en una creación única y autónoma, para la cual haya sido concebida sin que sea posible atribuir separadamente a cualquiera de ellos un derecho sobre el conjunto de la obra realizada.

iii. "Obra compuesta e independiente", entendiendo por tal la obra nueva que incorpore una obra preexistente sin la colaboración del autor de esta última.

d. Se considerarán "obras derivadas" aquellas que se encuentren basadas en una obra o en una obra y otras preexistentes, tales como: las traducciones y adaptaciones; las revisiones, actualizaciones y anotaciones; los compendios, resúmenes y extractos; los arreglos musicales y, en general, cualesquiera transformaciones de una obra literaria, artística o científica, salvo que la obra resultante tenga el carácter de obra conjunta en cuyo caso no será considerada como una obra derivada a los efectos de esta licencia. Para evitar la duda, si la obra consiste en una composición musical o grabación de sonidos, la sincronización temporal de la obra con una imagen en movimiento ("synching") será considerada como una obra derivada a los efectos de esta licencia.

e. Tendrán la consideración de "obras audiovisuales" las creaciones expresadas mediante una serie de imágenes asociadas, con o sin sonorización incorporada, así como las composiciones musicales, que estén destinadas esencialmente a ser mostradas a través de aparatos de proyección o por cualquier otro medio de comunicación pública de la imagen y del sonido, con independencia de la naturaleza de los soportes materiales de dichas obras.

f. El "licenciador" es la persona o la entidad que ofrece la obra bajo los términos de esta licencia y le cede los derechos de explotación de la misma conforme a lo dispuesto en ella.

g. "Usted" es la persona o la entidad que ejercita los derechos cedidos mediante esta licencia y que no ha violado previamente los términos de la misma con respecto a la obra, o que ha recibido el permiso expreso del licenciador de ejercitar los derechos cedidos mediante esta licencia a pesar de una violación anterior.

h. La "transformación" de una obra comprende su traducción, adaptación y cualquier otra modificación en su forma de la que se derive una obra diferente. Cuando se trate de una base de datos según se define más adelante, se considerará también transformación la reordenación de la misma. La creación resultante de la transformación de una obra tendrá la consideración de obra derivada.

i. Se entiende por "reproducción" la fijación de la obra en un medio que permita su comunicación y la obtención de copias de toda o parte de ella.

j. Se entiende por "distribución" la puesta a disposición del público del original o copias de la obra mediante su venta, alquiler, préstamo o de cualquier otra forma.

k. Se entenderá por "comunicación pública" todo acto por el cual una pluralidad de personas pueda tener acceso a la obra sin previa distribución de ejemplares a cada una de

ellas. No se considerará pública la comunicación cuando se celebre dentro de un ámbito estrictamente doméstico que no esté integrado o conectado a una red de difusión de cualquier tipo. A efectos de esta licencia se considerará comunicación pública la puesta a disposición del público de la obra por procedimientos alámbricos o inalámbricos, incluida la puesta a disposición del público de la obra de tal forma que cualquier persona pueda acceder a ella desde el lugar y en el momento que elija.

l. La "explotación" de la obra comprende su reproducción, distribución, comunicación pública y transformación.

m. Tendrán la consideración de "bases de datos" las colecciones de obras ajenas, de datos o de otros elementos independientes como las antologías y las bases de datos propiamente dichas que por la selección o disposición de sus contenidos constituyan creaciones intelectuales, sin perjuicio, en su caso, de los derechos que pudieran subsistir sobre dichos contenidos.

n. Los "elementos de la licencia" son las características principales de la licencia según la selección efectuada por el licenciador e indicadas en el título de esta licencia: Reconocimiento de autoría (Reconocimiento), Compartir de manera igual (CompartirIgual).

2. Límites y uso legítimo de los derechos. Nada en esta licencia pretende reducir o restringir cualesquiera límites legales de los derechos exclusivos del titular de los derechos de propiedad intelectual de acuerdo con la Ley de Propiedad Intelectual o cualesquiera otras leyes aplicables, ya sean derivados de usos legítimos, tales como el derecho de copia privada o el derecho a cita, u otras limitaciones como la derivada de la primera venta de ejemplares.

3. Concesión de licencia. Conforme a los términos y a las condiciones de esta licencia, el licenciador concede (durante toda la vigencia de los derechos de propiedad intelectual) una licencia de ámbito mundial, sin derecho de remuneración, no exclusiva e indefinida que incluye la cesión de los siguientes derechos:

a. Derecho de reproducción, distribución y comunicación pública sobre la obra.

b. Derecho a incorporarla en una o más obras conjuntas o bases de datos y para su reproducción en tanto que incorporada a dichas obras conjuntas o bases de datos.

c. Derecho para efectuar cualquier transformación sobre la obra y crear y reproducir obras derivadas.

d. Derecho de distribución y comunicación pública de copias o grabaciones de la obra, como incorporada a obras conjuntas o bases de datos.

e. Derecho de distribución y comunicación pública de copias o grabaciones de la obra, por medio de una obra derivada.

f. Para evitar la duda, sin perjuicio de la preceptiva autorización del licenciador, y especialmente cuando la obra se trate de una obra audiovisual, el licenciador renuncia

al derecho exclusivo a percibir, tanto individualmente como mediante una entidad de gestión de derechos, o varias, (por ejemplo: SGAE, Dama, VEGAP), los derechos de explotación de la obra, así como los derivados de obras derivadas, conjuntas o bases de datos, si dicha explotación pretende principalmente o se encuentra dirigida hacia la obtención de un beneficio mercantil o la remuneración monetaria privada.

Los anteriores derechos se pueden ejercitar en todos los medios y formatos, tangibles o intangibles, conocidos o por conocer. Los derechos mencionados incluyen el derecho a efectuar las modificaciones que sean precisas técnicamente para el ejercicio de los derechos en otros medios y formatos. Todos los derechos no cedidos expresamente por el licenciador quedan reservados.

4. Restricciones. La cesión de derechos que supone esta licencia se encuentra sujeta y limitada a las restricciones siguientes:

a. Usted puede reproducir, distribuir o comunicar públicamente la obra solamente bajo los términos de esta licencia y debe incluir una copia de la misma, o su Identificador Uniforme de Recurso (URI), con cada copia o grabación de la obra que usted reproduzca, distribuya o comunique públicamente. Usted no puede ofrecer o imponer ningún término sobre la obra que altere o restrinja los términos de esta licencia o el ejercicio de sus derechos por parte de los cesionarios de la misma. Usted no puede sublicenciar la obra. Usted debe mantener intactos todos los avisos que se refieran a esta licencia y a la ausencia de garantías. Usted no puede reproducir, distribuir o comunicar públicamente la obra con medidas tecnológicas que controlen el acceso o uso de la obra de una manera contraria a los términos de esta licencia. Lo anterior se aplica a una obra en tanto que incorporada a una obra conjunta o base de datos, pero no implica que éstas, al margen de la obra objeto de esta licencia, tengan que estar sujetas a los términos de la misma. Si usted crea una obra conjunta o base de datos, previa comunicación del licenciador, usted deberá quitar de la obra conjunta o base de datos cualquier crédito requerido en el apartado 4c, según lo que se le requiera y en la medida de lo posible. Si usted crea una obra derivada, previa comunicación del licenciador, usted deberá quitar de la obra derivada cualquier crédito requerido en el apartado 4c, según lo que se le requiera y en la medida de lo posible.

b. Usted puede reproducir, distribuir o comunicar públicamente una obra derivada solamente bajo los términos de esta licencia, o de una versión posterior de esta licencia con sus mismos elementos principales, o de una licencia iCommons de Creative Commons que contenga los mismos elementos principales que esta licencia (ejemplo: Reconocimiento-CompartirIgual 2.5 Japón). Usted debe incluir una copia de la esta licencia o de la mencionada anteriormente, o bien su Identificador Uniforme de Recurso (URI), con cada copia o grabación de la obra que usted reproduzca, distribuya o comunique públicamente. Usted no puede ofrecer o imponer ningún término respecto de las obras derivadas o sus transformaciones que alteren o restrinjan los términos de esta licencia o el ejercicio de sus derechos por parte de los cesionarios de la misma. Usted debe mantener intactos todos los avisos que se refieran a esta licencia y a la

ausencia de garantías. Usted no puede reproducir, distribuir o comunicar públicamente la obra derivada con medidas tecnológicas que controlen el acceso o uso de la obra de una manera contraria a los términos de esta licencia. Lo anterior se aplica a una obra derivada en tanto que incorporada a una obra conjunta o base de datos, pero no implica que éstas, al margen de la obra objeto de esta licencia, tengan que estar sujetas a los términos de esta licencia.

c. Si usted reproduce, distribuye o comunica públicamente la obra o cualquier obra derivada, conjunta o base datos que la incorpore, usted debe mantener intactos todos los avisos sobre la propiedad intelectual de la obra y reconocer al autor original, de manera razonable conforme al medio o a los medios que usted esté utilizando, indicando el nombre (o el seudónimo, en su caso) del autor original si es facilitado, y/o reconocer a aquellas partes (por ejemplo: institución, publicación, revista) que el autor original y/o el licenciador designen para ser reconocidos en el aviso legal, las condiciones de uso, o de cualquier otra manera razonable; el título de la obra si es facilitado; de manera razonable, el Identificador Uniforme de Recurso (URI), si existe, que el licenciador especifica para ser vinculado a la obra, a menos que tal URI no se refiera al aviso sobre propiedad intelectual o a la información sobre la licencia de la obra; y en el caso de una obra derivada, un aviso que identifique el uso de la obra en la obra derivada (e.g.,"traducción castellana de la obra de Autor Original", o "guión basado en obra original de Autor Original"). Tal aviso se puede desarrollar de cualquier manera razonable; con tal de que, sin embargo, en el caso de una obra derivada, conjunta o base datos, aparezca como mínimo este aviso allá donde aparezcan los avisos correspondientes a otros autores y de forma comparable a los mismos.

d. En el caso de la inclusión de la obra en alguna base de datos o recopilación, el propietario o el gestor de la base de datos deberá renunciar a cualquier derecho relacionado con esta inclusión y concerniente a los usos de la obra una vez extraída de las bases de datos, ya sea de manera individual o conjuntamente con otros materiales.

5. Exoneración de responsabilidad

A menos que se acuerde mutuamente entre las partes, el licenciador ofrece la obra tal cual (on an "as-is" basis) y no confiere ninguna garantía de cualquier tipo respecto de la obra o de la presencia o ausencia de errores que puedan o no ser descubiertos. algunas jurisdicciones no permiten la exclusión de tales garantías, por lo que tal exclusión puede no ser de aplicación a usted.

6. Limitación de responsabilidad

Salvo que lo disponga expresa e imperativamente la ley aplicable, en ningún caso el licenciador será responsable ante usted por cualquier teoría legal de cualesquiera daños resultantes, generales o especiales (incluido el daño emergente y el lucro cesante), fortuitos o causales, directos o indirectos, producidos en conexión con esta licencia o el uso de la obra, incluso si el licenciador hubiera sido informado de la posibilidad de tales daños.

7. Finalización de la licencia

a. Esta licencia y la cesión de los derechos que contiene terminarán automáticamente en caso de cualquier incumplimiento de los términos de la misma. Las personas o entidades que hayan recibido obras derivadas, conjuntas o bases de datos de usted bajo esta licencia, sin embargo, no verán sus licencias finalizadas, siempre que tales personas o entidades se mantengan en el cumplimiento íntegro de esta licencia. Las secciones 1, 2, 5, 6, 7 y 8 permanecerán vigentes pese a cualquier finalización de esta licencia.

b. Conforme a las condiciones y términos anteriores, la cesión de derechos de esta licencia es perpetua (durante toda la vigencia de los derechos de propiedad intelectual aplicables a la obra). A pesar de lo anterior, el licenciador se reserva el derecho a divulgar o publicar la obra en condiciones distintas a las presentes, o de retirar la obra en cualquier momento. No obstante, ello no supondrá dar por concluida esta licencia (o cualquier otra licencia que haya sido concedida, o sea necesario ser concedida, bajo los términos de esta licencia), que continuará vigente y con efectos completos a no ser que haya finalizado conforme a lo establecido anteriormente.

8. Miscelánea

a. Cada vez que usted explote de alguna forma la obra, o una obra conjunta o una base datos que la incorpore, el licenciador original ofrece a los terceros y sucesivos licenciatarios la cesión de derechos sobre la obra en las mismas condiciones y términos que la licencia concedida a usted.

b. Cada vez que usted explote de alguna forma una obra derivada, el licenciador original ofrece a los terceros y sucesivos licenciatarios la cesión de derechos sobre la obra original en las mismas condiciones y términos que la licencia concedida a usted.

c. Si alguna disposición de esta licencia resulta inválida o inaplicable según la Ley vigente, ello no afectará la validez o aplicabilidad del resto de los términos de esta licencia y, sin ninguna acción adicional por cualquiera las partes de este acuerdo, tal disposición se entenderá reformada en lo estrictamente necesario para hacer que tal disposición sea válida y ejecutiva.

d. No se entenderá que existe renuncia respecto de algún término o disposición de esta licencia, ni que se consiente violación alguna de la misma, a menos que tal renuncia o consentimiento figure por escrito y lleve la firma de la parte que renuncie o consienta.

e. Esta licencia constituye el acuerdo pleno entre las partes con respecto a la obra objeto de la licencia. No caben interpretaciones, acuerdos o términos con respecto a la obra que no se encuentren expresamente especificados en la presente licencia. El licenciador no estará obligado por ninguna disposición complementaria que pueda aparecer en cualquier comunicación de usted. Esta licencia no se puede modificar sin el mutuo acuerdo por escrito entre el licenciador y usted.