RESUMEN DE LAS CLASES DE ANÁLISIS NUMÉRICO I

Índice general

Prólogo

Esto no pretende ser un libro sobre Análisis Numérico ni algo que se le parezca. Simplemente es un resumen, incompleto, de las clases dadas en el ámbito de la Facultad de Ingeniería, durante los años 2005 y 2006, orientadas originalmente a la parte práctica y luego reconvertidas como clases teóricas durante el primer cuatrimestre de 2007.

El objetivo es dar una guía de los temas y enfocar a los alumnos en las cuestiones más importantes del Análisis Numérico que suelen aplicarse en el ámbito de la ingeniería. No intenta ser un manual ni un libro de texto, sino simplemente servir como ayuda-memoria a la hora de repasar lo visto en clase. Textos y libros existen por doquier y algunos de ellos se refieren en la bibliografía, a los cuales no busca reemplazar. Es más, muchas de las demostraciones se deben buscar en esos textos.

Al mismo tiempo, algunos temas que no suelen incluirse en los libros tradicionales se han desarrollado con mayor o menor fortuna. Dos ejemplos de ello son el *Método de Interpolación Baricéntrica de Lagrange*, una forma alternativa de construir los polinomios interpolantes de Lagrange, y una aproximación al *Método de los Gradientes Conjugados* para resolver sistemas de ecuaciones lineales en forma iterativa. El primero de ellos no figura en ningún libro conocido y es una interesante alternativa para desarrollar en una computadora, tal como refieren quienes publicaron el método, los matemáticos Jean-Paul Berrut (Départment de Mathématiques, Université de Fribourg) y Lloyd N. Trefethen (Computing Laboratory, Oxford University).

El segundo, en realidad, aparece en varios libros dedicados al *método de los elementos finitos*, pero no siempre en los textos de Análisis Numérico. Por ejemplo, recién en la séptima edición del libro *Análisis Numérico* de Burden & Faires se lo incluye como parte importante del libro. En otros textos ni siquiera se menciona, a pesar de ser uno de los métodos iterativos más importante para resolver sistemas ralos de ecuaciones lineales, sobre todo en los últimos años, gracias al desarrollo de las computadoras personales. Para las clases, tanto teóricas como prácticas, la base que se usó es la publicada por Jonathan Richard Shewchuk (School of Computer Science, Carnegie Mellon University), que es de libre disponibilidad en la web, y muy buena desde el punto de vista de la interpretación y comprensión del método.

El resto de los temas corresponde a los que tradicionalmente se dan en los cursos de Análisis Numérico I de la facultad y en la mayoría de los cursos afines en Argentina y el resto del mundo.

Finalmente, este resumen no sería posible sin la ayuda de todos los docentes que han intervenido e intervienen en el curso 008 de Análisis Numérico I de la Facultad de Ingeniería de la Universidad de Buenos Aires. Sus observaciones y críticas a las clases prácticas y teóricas han servido para delinear los temas y tratar de lograr la mejor manera de explicarlos. También de los alumnos de los dos últimos cuatrimestres, quienes han aportado mucho revisandolo y encontrando errores y puntos que no resultaron muy claros o fáciles de seguir.

Rodolfo A. Schwarz, Buenos Aires, agosto de 2008.

Capítulo 1

Errores en los métodos numéricos

1.1. Una definición de Análisis Numérico

Es usual que el análisis numérico esté asociado estrictamente a la siguiente definición general: *Es el estudio de los errores de redondeo.* De acuerdo con Lloyd Trefethen (véase [12]), profesor en la universidad de Oxford, esta definición es errónea. Entiende que si esta percepción es correcta, resulta poco sorprendente, entonces, que el análisis numérico sea visto como una asignatura aburrida y tediosa. Es cierto que los errores de redondeo son inevitables, y que su análisis es complejo y tedioso, pero *no son fundamentales.* Al analizar varios libros dedicados al tema, encuentra que los capítulos iniciales siempre están referidos al error de redondeo o sus temas asociados: precisión, exactitud, aritmética finita, etc. Veamos algunos ejemplos de la bibliografía disponible en español:

- *Burden & Faires, Métodos Numéricos (2005)*: 1. Preliminares matemáticos y análisis del error.
- *González, Análisis Numérico, primer curso (2002)*: 1. Errores en el cálculo numérico.
- *Curtis & Wheatley, Análisis numérico con aplicaciones (2002)*: 0. Cálculo numérico y computadoras (0.5 Aritmética por computadoras y errores).
- *Nakamura, Métodos numéricos aplicados con software (1992)*: 1. Causas principales de errores en los métodos numéricos.
- *Ramírez González y otros, Cálculo Numérico con Mathematica (2005)*: 1. Introducción al Cálculo Numérico. Errores.
- *Maron & Lopéz, Análisis Numérico, un enfoque práctico (1998)*: 1. Algoritmos, errores y dispositivos digitales.
- *Quintana y otros, Métodos numéricos con aplicaciones en Excel (2005)*: Capítulo 1. Definición de error.

Esto ayuda a que los alumnos tengan una percepción equivocada del objeto principal de la materia. Para evitar esto, Trefethen propone una definición alternativa:

> *Análisis numérico es el estudio de los algoritmos para resolver problemas de la matemática continua.*

Para él la palabra clave es *algoritmo*. De hecho, en Wikipedia podemos encontrar esta definición:

> *El análisis numérico es la rama de la matemática que se encarga de diseñar algoritmos para, a través de números y reglas matemáticas simples, simular procesos matemáticos más complejos aplicados a procesos del mundo real*;

cuya referencia es justamente, ¡Lloyd (Nick) Trefethen! Y, según él, el principal objetivo del análisis numérico es diseñar algoritmos para aproximar valores desconocidos (no lo conocido de antemano), y hacerlo en forma rápida, muy rápida.

Por esa razón, este capítulo tiene por objeto desmitificar la influencia de los errores al aplicar métodos numéricos, y en particular, la influencia del error de redondeo como fuente básica de los problemas en la utilización de algoritmos para resolver problemas matemáticos, aún cuando la existencia de los mismos debe llevar a tenerlos en cuenta en determinados casos en los que no se los puede soslayar. Para ello, empezaremos viendo los errores que intervienen en cualquier procedimiento o cálculo numérico.(Para un análisis más detallado acerca del estudio de los errores y la estabilidad de los algoritmos, véase [5].)

1.2. El concepto y las fuentes de error

1.2.1. Introducción

Tal como dijimos en la definición de análisis numérico, su objetivo principal no es analizar en detalle los errores que intervienen en el cómputo de cantidades. Pero sí es uno de los puntos en los cuales cualquier matemático (o de otra rama de la ciencia o tecnología asociada a la matemática) que se dedique a desarrollar algoritmos deberá ser un especialista en el tema. ¿Por qué? Simplemente, porque sus algoritmos serán utilizados para resolver problemas que seguramente no tengan una solución analítica o que la obtención de esa solución está fuera de los alcances del usuario de ese algoritmo. Por ejemplo, es usual que los ingenieros utilicen programas que resuelven estructuras por el *método de los elementos finitos* para dimensionar determinadas piezas o establecer las formas definitivas de las mismas, afín de optimizar el uso de los materiales o para darle ciertas características especiales a la estructura. Si bien es posible que varios de esos problemas puedan ser resueltos con modelos analíticos, lo más probable es que esos modelos analíticos sólo tengan una definición general (aún cuando sea compleja) en forma de ecuaciones diferenciales o de sistemas de ecuaciones diferenciales, tanto ordinarias como en derivadas parciales. Aún cuando existen métodos de resolución analíticos (simbólicos) para las ecuaciones diferenciales, las condiciones de borde de un problema particular puede hacer inútil la búsqueda de soluciones analíticas o simbólicas. Por lo tanto, el único camino viable para obtener una respuesta al problema planteado es la aplicación de un método numérico.

¿Y si no contamos con una solución analítica, cómo sabremos si los resultados obtenidos sirven? Esta es una de las razones por las cuales los analistas numéricos deben ocuparse de analizar qué tipos de errores afectan a los algortimos que desarrollan y hasta qué punto son responsables de los posibles errores en los resultados que se obtendrán por aplicaciones de los mismos. Pero debe tenerse en cuenta que, por otro lado, estos algoritmos deben ser rápidos (de convergencia rápida) y que serán aplicados en computadoras, algo que no suele remarcarse con debida propiedad, que, por supuesto, están sometidas a limitaciones propias.

El problema de los errores en los cálculos no es propiedad del siglo XX (o XXI) y de los que sigan. Desde los inicios de la matemática y de las ciencias asociadas, es un problema que interesó e interesa a todos los involucrados. Como ejemplo, tomemos un típico método de interpolación que se enseña en cualquier curso, el de los *polinomios de Lagrange*. La fórmula para

obtener los polinomios es:

$$P_n(x) = \sum_{j=0}^{n} f_j \; l_j(x); \quad l_j(x) = \frac{\prod_{\substack{k=0 \\ k\neq j}}^{n} (x - x_k)}{\prod_{\substack{k=0 \\ k\neq j}}^{n} (x_j - x_k)} \tag{1.1}$$

El propio Lagrange advertía en su época, que su método no era totamente confiable, pues muchas veces los resultados obtenidos no eran correctos. De todos modos, el método suele estudiarse como una herramienta teórica a pesar de que tiene las siguientes desventajas:

1. Cada evaluación de $p(x)$ requiere $O(n^2)$ sumas/restas y multiplicaciones/divisiones.
2. Añadir un nuevo par de datos $x_{n+1}; f_{n+1}$ requiere recalcular todo de cero.
3. *El cálculo es numéricamente inestable.*

En tanto que las dos primeras se refieren a la eficiencia del algoritmo para obtener el polinomio, la última está estrictamente relacionda con los errores que pueden aparecer por las operaciones de cálculo involucradas en el procedimiento. Esto último se hizo muy evidente al utilizar las computadoras como elemento de cálculo. En consecuencia, para analizar cuan inestable (y/o mal condicionado) es el algoritmo, debemos analizar cómo se propagan los errores. Veremos a continuación el concepto y la definición de lo que denominamos *error*.

1.2.2. Concepto de error

La palabra *error* suele llevar a interpretaciones confusas según quien la exprese. En el lenguaje coloquial de uso diario, el concepto de error está relacionado con falla o mal hecho. Una expresión como «el error fue ...» suele asociarse con la causa que produjo un resultado no aceptable o equivocado, y que por lo tanto, debe ser evitado o enmendado para que al hacer de nuevo el cálculo (o cualquier otra cosa), el resultado obtenido sea aceptable o correcto.

En cambio, en el ámbito del análisis numérico (y en general, en las ciencias e ingeniería), el término error está relacionado específicamente con la incertidumbre de los datos de ingreso como de los resultados obtenidos, *sin que esto signifique necesariamente que los resultados sean equivocados*. Dicho de otra manera, no pone en duda la confiabilidad del método en sí, sino que analiza el grado de incertidumbre de los valores numéricos. En la ingeniería esto es de particular relevancia, puesto que los datos que utilizamos provienen de mediciones en campo, estimaciones probabilísticas, hipótesis y modelos matemáticos simplificados, o de la experiencia profesional. Rara vez se cuenta con datos con validez «exacta». Sin embargo, si una leve modificación de estos datos produce resultados considerablemente diferentes que no reflejan la realidad, estamos ante la presencia de un problema que sí puede objetar el procedimiento utilizado. Es decir, el procedimiento es inestable o mal condicionado, conceptos diferentes.

Para analizar cuan confiable es un procedimiento o algoritmo, se vuelve necesario el estudio de los errores que afectan los cálculos y las operaciones que intervienen en dicho algoritmo, y cómo se propagan hasta afectar los resultados que éste entrega.

1.2.3. Fuentes de error

Las fuentes de error que analizaremos son las siguientes:

- **Error inherente**: Es el error de los datos de entrada que puede estar dado por la precisión en la medición de los datos, por la representación numérica, por provenir de cálculos previos, etc.

- **Error de redondeo/corte**: Es el error debido estrictamente a la representación numérica utilizada y está asociado a la precisión usada en los cálculos, generalmente una calculadora o una computadora.
- **Error de truncamiento/discretización**: Es el error que aparece al transformar un procedimiento infinito en uno finito, por ejemplo, transformar una serie de infinitos términos en una función finita, o de usar una aproximación discreta para representar un fenómeno continuo.
- **Error del modelo matemático**: Es el debido a las simplificaciones e hipótesis introducidas para definir el modelo matemático que representa el problema físico.
- **Error humano y/o de la máquina**: Es el error que se produce por la intervención humana, ya sea por una mala transcripción o interpretación incorrecta de los datos originales, por programas de computación mal hechos y/o fallas en el diseño, implementación o configuración de programas o computadoras.

La última fuente de error suele ser asociada al concepto coloquial de «error». Desde la óptica del análisis numérico, los dos últimos errores están fuera de su alcance, si bien no deben ser despreciados a la hora de evaluar los resultados obtenidos, en particular, el debido al modelo matemático.

1.3. Error absoluto y error relativo

Empezaremos por analizar las fórmulas más sencillas de error. Supongamos que obtenemos de alguna forma (por ejemplo, una medición) cierto valor $\bar{m}$. Sabemos que el valor «exacto» de dicho valor es m. Como conocemos ese valor m podemos definir dos tipos de errores:

1. **Error absoluto**: $e_a = m - \bar{m}$;
2. **Error relativo**: $e_r = \dfrac{m - \bar{m}}{m} = \dfrac{e_a}{m}$ (siempre que $m \neq 0$).

Generalmente, el error relativo es una medida mucho más representativa del error, especialmente cuando $|m| >> 1$. Cuando $|m| \approx 1$, entonces ambos errores coinciden. En la práctica suele ser poco probable conocer el valor m, por lo que no podemos calcular e_a ni e_r. Entonces, ¿cómo sabemos qué error estamos teniendo? Si no conocemos la solución del problema pareciera que no hay forma de saberlo.

Partamos de no conocer m y de que el valor $\bar{m}$ fue obtenido por medición usando un instrumento cuya precisión[1] es e_m (por error de medición). Si tomamos el concepto de error absoluto podemos obtener una idea del valor de m. En efecto, tenemos que:

$$e_m = e_a = m - \bar{m} \;\Rightarrow\; m = \bar{m} + e_a;$$

que podemos generalizar a:

$$m = \bar{m} \pm e_a;$$

si tenemos en cuenta que el valor de e_a puede ser positivo o negativo. Así, una forma más general de escribir el error absoluto y el relativo es:

1. $|E| = |e_a| = |m - \bar{m}|$;
2. $|e_r| = \left|\dfrac{m - \bar{m}}{m}\right| = \dfrac{|m - \bar{m}|}{|m|} = \dfrac{|E|}{|m|}$.

[1] En este caso, precisión se refiere a la unidad más chica que el instrumento puede medir.

Como hemos supuesto que $e_a = e_m$, sabemos cual es nuestro error absoluto, pero seguimos sin saber cuál es nuestro error relativo. Tenemos dos posibilidades para obtener m:

$$m = \bar{m} + e_a \quad \text{o} \quad m = \bar{m} - e_a,$$

y entonces el error relativo sería:

$$e_r = \frac{e_a}{\bar{m} + e_a} \quad \text{o} \quad e_r = \frac{-e_a}{\bar{m} - e_a} \Rightarrow |e_r| = \frac{|e_a|}{|\bar{m} + e_a|} \quad \text{o} \quad |e_r| = \frac{|-e_a|}{|\bar{m} - e_a|}.$$

Resulta, entonces, más conveniente definirlo como:

$$e_r = \frac{|e_a|}{|\bar{m}|},$$

cuando se conoce $\bar{m}$ y e_a.

1.4. Propiedades de los algoritmos

Hemos dicho que el análisis numérico se ocupa de estudiar algoritmos para resolver problemas de la matemática continua. Dado que estos algoritmos son una aproximación al problema matemático, resulta evidente que los resultados obtenidos estarán afectados por alguno de los errores mencionados. Y como en muchas ocasiones los datos de entrada de ese algoritmo también tienen errores, la pregunta que surge inmediatamente es: ¿cómo sabemos si los resultados que arroja el algoritmo son confiables? La pregunta no tiene una única respuesta, depende del tipo de error que analicemos o que tenga mayor influencia y de las características del problema matemático. Podemos tener varias aproximaciones acerca de un algoritmo, a saber:

1. Una primera aproximación a una respuesta sería analizar cuan sensible son los resultados que arroja un algoritmo cuando los datos de entrada se modifican levemente, o sea, cuando sufren una perturbación. Un análisis de este tipo tiene dos formas ser encarado, por un lado, estudiando la *propagación de errores* (en inglés, *forward error*), es decir, perturbar los datos de entrada y ver qué consecuencia tiene en el resultado. Pero también se puede estudiar de manera inversa, partir de una perturbación en los resultados, y analizar qué grado de perturbación pueden sufrir los datos de entrada, metodología que se conoce como *análisis retrospectivo* (en inglés, *backward error*). En ambos casos estamos estudiando la influencia del *error inherente*.

2. Una segunda aproximación puede ser analizar el algoritmo con diferentes representaciones numéricas en los datos de entrada y estudiar qué ocurre con los resultados. En este caso estudiamos la incidencia del *error de redondeo*.

3. Finalmente, y tal vez el más sencillo de todos, otra aproximación puede ser analizar qué ocurre cuando se trunca un procedimiento o discretiza el dominio de nuestro problema matemático. Este tipo de análisis puede que requiera solamente de un trabajo algebraico más que numérico, y, a veces, suele combinarse con el error de redondeo.

La enumeración anterior en tres aproximaciones es a los efectos de identificar las causas y la forma de encarar el problema. Sin embargo, la realidad suele ser mucho más compleja, y los errores que surgen de aplicar un algoritmo o varios, resultan ser una combinación de todos y dependen, muchas veces, de las características de los datos del problema.

1.4.1. Condición de un problema

El primer caso, el análisis de la propagación de los errores inherentes, permite establecer si el problema está *bien o mal condicionado*. Si al analizar un pequeño cambio (o perturbación) en los datos el resultado se modifica levemente (o tiene un pequeño cambio) enonces estamos ante un problema **bien condicionado**. Si, por el contrario, el resultado se modifica notablemente o se vuelve oscilante, entonces el problema está **mal condicionado**. Si éste fuera el caso, no hay forma de corregirlo cambiando el algoritmo (como se verá después) pues el problema está en el modelo matemático.

Definición 1.1. Un problema matemático (numérico) se dice que está *bien condicionado* si pequeñas variaciones en los datos de entrada se traducen en pequeñas variaciones de los resultados.

Observación 1.1.1. Un problema mal condicionado puede ser resuelto con exactitud, si realmente es posible, solamente si se es muy cuidadoso en los cálculos.

Observación 1.1.2. Si f representa al algoritmo «real» y f^* al algoritmo «computacional», y x a la variable «real» y x^* a la variable «computacional», entonces el error en los resultados se puede definir como:

$$|f(x) - f^*(x^*)| \leq \underbrace{|f(x) - f(x^*)|}_{\text{condición}} + \underbrace{|f(x^*) - f^*(x)|}_{\text{estabilidad}} + \underbrace{|f^*(x) - f^*(x^*)|}_{\text{truncamiento}}.$$

Veremos más adelante que las «pequeñas variaciones» en los datos de entrada están asociadas al problema en cuestión. No es posible «a priori» definir cuantitativamente cuando una variación es «pequeña» y cuando no lo es. El análisis de los errores inherentes es importante para establecer la «sensibilidad» del modelo numérico a los cambios en los datos, puesto que rara vez los datos de entrada están exentos de error.

1.4.2. Estabilidad de un algoritmo

El segundo caso es el que suele ser un «dolor de cabeza» para los analistas numéricos. Si analizamos un algoritmo ingresando los datos con diferentes representaciones numéricas, esto es, con diferente precisión, y los resultados no cambian demasiado (salvo por pequeñas diferencias en los decimales), entonces estamos en presencia de un algoritmo *estable*. Caso contrario, el algoritmo es **inestable**.

El último caso está asociado a procedimientos o algoritmos basados en series o iteraciones «infinitas», y suelo combinarse con alguno de los otros errores, como veremos más adelante.

En consecuencia, lo que debemos buscar de un algoritmo es que sea ***estable***. ¿Qué significa esto en la práctica? Supongamos (una vez más, supongamos) que E_n mide un cierto error cometido en el paso n de un algoritmo. Podemos expresar este error en función del error inicial, que puede tener una de estas dos expresiones:

1. Error con crecimiento lineal: $E_n \approx c \cdot n \cdot E_0$
2. Error con crecimiento exponencial: $E_n \approx c^n \cdot E_0$

Es evidente que el primer error es «controlable», en tanto que el segundo, no. Puesto que es imposible que no haya errores al trabajar con un algoritmo, lo que se debe buscar es que el error siga una ley lineal (como en el primer caso) y no una ley exponencial. A partir de esta comprobación se desprende la siguiente definición:

Definición 1.2. Un algoritmo se considera *estable* cuando la propagación de los errores de redondeo es lineal o cuasi-lineal.

En cambio, un algoritmo que propaga los errores en forma exponencial es **inestable**.

Una de las razones principales de analizar la propagación de los errores de redondeo es conseguir que un algoritmo sea estable. Sin embargo, debemos tener bien presente que un algoritmo estable en ciertas condiciones puede volverse inestable en otras, por lo que muchas veces no existe el algoritmo «universal». Dado que la estabilidad (o la inestabilidad) es una propiedad exclusiva del algoritmo, si un problema se vuelve inestable podemos, muchas veces, corregirlo cambiando el algoritmo inestable por otro estable. (Sin embargo, nunca hay que olvidar que un problema puede volverse mal condicionado para determinadas condiciones de base, lo que hace más complejo el análisis.)

Veamos un ejemplo que muestra la inestabilidad de un algoritmo. Tomemos la siguiente integral definida:

$$y_n = \int_0^1 \frac{x^n}{x+10}\mathrm{d}x;$$

con $n = 1;\ 2;\ \ldots\ ;\ 34$.

Es fácil ver que las primeras integrales analíticas son relativamente sencillas de obtener (por ejemplo, para $n = 1$ o $n = 2$). En efecto, si queremos hallar y_1 podemos hacer:

$$\begin{aligned} y_1 &= \int_0^1 \frac{x}{x+10}\mathrm{d}x = x|_0^1 - 10\ \ln(x+10)|_0^1 = 1 - 10\ \ln\left(\frac{11}{10}\right); \\ y_1 &= 1 - 10\ \ln(1,1) = 0,0468982019570. \end{aligned}$$

Pero si queremos obtener y_{15} la situación ya no es tan sencilla. Deberíamos calcular la siguiente integral:

$$y_{15} = \int_0^1 \frac{x^{15}}{x+10}\mathrm{d}x.$$

Para facilitar el cálculo de cada una de las y_n integrales, desarrollemos un algoritmo que nos permita obtener los valores de las mismas sin tener que integrar o que al menos utilice aquellas integrales «fáciles». Para un $n > 1$ cualquiera podemos decir que:

$$\begin{aligned} y_n + 10\,y_{n-1} &= \int_0^1 \frac{x^n + 10\,x^{n-1}}{x+10}\mathrm{d}x = \int_0^1 \frac{x+10}{x+10}\,x^{n-1}\mathrm{d}x = \int_0^1 x^{n-1}dx \Rightarrow \\ y_n + 10\,y_{n-1} &= \frac{1}{n} \ \Rightarrow\ y_n = \frac{1}{n} - 10\,y_{n-1} \end{aligned}$$

Si queremos calcular y_1 necesitamos obtener y_0, que también resulta muy sencillo de obtener, pues:

$$\begin{aligned} y_0 &= \int_0^1 \frac{1}{x+10}\mathrm{d}x = \ln(x+10)|_0^1 = \ln(11) - \ln(10) \\ y_0 &= \ln(1,1) = 0,0953101798043. \end{aligned}$$

Para analizar si el algoritmo arroja resultados confiables, empezaremos por calcular algunos valores. Hemos calculado el valor de y_1 en forma analítica, por lo tanto, tenemos un valor de comprobación. Por otro lado, por las características del problema sabemos que $0 \leq y_n \leq 1$. Si definimos las funciones $f_i(x) = \dfrac{x^i}{x+10}$ y las graficamos, podemos ver que el área bajo esas funciones es menor a $\frac{0,1}{2} = 0,05$. En la figura 1.1 se pueden ver representadas algunas de estas curvas.

Para comprobar la eficacia del algoritmo hemos utilizado dos programas muy conocidos: el MathCAD® y el MS Excel®. Con el primero hemos calculado las y_i en forma analítica y con el algoritmo dado; con el segundo, sólo con el algoritmo. En la tabla 1.1 se tienen los resultados obtenidos.

Tabla 1.1: Cálculo de los y_i

i	Analítico	MathCAD®	MS Excel®
1	0,0468982019567514000	0,04689820195675065	0,04689820195675
2	0,0310179804324860060	0,03101798043248486	0,03101798043248
3	0,0231535290084732900	0,02315352900839845	0,02315352900857
4	0,0184647099152671080	0,01846470991601545	0,01846470991435
5	0,0153529008473289370	0,01535290083984547	0,01535290085650
6	0,0131376581933772860	0,01313765826821192	0,01313765810168
7	0,0114805609233700040	0,01148056017502363	0,01148056184036
8	0,0101943907662999780	0,01019439824976364	0,01019438159642
9	0,0091672034481113700	0,00916712861347462	0,00916729514693
10	0,0083279655188863120	0,00832871386525371	0,00832704853072
11	0,0076294357202277880	0,00762195225655373	0,00763860560192
12	0,0070389761310554600	0,0071138107677959	0,00694727731410
13	0,0065333156125223285	0,00578469245117427	0,00745030378206
14	0,0060954153033481685	0,01357887897739715	-0,00307446639198
15	0,0057125136331849920	-0,06912212310730485	0,09741133058647
16	0,0053748636681500880	0,7537212310730486	-0,91161330586469
17	0,0050748927302638440	-7,478388781318721	9,174956588
18	0,0048066282529171250	74,83944336874276	-91,69401033
19	0,0045652964181971910	-748,3418021084802	916,9927348
20	0,0043470358180281100	7483,468021084803	-9169,877348
21	0,0041486894387665300	-74834,63259180041	91698,8211
22	0,0039676510668801740	748346,3713725496	-916988,1656
23	0,0038017502007634874	-7483463,670247235	9169881,699
24	0,0036491646590318034	74834636,74413903	-91698816,95
25	0,0035083534096819780	-748346367,4013903	916988169,5
26	0,0033780043647186900	7483463674,052364	-9169881695
27	0,0032569933898501480	-74834636740,48662	91698816953
28	0,0031443518157842460	748346367404,902	$-9,16988 \times 10^{11}$
29	0,0030392404628472014	-7483463674048,985	$9,16988 \times 10^{12}$
30	0,0029409287048613280	74834636740489,89	$-9,16988 \times 10^{13}$
31	0,0028487774675157640	-748346367404898,9	$9,16988 \times 10^{14}$
32	0,0027622253248423658	7483463674048989	$-9,16988 \times 10^{15}$
33	0,0026807770546066550	-74834636740489890	$9,16988 \times 10^{16}$
34	0,0026039941598158087	748346367404898800	$-9,16988 \times 10^{17}$

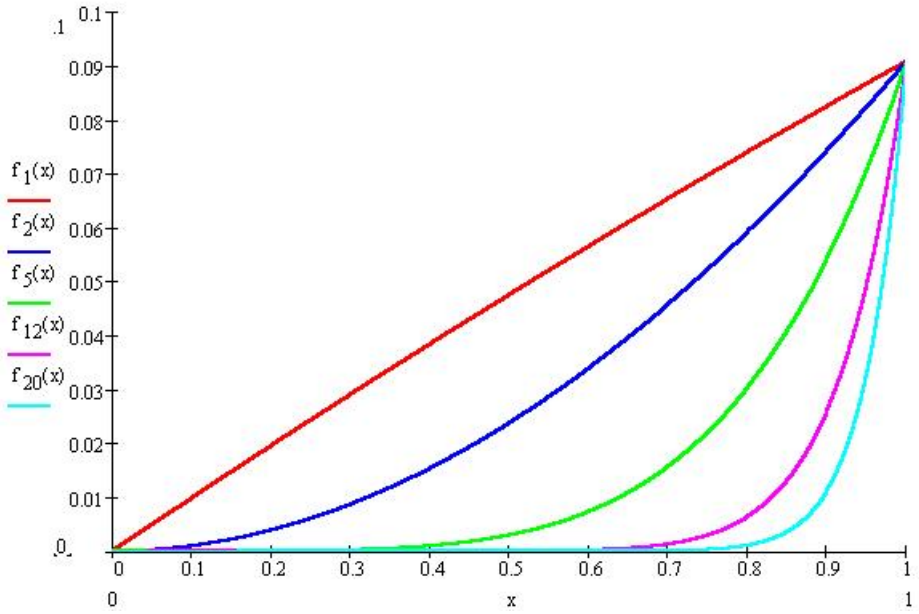


Figura 1.1: Curvas de las distintas funciones.

Podemos ver que los primeros valores obtenidos con el algoritmo, tanto en MathCAD® como en MS Excel®, resultan una buena aproximación de los valores de y_n. Los problemas aparecen para y_{12} y siguientes. Detengámonos a analizar los resultados obtenidos a partir de y_{12}.

En el caso particular de este último, el error empieza a ser considerable, comparando con los resultados anteriores. En los siguientes valores se obtienen algunos resultados realmente curiosos. Ya hemos visto que los valores de y_n, o sea, las áreas bajo las curvas, están limitados superiormente por $0,05$. Además podemos ver que $y_n > y_{n+1}$, es decir, que las áreas bajo la curva van disminuyendo a medida que crece n. Si miramos los obtenidos con MS Excel®, el y_{13} es mayor que el y_{12}, algo que no es posible según lo visto antes. Y el y_{14} es, ¡negativo! El área bajo la curva no puede ser negativa. Con los resultados obtenidos con el MathCAD® ocurre algo similar. Para este programa, y_{14} da mayor que y_{13}, y el y_{15} da negativo, ambos resultados incorrectos.

A partir de estos valores, los resultados se vuelven oscilantes (cambian de signo), y mayores que uno ($y_n > 1$ para $n = 17; 18; \ldots; 34$), algo que por el tipo de curva no es posible, como ya vimos. En consecuencia, resulta evidente que el algoritmo tiene algún problema para calcular los valores de y_n cuando $n \geq 12$, por lo que no nos sirve para obtener el y_{34}. Aún cuando no tuviéramos el resultado exacto, mirando la curva nos daríamos cuenta que hay un diferencia muy grande entre el valor «real» y el obtenido con el algoritmo. Más aún, el error que estamos teniendo no sigue una ley lineal sino una ley exponencial (se va multiplicando por 10), lo que dice claramente que el algoritmo analizado es *inestable*.

Este ejemplo nos muestra cómo un algoritmo mal diseñado nos puede entregar resultados que inicialmente son bastante aproximados pero que en pasos posteriores son incorrectos, y por lo tanto, inútiles.

Definición 1.3. Un algoritmo debe ser diseñado procurando que sea bien condicionado y estable.

Observación 1.3.1. Un algoritmo inestable a la larga da resultados incorrectos, por más que esté bien condicionado.

Es por eso que debemos desarrollar algún tipo de análisis que nos permita detectar si un algoritmo está bien condicionado o no y si es estable o no. Para ello, empezaremos por analizar algunos tipos de error.

1.5. Errores

1.5.1. Error inherente

Éste suele ser el error más fácil de entender. Es el que está relacionado directamente con los datos de entrada o de base. Dado que estos datos suelen provenir de mediciones, cálculos anteriores, proyecciones estadísticas, etc., el valor numérico de los datos no es «exacto» sino que está asociado a un intervalo de validez. Cuando se mide una longitud con una cinta métrica con divisiones hasta el centímetro, el error por la apreciación del instrumento es un centímetro o medio centímetro (5 mm). Es decir, si mide 145,01 m, en realidad, se está diciendo que el valor es $145,01 \pm 0,01$ o $145,010 \pm 0,005$. Lo mismo ocurre si los datos se obtienen por un cálculo anterior o una estimación estadística. En esos casos, el error se obtiene por otros métodos.

Veamos un ejemplo. Supongamos que tenemos las siguientes cantidades, $a = 3,0 \pm 0,1$ y $b = 5,0 \pm 0,1$ y queremos hallar $z = a + b$. Lo que deberemos hacer es:

$$z = (3,0 \pm 0,1) + (5,0 \pm 0,1)$$

Al efectuar esta operación obtendremos cinco resultados posibles: 7,8; 7,9; 8,0; 8,1 y 8,2. Es decir, z está en el intervalo [7,8; 8,2], o, lo que es lo mismo, $z = 8,0 \pm 0,2$. Así cualquier resultado obtenido dentro del intervalo dado se puede considerar «correcto».

Esto muestra la sencillez del análisis cuando las operaciones son pocas (en esta caso, una). Sin embargo, si el algoritmo es más complejo, hacer las n combinaciones posibles de operaciones con los datos de ingreso puede ser imposible y nada práctico. De ahí que el análisis de la propagación de los errores inherentes es la forma más conveniente para establecer la incidencia de los mismos en los resultados finales. Más adelante vermos la diversas formas de analizar esta propagación.

1.5.2. Error de redondeo

Antes de analizar el error de redondeo, veremos la manera de representar un número según la forma adoptada. A partir de esta representación se entenderá cual es la incidencia del error en los cálculo efectuados con ayuda de una computadora.

Representación numérica

Para empezar, supongamos el siguiente número: $\frac{4}{3}$. En el sistema decimal suele representarse como $1,3333\ldots$. Una forma alternativa es:

$$\frac{4}{3} \cong \left(\frac{1}{10} + \frac{3}{10^2} + \frac{3}{10^3} + \frac{3}{10^4} + \frac{3}{10^5} + \ldots\right) \times 10^1 = 1,3333\ldots ;$$

o sea, un número que sólo puede representarse con una serie de infinitos términos, algo imposible desde el punto de vista práctico. Su única forma de expresión «exacta» es simbólica. Una calculadora, por ejemplo, sólo puede representarlo en forma numérica (en base diez, como la escrita arriba) y, por ende, la única representación posible es finita[2]. En consecuencia, debe truncarse esta serie en «n» términos. Por ejemplo, una representación posible es:

$$\frac{4}{3} \cong \left(\frac{1}{10} + \frac{3}{10^2} + \frac{3}{10^3} + \frac{3}{10^4}\right) \times 10^1 = 0,1333 \times 10^1 = 1,333.$$

Podemos ver que esta representación está formada por un coeficiente (0,1333), una base (10) y un exponente (1). Esta forma de representación se conoce como **representación de coma (punto) flotante**. Una generalización de esta representación se puede escribir como:

$$fl(x) = \pm 0, d_1 d_2 d_3 \ldots d_{t-1} d_t \times 10^e = \pm \left(\frac{d_1}{10} + \frac{d_1}{10^2} + \frac{d_1}{10^3} + \ldots + \frac{d_1}{10^{t-1}} + \frac{d_1}{10^t}\right) \times 10^e.$$

[2]Distinto sería el caso si se usara base 3. Entonces $\frac{4}{3}$ sería igual a 1,1; una representación «exacta».

La forma normalizada es que d_1 sea distinto de cero ($1 \leq d_1 \leq 9$) y que los restantes d_i estén comprendidos en el siguiente intervalo: $0 \leq d_1 \leq 9$, para $i = 2;\ 3;\ 4; \ldots;\ t$. También se limita el exponente e, con dos valores, $I < 0$ y $S > 0$, por lo que se cumple que $I \leq e \leq S$. Así, podemos hallar el máximo numéro a representar, que es $0,99\ldots99 \times 10^S \approx 10^S$, y el más chico, $0,10\ldots00 \times 10^I = 10^{I-1}$.

Una vez definida la forma de representar los números, pasemos a definir nuestra *precisión*, que significa cuantos términos d_i usaremos, esto es, el t que vimos, y el exponente e de la base.

Para complicar más las cosas, las calculadoras y fundamentalmente, las computadoras, usan una representación numérica con base 2 [3]. Esto trae ventajas y desventajas. Por ejemplo, puesto que se usa base 2, los d_i sólo pueden valer 0 o 1, con excepción del d_1, que vale siempre 1. Esto facilita la representación de los números y las operaciones. Pero las desventaja es que sólo los números que pueden representarse como sumas de $\frac{1}{2^i}$ resultan exactos. Veamos cómo funciona esto.

Supongamos que tomamos nuestro sistema de representación binario para representar nuestro número inicial, $\frac{4}{3}$. Tomemos que la cantidad de términos, t, sea 8 y dejemos por un momento libre de restricciones el exponente e. Entonces, tendremos lo suguiente:

$$\frac{4}{3} \rightarrow 0,10101010 \times 2 = \left(\frac{1}{2} + \frac{1}{8} + \frac{1}{32} + \frac{1}{128}\right) \times 2 = 0,6640625 \times 2 = 1,328125;$$

número parecido al buscado pero no igual. Esto nos muestra que existe una limitación cuando utilizamos una computadora (o una calculadora) para representar números que no tienen una representación directa en base binaria. Asociada a esta limitación, la de poder representar sólo una cantidad finita de números, surge el error por corte o redondeo.

No siempre se entiende la incidencia del error por la representación numérica. Un ejemplo que ya es tradicional de lo catastrófico que puede ser tomar una representación numérica sin analizar su incidencia, es la falla de la batería de misilies Patriot en Dharan, Arabia Saudita, durante la Guerra del Golfo en 1991, en detectar un misil SCUD iraquí, que resultó 28 soldados muertos y casi 100 heridos.

El problema estaba en el programa de rastreo del sistema de detección de blancos enemigos. El sistema disponía de contador de tiempo, en números enteros, que registraba las ventanas de rastreo de una décima de segundo $(0,1)$ para detectar los blancos, contador que luego se multiplicaba por $\frac{1}{10}$ para transformarlo en tiempo real. El programa trabajaba con una representación numérica de $\frac{1}{10}$ en base 2, cuya representación es $0,0001100110011001100110011001100\ldots$, que es periódica, o sea, infinita. Como no se puede trabajar con una representación infinita, el programa adoptó una «precisión» de 24 bits. Por ese motivo la representación de $0,1$ se redujo a $0,00011001100110011001100$, con el consiguiente error de corte/redondeo, dado por el número binario $0,00000000000000000000000011001100110011001100\ldots$, que en representación decimal es aproximadamente $0,000000095$ $(9,5 \cdot 10^{-8})$. Esta diferencia, que parece pequeña, luego de 100 horas de operación continua se convirtió en algo peligroso: al multipplicar 100 horas por ese error en la representación numérica del programa $(0,000000095 \cdot 100 \cdot 60 \cdot 60 \cdot 10)$, dío una diferencia de $0,34$ segundos respecto del tiempo «real». La consecuencia de esa diferencia es que la ventana de detección «se corrió» $0,34$ segundos (respecto de $0,10$), por lo que el misil SCUD iraquí (que vuela a 1,600 km/s) no fue detectado por el sistema de rastreo en la siguiente ventana y el sistema de alerta lo consideró una falsa alarma, permitiendo que el misil enemigo impactara en la base de Dharan.

Lo dramático en este caso es que esa falla en el sistema de rastreo se había detectado, estableciémdose que el sistema debía ser reiniciado cada ocho horas de operación continua, porque a las ocho horas la ventana de rastreo se desplazaba un 20 %. Pero la modificación del procedimiento operativo fue enviado un día después del incidente.

[3] Existen, sin embargo, procesadores que no usan una representación binaria.

Error por corte/redondeo

Volvamos a nuestro sistema decimal tradicional. Supongamos ahora que nuestros números se pueden representar de la siguiente manera:

$$fl(x) = \pm\,(0, d_1 d_2 d_3 \ldots d_t d_{t+1} d_{t+2} \ldots) \times 10^e.$$

Si nuestra precisión elegida es t, entonces debemos «recortar» el número definido arriba, pues no podemos representar los d_i para $i > t$. En consecuencia, tenemos dos alternativas básicas para efectuar dicho recorte:

1. **Corte**: Ignorar los dígitos d_i cuando $i > t$.
2. **Redondeo**: Sumar 1 a d_t si $d_{t+1} \geq \frac{10}{2}$ e ignorar los restantes d_i para $i > t+1$, o aplicar corte si $d_{t+1} < \frac{10}{2}$.

Esto nos permite obtener una cota del error absoluto para ambos casos:

$$e_A = \begin{cases} 10^{-t} \times 10^e & \text{para corte} \\ \frac{1}{2}10^{-t} \times 10^e & \text{para redondeo.} \end{cases}$$

Y como definimos el error absoluto, también podemos definir un límite para el error relativo, que será:

1. **Corte**: $e_r \leq \dfrac{10^{-t} \times 10^e}{0,1 \times 10^e} = 10^{1-t}$.
2. **Redondeo**: $e_r \leq \dfrac{1}{2}\dfrac{10^{-t} \times 10^e}{0,1 \times 10^e} = \dfrac{1}{2}10^{1-t}$.

Al valor 10^{1-t} lo identificaremos con la letra μ, y resulta ser importante porque nos da una idea del error relativo que cometemos al utilizar una representación de coma flotante. Suele denominarse como **unidad de máquina** o **unidad de redondeo**. El negativo del exponente de μ suele llamarse también *cantidad de dígitos significativos.*

Dígitos de guarda

Supongamos el siguiente caso. Tomemos el número 0,1425 que debe ser redondeado a tres dígitos significativos. Aplicando el criterio anterior rápidamente obtenemos que el resultado es 0,143 pero, ¿es correcto este redondeo? ¿Por qué no redondear a 0,142; si está a medio camino de ambos? Supongamos que hacemos la operación $2 \times 0,1425$, cuyo resultado es $0,2850$, ¿qué pasa con la misma operación si el número está redondeado? Evidentemente da diferente puesto que la operación es $2 \times 0,143$ cuyo resultado es $0,286$. La diferencia entre ambos es $0,001$ que es justamente la unidad de redondeo. Esto se vuelve aún más importante cuando se tiene la resta de números similares ($a - b$ con $a \approx b$). De ahí que la mayoría de las computadoras actuales (y los programas) trabajen con lo que se conoce como «dígitos de guarda», es decir, más precisión que la mostrada en forma «normal» en pantalla. Pero este ejemplo sirve además para desarrollar otra forma de redondeo.

Redondeo exacto

Tal como dijimos, el número $0,1425$ está mitad de camino de ser redondeado a $0,143$ como a $0,142$. Este problema ha llevado a desarrollar el concepto de «redondeo exacto», que consiste en redondear todos los números que terminan en 5 de manera de que el último dígito significativo sea par. En consecuencia, aplicando este criterio, $0,1425$ se redondea a $0,142$ y no a $0,143$. El criterio va de la mano del «dígito de guarda» y debería ser el redondeo «normal». (Para más detalles respecto a dígitos de guarda y el redondeo exacto, véase [3].)

1.5.3. Error de truncamiento/discretización

Este error surge de aproximar procesos continuos mediante procedimientos discretos o de procesos «infinitos» mediante procedimientos «finitos». Como ejemplo del primer caso suele tomarse la diferenciación numérica como forma de aproximar el cálculo de una derivada en un punto (o su equivalente, la integración numérica), en tanto que para el otro, el ejemplo más es usual es la utilización de métodos iterativos para resolver sistemas de ecuaciones lineales.

En general, este error está asociado al uso de la serie de Taylor para aproximar funciones, de modo que estimar una cota del error no conlleva una dificultad mayor. Sin embargo, en él suelen interactuar el error inherente y/o el de redondeo, con lo que muchas veces su influencia no es bien advertida o es muy reducida. Para ello veamos un ejemplo típico.

Supongamos que queremos calcular una aproximación de $f'(x_0)$ para una función continua, pues no es posible obtener la derivada en forma analítica o resulta muy difícil. Por lo tanto, usaremos un entorno del punto x_0 para calcular $f'(x_0)$ utilizando solamente $f(x)$. Para ello nos valdremos de la serie de Taylor. En efecto, para cualquier punto distante h de x_0 tendremos:

$$f(x_0+h) = f(x_0) + f'(x_0)h + f''(x_0)\frac{h^2}{2} + f'''(x_0)\frac{h^3}{6} + f''''(x_0)\frac{h^4}{24} + \dots \ .$$

Entonces podemos despejar $f'(x_0)$, que resulta ser:

$$f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \left[f''(x_0)\frac{h}{2} + f'''(x_0)\frac{h^2}{6} + f''''(x_0)\frac{h^3}{24} + \dots\right].$$

Si nuestro algoritmo para aproximar $f'(x_0)$ es:

$$\frac{f(x_0+h)-f(x_0)}{h},$$

el error que cometemos en la aproximación está dado por:

$$\left|f'(x_0) - \frac{f(x_0+h)-f(x_0)}{h}\right| = \left|f''(x_0)\frac{h}{2} + f'''(x_0)\frac{h^2}{6} + f''''(x_0)\frac{h^3}{24} + \dots\right|.$$

El término de la derecha es el denominado *error de truncamiento*, pues es lo que se *truncó* a la serie de Taylor para aproximar el valor buscado. Este error suele asociarse también con la convergencia (o la velocidad de convergencia), que suele representarse como $O(n)$ (generalmente, como $O(h^n)$), siendo n el parámetro que determina la velocidad o la convergencia. En nuestro caso, y dado que h generalmente es menor a 1, podemos decir que la aproximación es del tipo:

$$f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} + O(h),$$

que indica que el error que se comete es proporcional a h. (Está claro que además están los términos con h^2, h^3, etc., pero como $h < 1$ entonces $h^2 << h$, $h^3 << h^2$, etc., la influencia de éstos es mucho menor y despreciable.)

Nuevamente, supongamos por un momento que se cumple que todas las derivadas $f^{<i>}(x_0) = 0$ para $i \geq 3$. Entonces tendremos que:

$$\left|f'(x_0) - \frac{f(x_0+h)-f(x_0)}{h}\right| = \frac{h}{2}\left|f''(\xi)\right| \ \text{ con } \xi \in [x; x+h],$$

con lo cual, si conociéramos $f''(\xi)$, podríamos acotar el error que estamos cometiendo por despreciar el término $\frac{h}{2}f''(x_0)$.

Como ejemplo, apliquemos este algoritmo para obtener la derivada en $x_0 = 0,45$ ($f'(0,45)$) de la función $f(x) = \text{sen}(2\pi x)$. Como verificación tomemos el valor analítico de la derivada en

cuestión: $f'(0,45) = 2\pi\cos(2\pi \cdot 0,45) = -5,97566$. Para calcular la aproximación tomemos $h = 0,1$. Así, tendremos.

$$f'(0,45) = \frac{f(0,55) - f(0,45)}{0,1} = \frac{\text{sen}(2\pi \cdot 0,55) - \text{sen}(2\pi \cdot 0,45)}{0,1} = -6,18034.$$

En la tabla 1.2 podemos ver los resultados obtenidos para distintos h.

Tabla 1.2: Valores de $f'(x_0)$ en función de h

h	$\mathbf{f'(x_0)}$	**Error**
10^{-1}	-6,18033988749895	$2,04676 \times 10^{-1}$
10^{-2}	-6,03271072100927	$5,70464 \times 10^{-2}$
10^{-3}	-5,98172474217345	$6,06041 \times 10^{-3}$
10^{-4}	-5,97627391137889	$6,09582 \times 10^{-4}$
10^{-5}	-5,97572532307633	$6,09936 \times 10^{-5}$
10^{-6}	-5,97567042914804	$6,09966 \times 10^{-6}$
10^{-7}	-5,97566494175972	$6,12277 \times 10^{-7}$
10^{-8}	-5,97566438553798	$5,60549 \times 10^{-8}$
10^{-9}	-5,97566451876474	$1,89282 \times 10^{-7}$
10^{-10}	-5,97566607307698	$1,74359 \times 10^{-6}$
10^{-11}	-5,97566995885756	$5,62937 \times 10^{-6}$
10^{-12}	-5,97544236313752	$2,21966 \times 10^{-4}$
10^{-13}	-5,97633054155722	$6,66212 \times 10^{-4}$
10^{-14}	-5,99520433297584	$1,95400 \times 10^{-2}$
10^{-15}	-5,88418203051333	$9,14823 \times 10^{-2}$
10^{-16}	-8,32667268468867	$2,35101$

Si observamos con atención, veremos que el algoritmo utilizado aproxima muy bien el valor buscado hasta $h = 10^{-8}$. Si estimamos la cota de error con $f''(x_0)\frac{10^{-8}}{2}$ obtenemos una valor muy parecido al error indicado en la tabla 1.2 [4]:

$$f''(0,45)\frac{10^{-8}}{2} = 6,09975 \times 10^{-8}\ (5,60549 \times 10^{-8}).$$

Sin embargo, a partir de $h < 10^{-8}$ el error vuelve a crecer. En la figura 1.2 se puede ver como evoluciona el error:

Si analizamos en detalle, vemos que la tendencia del error de truncamiento es lineal (en escala logarítmica) pero para $h < 10^{-8}$ el error aumenta y no sigue una ley determinada. Este «empeoramiento» de la aproximación se debe a la incidencia del error de redondeo, es decir, la unidad de máquina pasa a ser más importante que el error de truncamiento. Es por eso que no siempre el utilizar una «mejor precisión» ayuda a mejorar los resultados finales. En este tipo de problemas, es conveniente que el error que domine los cálculos sea el de truncamiento/discretización.

Veremos más adelante que esta incidencia del paso en el cálculo de una aproximación numérica de la derivada primera, nos alerta de la inestabilidad de la diferenciación numérica, es decir, es muy sensible a la propagación del error de redondeo.

[4] En forma rigurosa deberíamos hallar ξ, pero dado que el intervalo es tan pequeño, puede tomarse x_0.

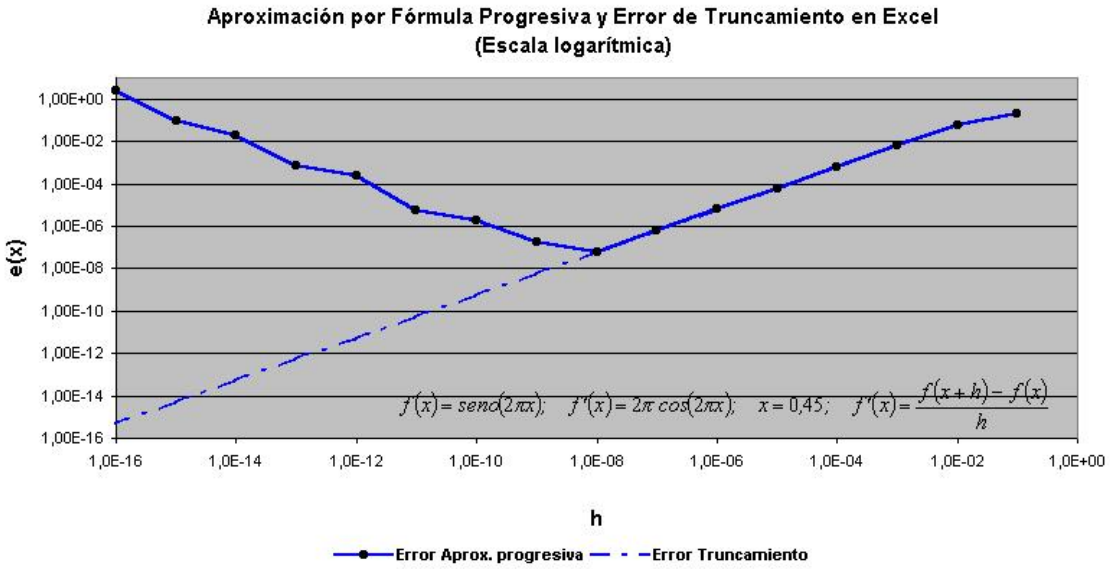


Figura 1.2: Evolución del error del algoritmo.

1.5.4. Errores por «overflow» y «underflow»

Asociados a la representación numérica existen otros dos tipos de errores. Son los denominados errores por «overflow» y por «underflow». Estos errores surgen por las limitaciones de nuestro sistema para representar números muy grandes («overflow») o muy chicos («undeflow»). Es usual que los manuales del usuario de una calculadora indiquen el número más grande (y el más chico) que puede ser representado. Por ejemplo, las calculadoras Casio de la década de los 80 no podían representar $n!$ si $n > 69$ pues el número más grande que podían representar era $9,999999999 \times 10^{99}$ (69!=$1,71122452428141 \times 10^{98}$ y 70!=$1,19785716699699 \times 10^{100}$). Algo similar ocurre con los números muy chicos.

Un error muy común es «olvidarse» que en los cálculos intermedios pueden aparacer números muy grandes o muy chicos, fuera del rango de nuestra representación numérica, que vuelven a un algoritmo inútil. Por ejemplo, supongamos que nuestro sistema de representación numérica en una calculadora represente solamente los números entre -10.000 y -0,0001; y entre 0,0001 y 10.000. Si queremos obtener el resultado de $\sqrt{101^2 - 50}$, como $101^2 = 10.201 > 10.000$ y no lo puede representar, indicará un error por «overflow», es decir, número más grande que el máximo a representar, y cortará la ejecución del algoritmo.

El error por «underflow» es parecido. En este caso, el problema es no poder representar un número muy pequeño, por lo que lo define como cero (0). Si modificamos levemente el ejemplo anterior, y queremos obtener el resultado de $\sqrt{0,01 - 0,006^2}$, como $0,006^2 = 0,000036 < 0,0001$ y no le es posible representarlo, hará $0,006^2 = 0,0000$ y la operación quedará como $\sqrt{0,01 - 0,0} = \sqrt{0,01} = 0,1$.

La diferencia entre ambos es que el error por «overflow» no pasa desapercibido, mientras que el «underflow» sí, y en consecuencia, puede ser más peligroso.

1.6. Propagación de errores

Hemos visto varios ejemplos que nos mostraron en forma evidente la incidencia que pueden llegar a tener los errores en los resultados que entrega un algoritmo, particularmente, el error de redondeo. Veremos a continuación la propagación de dos de los errores más problemáticos, el inherente y el de redondeo.

1.6.1. Propagación del error inherente

Supongamos que tenemos un problema numérico tal que podemos expresarlo como $X \rightarrow Y(X)$, siendo X un vector de $\Re^n$, que corresponde a los datos de entrada, e Y un vector de $\Re^m$, que corresponde a los resultados. Podemos escribir entonces que:

$$X \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \rightarrow Y(X) = \begin{bmatrix} y_1(X) \\ y_2(X) \\ \vdots \\ y_m(X) \end{bmatrix},$$

donde $y_i(X) : \Re^n \rightarrow \Re$; $Y(x) : \Re^n \rightarrow \Re^m$.

Por otra parte, supongamos que en lugar de X conocemos $\tilde{X}$, es decir, una aproximación de X; podemos definir que $e_{x_i} = x_i - \tilde{x}_i$, que también conocemos. Y nuestra última suposición es que las $y_i(X)$ pertenecen a $C^\infty(X)$, lo que nos permite desarrollar $Y(X)$ en una serie de Taylor alrededor de $\tilde{X}$:

$$Y(X) = Y\left(\tilde{X}\right) + \frac{\partial\left[y_1(\tilde{X}); y_2(\tilde{X}); \ldots; y_m(\tilde{X})\right]}{\partial\left[x_1; x_2; \ldots; x_n\right]}(X - \tilde{X}) + T\left(X - \tilde{X}\right).$$

Podemos suponer ahora que $e_{x_i} = x_i - \tilde{x}_1$ para $i \in [1, n]$ es muy pequeño, y que por eso $T\left(X - \tilde{X}\right)$ es despreciable, con lo que nos queda:

$$y_i(X) - y_i\left(\tilde{X}\right) = \sum_{j=1}^{n}\left[\frac{\partial y_i\left(\tilde{X}\right)}{\partial x_j}(x_j - \tilde{x}_j)\right] \quad \text{para} \;\; i = 1; 2; \ldots; m,$$

que por analogía a e_{x_i} podemos expresar como:

$$e_{y_i} = \sum_{j=1}^{n}\frac{\partial y_i\left(\tilde{X}\right)}{\partial x_j}e_{x_j}; \;\; \text{para} \;\; i = 1; 2; \ldots; m,$$

que nos da el error de y_i en función de del error de x_j. Esta expresión es muy útil porque nos permite obtener o determinar el error de un resultado si conocemos el error de los datos de entrada, es decir, *cómo se propagan* los errores inherentes. Veamos algunos ejemplos:

1. **Suma**: Hagamos $y(x_1; x_2) = x_1 + x_2$, entonces tendremos:

$$e_y = e_{x_1+x_2} = \frac{\partial y(\tilde{x}_1; \tilde{x}_2)}{\partial x_1}e_{x_1} + \frac{\partial y(\tilde{x}_1; \tilde{x}_2)}{\partial x_2}e_{x_2},$$

   o sea,

$$e_y = 1 \cdot e_{x_1} + 1 \cdot e_{x_2} \;\Rightarrow\; e_y = e_{x_1} + e_{x_2}.$$

   El error relativo será:

$$e_{r_y} = \frac{e_y}{y} = \frac{e_{x_1} + e_{x_2}}{x_1 + x_2} = \frac{e_{x_1}}{x_1 + x_2} + \frac{e_{x_2}}{x_1 + x_2}.$$

   Sabemos que $e_{x_1} = x_1 \cdot e_{r_{x_1}}$ y $e_{x_2} = x_1 \cdot e_{r_{x_2}}$, por lo que podemos escribir:

$$e_{r_y} = \frac{x_1 \cdot e_{r_{x_1}}}{x_1 + x_2} + \frac{x_2 \cdot e_{r_{x_2}}}{x_1 + x_2} = \frac{x_1}{x_1 + x_2}e_{r_{x_1}} + \frac{x_2}{x_1 + x_2}e_{r_{x_2}}.$$

2. **Producto**: En este caso tenemos $y(x_1; x_2) = x_1 \cdot x_2$, entonces:

$$e_y = x_2 \cdot e_{x_1} + x_1 \cdot e_{x_2}.$$

El error relativo para el producto será:

$$e_{r_y} = \frac{e_y}{y} = \frac{x_2 \cdot e_{x_1}}{x_1 \cdot x_2} + \frac{x_1 \cdot e_{x_2}}{x_1 \cdot x_2} = e_{r_{x_1}} + e_{r_{x_2}}.$$

Hasta aquí no pareciera haber problemas. Sin embargo, raramente se conoce el error con su signo, de ahí que lo que se busca es una *cota* del error, no el error en sí mismo. En ese caso, las expresiones del error relativo se modifican levemente:

1. **Suma**: $e_{r_y} = \frac{|x_1|}{|x_1 + x_2|} \left|e_{r_{x_1}}\right| + \frac{|x_2|}{|x_1 + x_2|} \left|e_{r_{x_2}}\right|$.
2. **Producto**: $e_{r_y} = \left|e_{r_{x_1}}\right| + \left|e_{r_{x_2}}\right|$.

A partir de este razonamiento es que la suma es una operación mal condicionada cuando se da que $|x_1| \approx |x_2|$ y $x_2 < 0$ es decir, la suma algebraica. Suponiendo que $e_{r_{x_i}} \leq r$ se tiene:

$$e_{r_y} = \frac{|x_1| + |x_2|}{|x_1 - x_2|} r.$$

lo que hace que e_{r_y} crezca en forma incontrolada, pues el coeficiente siempre es mayor a uno, y puede ser mucho mayor que 1 si $x_1 - x_2$ es muy chico.

Analizaremos ahora la propagación del error de redondeo.

1.6.2. Propagación del error de redondeo

Supongamos ahora que en nuestro problema no tenemos errores inherentes. Por lo tanto, para $X \to Y(X) : \Re^n \to \Re^m$ sólo tendremos errores de redondeo debido al algoritmo utilizado. Sea $P(X)$ nuestro algoritmo para obtener $Y(X)$. Si no hubieran errores por redondeo, entonces $Y(X) = P(X)$, pero lo que en realidad obtendremos es $\tilde{Y}(X) = P(X)$, es decir que podemos escribir que:

$$Y(X) = \tilde{Y}(X) + E(X) \to y_i(X) = \tilde{y}_i(X) \times \left[1 + \sum_{k=1}^{p} F_{i,k}(X)\epsilon_k\right],$$

con $|\epsilon_k| \leq \eta$, y donde los $F_{i,k}$ son los *factores de amplificación*.

1.6.3. Propagación de los errores inherentes y de redondeo

Ya hemos visto la expresión para calcualr la propagación de los errores inherentes, que es:

$$e_{y_i} = \sum_{j=1}^{n} \frac{\partial y_i(X)}{\partial x_j} e_{x_j} \cong \sum_{j=1}^{n} \frac{\partial y_i(\tilde{X})}{\partial x_j} e_{\tilde{x}_j}.$$

Como además tendremos $P(X)$ en vez de $Y(X)$, entonces:

$$e_{y_i} \cong e_{p_i} = \sum_{j=1}^{n} \frac{\partial p_i(\tilde{X})}{\partial x_j} e_{\tilde{x}_j},$$

y el error relativo será:

$$e_{r_{p_i}} = \frac{\sum_{j=1}^{n} \frac{\partial p_i(\tilde{X})}{\partial x_j} \tilde{x}_j}{p_i(\tilde{X})} e_{r_{\tilde{x}_j}},$$

en consecuencia, el coeficiente que afecta a $e_{r_{\tilde{x}_j}}$ será el *número de condición del problema*, que se define como :

$$C_{p_i} = \frac{\sum_{j=1}^{n} \frac{\partial p_i(\tilde{X})}{\partial x_j} \tilde{x}_j}{p_i(\tilde{X})}.$$

Del mismo modo, tendremos el *término de estabilidad*, que se define como:

$$y_i(X) - p_i(\tilde{X}) = p_i(\tilde{X}) \times \sum_{k=1}^{p} F_{i,k}(\tilde{X})\epsilon_k \Rightarrow T_e = \sum_{k=1}^{p} F_{i,k}(\tilde{X})\epsilon_k \cong \sum_{k=1}^{p} F_{i,k}(\tilde{X})\mu.$$

Si suponemos que $e_{r_{\tilde{x}_j}} \leq r$, entonces, tendremos:

$$e_{r_{y_i}} \cong C_{p_i} \cdot r + T_{e_i} \cdot \mu,$$

que será el **error relativo total**.

Finalmente, si suponemos ahora que $r \cong \mu$, entonces tenemos:

$$e_{r_{y_i}} \cong (C_{p_i} + T_{e_i}) \cdot \mu = C_{p_i} \frac{C_{p_i} + T_{e_i}}{C_{p_i}} \cdot \mu,$$

y podemos decir que un algoritmo es estable si:

$$\frac{C_{p_i} + T_{e_i}}{C_{p_i}} > / > 1 \rightarrow 1 + \frac{T_{e_i}}{C_{p_i}} > / > 1,$$

es decir, *un algoritmo es estable si los errores de redondeo no tienen gran incidencia en el error del resultado o al menos son del mismo orden que los errores inherentes* ($1 + \frac{T_{e_i}}{C_{p_i}} \cong 2$). Sin embargo, esta afirmación debe tomarse con cuidado. Dado que lo que se analiza es la relación $\frac{T_e}{C_p}$, debe tenerse en cuenta que si $C_p >> 1$ y $\frac{T_e}{C_p} \approx 1$ entonces $T_e >> 1$, por lo que es posible que el algoritmo sea inestable.

1.7. Gráfica de proceso

Una forma de obtener los coeficientes C_p y T_e es mediante la «gráfica de proceso». Ésta consiste en un diagrama de flujo que representa gráficamente todo el proceso de una operación dada, permitiendo el análisis de los errores relativos y de redondeo que intervienen en él. No se incluyen en esta gráfica los errores debidos a truncamiento/discretización, que deben ser analizados en forma separada.

En las figuras 1.3 y 1.4 se pueden ver las gráficas de proceso de la suma y el producto.

Analicemos brevemente los errores inherentes y de redondeo en ambos casos. Si nos fijamos en la gráfica de la suma, y tomamos una cota superior para los errores relativos inherentes de x e y, por ejemplo, $|e_{r_x}|\,;|e_{r_y}| < |r|$, entonces el coeficiente C_p se puede escribir como:

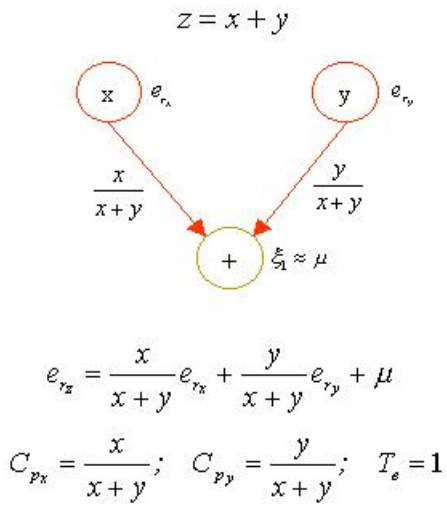
$$C_p = \frac{|x| + |y|}{|x + y|}$$

que es el mismo resultado obtendido antes para la suma. Algo similar se obtiene para el producto.

La ventaja de este método es que facilita el análisis del error de redondeo al introducirlo en cada operación, permitiendo el cálculo del término de estabilidad (T_e). Según estas gráficas, en ambos casos el T_e es igual a 1.

Veamos un ejemplo. Analicemos la propagación de errores del algoritmo inestable ya visto

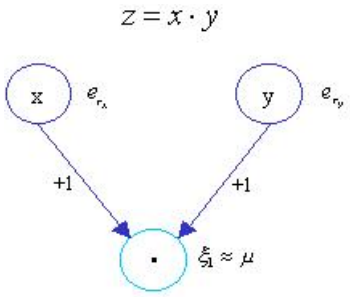
$$y_n = \frac{1}{n} - 10\, y_{n-1},$$



$$e_{r_z} = \frac{x}{x+y} e_{r_x} + \frac{y}{x+y} e_{r_y} + \mu$$

$$C_{p_x} = \frac{x}{x+y}; \quad C_{p_y} = \frac{y}{x+y}; \quad T_e = 1$$

Figura 1.3: Suma.



$$e_{r_z} = e_{r_x} + e_{r_y} + \mu$$

$$C_{p_x} = 1; \quad C_{p_y} = 1; \quad T_e = 1$$

Figura 1.4: Producto.

pero limitándonos a la tercera iteración, con $y_0 = \ln(1,1)$.

Podemos ver lo laborioso que resulta el armado de la gráfica, aún cuando lo hemos limitado hasta obtener el valor y_3.

Supongamos que y_0 no tiene error ($E_{y_0} = 0$) por lo tanto $e_{r_{y_0}} = 0$. También podemos considerar que todas las constantes no tienen errores inherentes, pues no son valores obtenidos por cálculo. En consecuencia, al no existir error inherente, lo único que se propaga es el error de redondeo de cada una de las operaciones. Así, el desarrollo completo de la propagación de los errores resulta ser:

$$\begin{aligned} e_{r_{y_3}} \quad = \quad & \frac{1}{y_3} [100 \cdot \mu_1 + 100 \cdot \mu_3 + 100 \cdot \mu_4 - 5 \cdot \mu_5 - 5 \cdot \mu_6 + 100 \cdot \mu_6 + \\ & -5 \cdot \mu_7 + 100 \cdot \mu_7 + \frac{1}{3} \cdot \mu_8 - 5 \cdot \mu_9 + 100 \cdot \mu_9 + \\ & +1000 \cdot y_0 (\mu_2 - \mu_3 - \mu_4 - \mu_6 - \mu_7 - \mu_8 - \mu_9)] . \end{aligned}$$

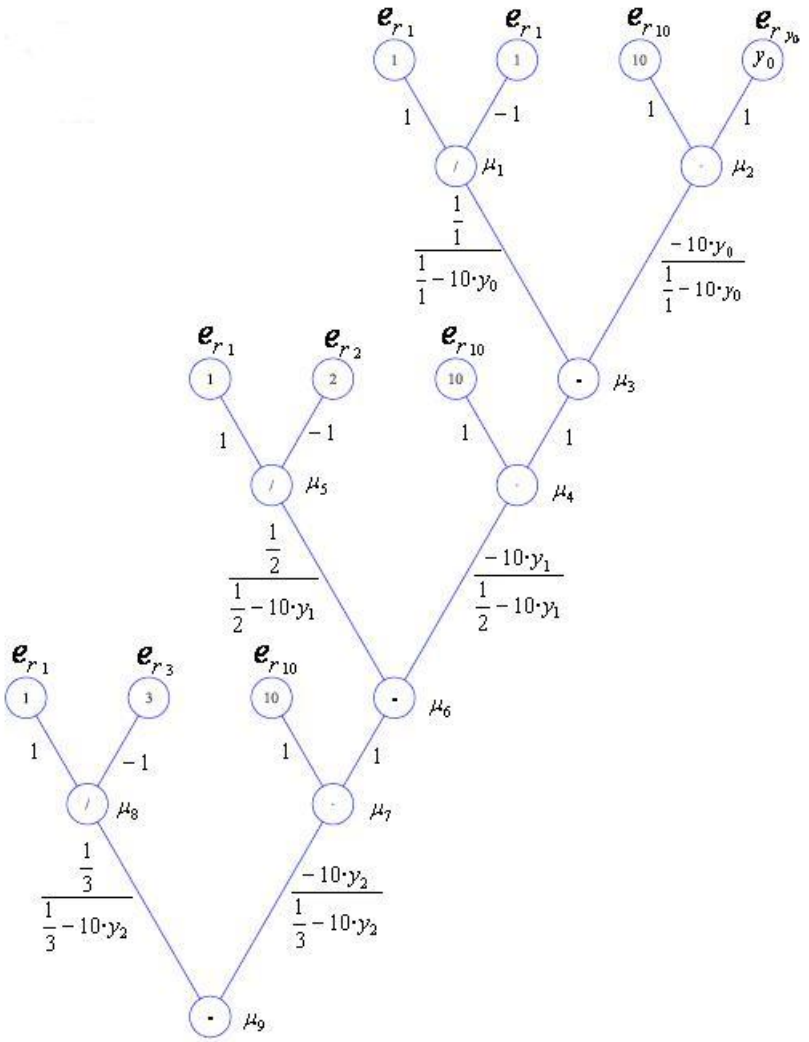


Figura 1.5: Gráfica de proceso del algoritmo.

Si además imponemos que $\mu_i < \mu$, y con esto definimos nuestra cota de error relativo, tendremos que

$$e_{r_{y_3}} = \frac{1}{y_3}\left(620 + 6000 \cdot y_0\right) \cdot \mu = \frac{1}{y_3}\left[620 + 6000 \cdot \ln(1,1)\right] \cdot \mu.$$

Como

$$y_3 = \frac{1}{3} - 10 \cdot \underbrace{\left[\frac{1}{2} - 10 \cdot \underbrace{(1 - 10 \cdot y_0)}_{y_1}\right]}_{y_2},$$

el valor de y_3 podemos escribirlo como:

$$y_3 = 95,3333\ldots - 1000 \cdot \ln(1,1).$$

De esta forma, tenemos que nuestro coeficiente T_e resulta ser

$$T_e = \frac{620 + 6000 \cdot \ln(1,1)}{95,3333\ldots - 1000 \cdot \ln(1,1)},$$

y si reemplazamos los valores numéricos, obtenemos el siguiente coeficiente de estabilidad para el caso de y_3:

$$T_e \approx 25319.$$

Resulta evidente que el algoritmo es inestable para cualquier valor de y_0. Si analizamos el coeficiente de condición (C_p), obtendremos lo siguiente:

$$C_p = \frac{1000 \cdot y_0}{95,3333\ldots - 1000 \cdot \ln(1,1)}.$$

Cuando reemplazamos los valores obtenemos que

$$C_p \approx 4116;$$

y si analizamos la relación entre C_p y T_e nos queda que

$$1 + \frac{T_e}{C_p} = 1 + \frac{25319}{4116} \approx 7,15 > 2;$$

lo que nos muestra que el algoritmo es inestable y, ciertamente, mal condicionado.

Vimos que la gráfica de proceso es bastante útil para obtener ambos coeficientes, pero también que puede convertirse en algo muy difícil de desarrollar cuando el algoritmo con miles (o millones) de pasos, como puede ser la resolución de un sistema de ecuaciones lineales mediante un método directo. Analizar millones de operaciones mediante la gráfica de proceso puede ser una tarea imposible. Por lo tanto, debemos buscar otra manera de estimar ambos coeficientes.

1.8. Perturbaciones experimentales

Supongamos que queremos estudiar la condición o la estabilidad de un algoritmo con miles de pasos. Ya dijimos que hacer la gráfica de proceso puede ser una tarea imposible. Entonces, ¿cómo hacemos para saber si dicho algoritmo está bien condicionado o es estable? Veamos. Para empezar, estudiemos cómo obtener una aproximación de la condición del problema. Puesto que la condición viene dada por la propagación (o no) de los errores relativos inherentes, busquemos la manera de obtener en forma numérica una estimación del coeficiente de condición, o sea, del C_p. En el mismo sentido, el término de estabilidad, T_e está relacionado con la propagación de los errores de redondeo. Busquemos también algún procedimiento que nos permita obtener una estimación de dicho coeficiente.

1.8.1. Estimación del número de condición

Partamos de la expresión final del error relativo de un resultado:

$$e_r = C_p\, r + T_e\, \mu$$

y supongamos por un momento que no tenemos errores de redondeo, es decir, despreciamos $T_e\, \mu$. En consecuencia, lo que tendremos es:

$$e_r = C_p\, r \;\Rightarrow\; C_p = \frac{e_r}{r}$$

Y con esto podemos estimar valor del C_p. ¿Cómo lo hacemos? Perturbando los valores de los datos de entrada. La idea es la siguiente: se toman los datos de entrada (x, y, etc.), y se aplica el algoritmo a analizar, obteniendo el resultado correspondiente. Luego se «perturban» los datos de entrada, es decir, se les incorpora un error. Con estos datos de entrada, se vuelve a calcular un resultado, que seguramente diferirá del anterior, pues los datos no son iguales. Este último

paso se puede hacer varias veces introduciendo distintas perturbaciones (errores) a los datos de entrada.

Una vez obtenidos los distintos valores de los resultados, tomamos el resultado sin perturbar como resultado «exacto», con el cual vamos a calcular los errores relativos de los otros resultados «perturbados». Con cada uno de éstos obtendremos diferentes e_{r_i}. Como además tendremos diferentes r_i, lo que obtendremos finalmente son diferentes C_{p_i}. Como hemos supuesto que los errores de redondeo son despreciables, todos los C_{p_i} deberían ser similares, con lo cual tendremos una estimación de la condición del problema, es decir, estimamos un C_p. Con esta estimación podremos establecer si el problema está bien o mal condicionado.

Veamos un ejemplo. Tomemos la siguiente función para calcular sen(x):

$$f(x) = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880},$$

función obtenida a partir del truncamiento de la serie de MacLaurin. Con ella calculemos sen($\frac{\pi}{4}$) y luego perturbemos el dato de entrada.

El primer resultado lo obtenemos con $x = \frac{\pi}{4}$:

$$f\left(\frac{\pi}{4}\right) = \frac{\pi}{4} - \frac{\left(\frac{\pi}{4}\right)^3}{6} + \frac{\left(\frac{\pi}{4}\right)^5}{120} - \frac{\left(\frac{\pi}{4}\right)^7}{5040} + \frac{\left(\frac{\pi}{4}\right)^9}{362880} = 0,70711$$

Perturbemos ahora x haciendo $x_1 = x \cdot (1 + 0,001)$ ($r_1 = 0,001$), y calculemos $f(x_1)$:

$$f\left(\frac{\pi}{4} \cdot (1 + 0,001)\right) = 0.70655$$

Introduzcamos una nueva perturbación, esta vez haciendo $x_2 = x \cdot (1 - 0,001)$ ($r_2 = -0,001$), y calculemos $f(x_2)$:

$$f\left(\frac{\pi}{4} \cdot (1 - 0,001)\right) = 0.70766$$

Ahora calculemos los dos C_p. Para el primer caso tenemos:

$$C_p = \frac{0,70711 - 0,70655}{0,70711} \cdot \frac{1}{0,001} = 0,78571$$

Para el segundo caso tenemos:

$$C_p = \frac{0,70711 - 0,70766}{0,70711} \cdot \frac{1}{-0,001} = 0,78509$$

Si calculamos el C_p en forma analítica obtenemos:

$$C_p = \frac{\partial f(x)}{\partial x} = \frac{d\, f(x)}{d\, x} = 1 - \frac{\left(\frac{\pi}{4}\right)^2}{2} + \frac{\left(\frac{\pi}{4}\right)^4}{24} - \frac{\left(\frac{\pi}{4}\right)^6}{720} + \frac{\left(\frac{\pi}{4}\right)^8}{40320} \approx cos\left(\frac{\pi}{4}\right) \Rightarrow C_p \approx 0,78540$$

Esto demuestra que la estimación del C_p es muy buena y que el problema está bien condicionado, pues $C_p < 1$ [5].

1.8.2. Estimación del término de estabilidad

Para obtener una estimación del término de estabilidad, seguiremos un esquema similar al visto para el número de condición. Partamos nuevamente de la expresión final para el error realtivo:

$$e_r = C_p\, r + T_e\, \mu$$

[5] De hecho, las calculadoras poseen algoritmos de este tipo para obtener los valores de las funciones trigonométricas y trascendentes.

Ahora consideremos como hipótesis que los errores inherentes son despreciables, por lo que podemos decir que el error relativo es:

$$e_r = T_e\, \mu.$$

El error relativo está definido como:

$$e_r = \frac{y - \bar{y}}{y},$$

por lo tanto podemos escribir:

$$\frac{y - \bar{y}}{y} = T_e\, \mu.$$

Al calcular el valor de y con dos «precisiones» diferentes t y s, ($\mu_s = 10^{1-s}$ y $\mu_t = 10^{1-t}$), y asumiendo que $t > s$, obtenemos los siguientes errores relativos:

$$e_{r_t} = \frac{y - \bar{y}_t}{y} = T_e\, \mu_t;\ e_{r_s} = \frac{y - \bar{y}_s}{y} = T_e\, \mu_s.$$

Si restamos e_{r_t} a e_{r_s} tenemos:

$$e_{r_s} - e_{r_t} = \frac{\bar{y}_t - \bar{y}_s}{y} = T_e\, (\mu_s - \mu_t),$$

de donde despejamos T_e:

$$T_e = \frac{\bar{y}_t - \bar{y}_s}{y\, (\mu_s - \mu_t)}.$$

Como el valor de y no lo conocemos, tomamos $\bar{y}_t$ en su lugar. En consecuencia, la expresión queda:

$$T_e = \frac{\bar{y}_t - \bar{y}_s}{\bar{y}_t\, (\mu_s - \mu_t)}.$$

Esta expresión nos permite obtener una estimación del T_e calculando dos aproximaciones de y, $\bar{y}_t$ y $\bar{y}_s$, con diferente precisión, utilizando el mismo algoritmo.

Como ejemplo, utilicemos el mismo algoritmo del caso anterior. Calculemos el valor de $\operatorname{sen}\left(\frac{\pi}{4}\right)$ con tres precisiones distintas: $s = 4$; $t = 8$ y $u = 15$. Para cada caso tendremos: $\bar{y}_s = 0,706$; $\bar{y}_t = 0,7071068$ y $\bar{y}_u = 0,70710678293687$. Con estos valores calculamos los T_e, tomando como valor de referencia $\bar{y}_u$. Así, obtenemos los siguientes valores:

$$T_{e_s} = \frac{\bar{y}_u - \bar{y}_s}{\bar{y}_u\, (\mu_s - \mu_u)} = \frac{0,70710678293687 - 0,706}{0,70710678293687\, (10^{-3} - 10^{-14})} = 1,565;$$

y

$$T_{e_t} = \frac{\bar{y}_u - \bar{y}_t}{\bar{y}_u\, (\mu_t - \mu_u)} = \frac{0,70710678293687 - 0,7071068}{0,70710678293687\, (10^{-7} - 10^{-14})} = 0,241.$$

Si analizamos un poco los valores obtenidos, vemos que en el primer caso el error de redondeo se amplifica, puesto que el T_e es mayor que 1. En cambio, en el segundo, la situación es muy buena porque los errores se mantienen acotados, no se amplifican ($T_e < 1$). Podríamos decir que calcular el valor de y con más precisión mejora el resultado final, pero hemos vistos que no siempre esto es cierto.

1.9. Inestabilidad en los algoritmos

Como hemos dicho, uno de los objetivos del análisis numérico es obtener algoritmos que estén bien condicionados y sean estables. Hasta ahora nos hemos referido a los principales errores que afectan a los algoritmos y hemos analizado los distintos errores y su propagación, según sea el caso. Además, hemos visto que la condición de un problema es independiente del algoritmo,

en tanto que la estabilidad es una «propiedad» el mismo. Es por eso que el análisis numérico se concentra más en estudiar cómo hacer que un algoritmo sea estable más que en analizar su condicionamiento, aunque en algunos casos este último análisis sea muy importante, como por ejemplo, para resolver sistemas de ecuaciones lineales.

La mayoría de los libros y cursos de análisis numérico hacen hincapié en varios conceptos para obtener un algoritmo estable. Alguno de éstos son:

1. La resta de dos números muy similares (cancelación) siempre debe ser evitada.
2. El problema del error de redondeo es su acumulación.
3. Aumentar la precisión en los cálculo mejora la exactitud de los resultados.

Según N. Higham (véase [5], cápitulo 1), estos conceptos son en realidad malos entendidos, y desarrolla algunos ejemplos que muestran que no siempre es así. Veamos alguno de ellos.

1.9.1. Cancelación

En su libro, Higham presenta el siguiente caso. Supongamos que debemos hacer la siguiente operación:

$$f(x) = \frac{1 - \cos(x)}{x^2},$$

con $x = 1,2 \times 10^{-5}$ y con $\cos(x) = c$ redondeado a 10 dígitos significativos, con un valor de

$$c = 0,9999999999;$$

de manera que

$$1 - c = 0,0000000001.$$

Al calcular $f(x) = \frac{1-c}{x^2}$ se obtiene $f(x) = \frac{10^{-10}}{1,44 \times 10^{-10}} = 0,6944\ldots$, resultado evidentenente incorrecto pues es claro que $0 \leq f(x) \leq 1/2$ para todo $x \neq 0$.

Al analizar la cota del error relativo para la resta $\hat{x} = \hat{a} - \hat{b}$, donde $\hat{a} = a(1 + \Delta a)$ y $\hat{b} = b(1 + \Delta b)$ obtiene:

$$\left| \frac{x - \hat{x}}{x} \right| = \left| \frac{-a\Delta a + b\Delta b}{a - b} \right| \leq max(|\Delta a|, |\Delta b|) \frac{|a| + |b|}{|a - b|}.$$

La cota del error relativo de $\hat{x}$ es muy grande cuando $|a - b| << |a| + |b|$. Por lo tanto, afirma que una resta con esta condición *da preeminencia a los errores iniciales.*

También afirma que la cancelación no siempre es mala, por varias razones. La primera es que los números a restar pueden ser libres de error. La segunda, que la cancelación puede ser una señal de un problema intrínsecamente mal condicionado y, por lo tanto, inevitable. Tercero, los efectos de la cancelación dependen del contexto en que se efectúa. Si $x >> y \approx z > 0$, la resta en la operación $x + (y - z)$ es inocua.

1.9.2. Acumulación del error de redondeo

Desde que se creó la primera computadora, la acumulación del error de redondeo ha sido uno de los «dolores de cabeza» de los especialistas, como se puede ver en esta frase: "La extraordinaria rapidez de las actuales máquinas significa que en un problema típico se realizan millones de operaciones con coma (punto) flotante. Esto quiere decir que la acumulación de errores de redondeo puede ser desastrosa". Para Higham esta afirmación, si bien cierta, no es del todo correcta o está mal enfocada. En muchas ocasiones la inestabilidad está dada por la incidencia de unos pocos errores de redondeo y no por la acumulación de millones de ellos. Un ejemplo en ese sentido está dado por el algoritmo del ejemplo inicial, en el cual el error está

dado por el redondeo de y_{n-1}, que se propaga a medida que el valor es cada vez más chico. Otro ejemplo es el cálculo de e usando su definición:

$$f(n) = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n,$$

tomando n finito pero lo suficientemente grande. En la tabla 1.3 podemos ver los resultados para distintos n obtenidas en MS Excel®.

Tabla 1.3: Valores de $f(n)$ y diferencia con e.

$\mathbf{n}$	$\mathbf{f(n)}$	$\mathbf{\lvert e - f(n)\rvert}$
10^1	2,593742460100000	$1,24539 \times 10^{-1}$
10^2	2,704813829421530	$1,34680 \times 10^{-2}$
10^3	2,716923932235590	$1,35790 \times 10^{-3}$
10^4	2,718145926824930	$1,35902 \times 10^{-4}$
10^5	2,718268237192300	$1,35913 \times 10^{-5}$
10^6	2,718280469095750	$1,35936 \times 10^{-6}$
10^7	2,718281694132080	$1,34327 \times 10^{-7}$
10^8	2,718281798347360	$3,01117 \times 10^{-8}$
10^9	2,718282052011560	$2,23553 \times 10^{-7}$
10^{10}	2,718282053234790	$2,24776 \times 10^{-7}$
10^{11}	2,718282053357110	$2,24898 \times 10^{-7}$
10^{12}	2,718523496037240	$2,41668 \times 10^{-4}$
10^{13}	2,716110034086900	$2,17179 \times 10^{-3}$
10^{14}	2,716110034087020	$2,17179 \times 10^{-3}$
10^{15}	3,035035206549260	$3,16753 \times 10^{-1}$

Como podemos observar, a medida que n aumenta, mejora la aproximación de e. Sin embargo, eso ocurre sólo para $n < 10^8$. Cuando $n \geq 10^9$ la aproximación se vuelva cada vez peor, como es el caso de $n = 10^{15}$. Al igual que en el ejemplo ya citado, el problema es la imposibilidad de representar correctamente $\frac{1}{n}$ cuando n es muy grande y, en consecuencia, un solo error de redondeo incide negativamente en el resultado obtenido.

1.9.3. Aumento de la precisión

El caso anterior muestra también que el aumento de la precisión no siempre significa una mejora en los resultados obtenidos. Es usual que cuando la única fuente de error es el redondeo, la forma tradicional de corregir esto es aumentar la precisión y ver qué ocurre con los resultados, comparando cuántos dígitos coinciden en los resultados original y con mayor precisión.

Pero en el caso de trabajar con un problema mal condicionado, el aumento de la precisión no resulta en una mejora en los resultados. En ese caso, es muy posible que los resultados obtenidos no tengan ningún dígito en común. Un ejemplo típico es el siguiente. Supongamos que resolvemos el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 10^{-4} & 2 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}.$$

Si utilizamos dos precisiones diferentes para resolver el sistema, una con cuatro decimales y otra con tres, obtenemos los siguientes vectores [x]:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 = \begin{bmatrix} 0,01 \\ 2 \end{bmatrix} \text{ con tres decimales,} \qquad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_2 = \begin{bmatrix} 1,0001 \\ 2 \end{bmatrix} \text{ con cuatro decimales.}$$

Vemos que el aumento de la precisión nos da un resultado completamente distinto para la primera componente y por consiguiente, no son comparables. Este un típico caso de una matriz considerada como «mal condicionada» y que debemos transformarla para obtener resultados mejores. Así, si intercambiamos filas tenemos:

$$\begin{bmatrix} 1 & 1 \\ 10^{-4} & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix},$$

la solución que obtenemos es:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1,0 \\ 2,0 \end{bmatrix}$$

cualquiera sea la precisión utilizada y que corresponde a la solución correcta.

Es evidente que el aumento en la precisión de los coeficientes no mejora los resultados. Este es un caso especial de matrices cuya solución merece un estudio más detallado que se verá en Sistemas de Ecuaciones Lineales.

1.10. Diseño de algoritmos estables

El análisis de los errores y, fundamentalmente, de la propagación de estos errores, nos ayuda a obtener algunos lineamientos para diseñar algoritmos estables, si bien no hay «recetas» simples para ello. La mejor recomendación es estar alerta en obtener un algoritmo estable cuando se lo diseña y no concentrarse solamente en otras cuestiones, como el costo computacional o la posibilidad de su «paralelización».

En su libro, Higham da una serie de lineamientos, entre los cuales se destacan los siguientes:

1. Evitar la resta de cantidades con errores.

2. Minimizar el tamaño de las cantidades intermedias relativas al resultado final. La razón es que si las cantidades intermedias son demasiado grandes, el resultado final puede ser consecuencia de una resta dañina. O visto de otra manera, cantidades grandes «tapan» los datos iniciales y en consecuencia, se pierde información.

3. Es más ventajoso escribir una expresión que actualice la información como

   $$\text{valor}_{\text{nuevo}} = \text{valor}_{\text{viejo}} + \text{pequeña corrección}$$

   si la pequeña corrección se puede calcular con muchos dígitos significativos [6]. Muchos de los métodos numéricos se expresan de esta forma, como por ejemplo, el método de Newton-Raphson, el Método de los Gradientes Conjugados para resolver sistemas de ecuaciones lineales, etc. Un ejemplo clásico es el método del refinamiento iterativo de la solución para un sistema de ecuaciones lineales de la forma $Ax = B$, en el que se calcula el residuo $r_1 = B - A\tilde{x}_1$, y con él un valor δ_1 resolviendo $A\delta_1 = r_1$, para luego mejorar el resultado obtenido con la iteración $\tilde{x}_2 = \tilde{x}_1 + \delta_1$.

4. Usar transformaciones bien condicionadas.

Una recomendación importante es que se revisen los resultados intermedios, es decir, los que se generan durante el procedimiento de cálculo. Esta práctica era muy común en los inicios de la computación electrónica. En su libro, Higham señala lo siguiente:

[6] Sin embargo, Higham mismo reconoce que no es necesario operar con muchos dígitos significativos para obtener buenos resultados utilizando este procedimiento. Véase [6]

Wilkinson, el padre del análisis de la propagación de errores, ganó una gran experiencia respecto a la estabilidad numérica gracias a ese tipo de revisión. Es irónico que con las grandes facilidades que se tienen hoy para rastrear los pasos de un algoritmo (ventanas múltiples, herramientas gráficas, impresoras rápidas), a veces se obtengan menos resultados que en esa época en las cuales sólo se contaba con papel y lámparas (válvulas).

Capítulo 2

Sistemas de Ecuaciones Lineales

2.1. Introducción

Una de las características fundamentales del uso de las computadoras es la dificultad para trabajar con método simbólicos. Si bien hoy existen varios programas que trabajan con matemática simbólica (Mathematica, Maple, MathCAD), no es lo más usual y mucha veces la capacidad de esos programas se ve excedida por las demanadas ingenieriles en cantidad de cálculo. Más de una vez la necesidad de obtener un resultado en el menor tiempo posible hace imperioso contar con algún método que estime el valor en forma numérica.

Buena parte de los problemas ingenieriles de la actualidad hacen un uso intensivo de sistemas de ecuaciones lineales, usualmente definidos como $Ax = B$. En particular, el uso extendido de programas que aplican el método de los elementos finitos o de las diferencias finitas es un ejemplo de ello. En esos programas, como los de análisis estructural, el núcleo principal del programa es la resolución de sistemas de ecuaciones lineales de grandes dimensiones ($1.000 \times 1.000, 10.000 \times 10.000$, etc.). En este tipo de problemas no resulta muy eficiente invertir la matriz de coeficientes para hallar la solución del sistema. También la aplicación de métodos de regresión múltiple requieren la solución de sistemas de ecuaciones lineales, algo usual en estadísitica. Podemos decir, entonces, que en ingeniería el uso de sistemas de ecuaciones lineales es una práctica habitual.

Por lo tanto, uno de los temas más importantes del análisis numérico es el estudio de la resolución de estos sistemas de ecuaciones. Si bien conocemos métodos muy precisos (exactos) para resolver sistemas de pequeñas dimensiones, el problema es analizar cómo resolver sistemas de grandes a muy grandes dimensiones.

Del álgebra lineal sabemos que podemos obtener la solución de $Ax = B$ si hacemos $x = A^{-1}B$, pero obtener la inversa de A no es una tarea sencilla, más si la matriz no sigue un patrón determinado o si está *mal condicionada*, concepto que estudiaremos más adelante.

Como introducción y repaso, veremos primero algunas definiciones para luego estudiar varios métodos que resuelven un sistema de ecuaciones sin invertir la matriz de coeficientes de manera muy eficiente y para distintas condiciones.

2.2. Definiciones

Empezaremos dar algunas definiciones relacionadas con los vectores y las matrices.

Definición 2.1. Una matriz que tiene la misma cantidad de filas que de columnas (A es de $n \times n$ dimensiones) se denomina *matriz cuadrada*.

Para que una matriz pueda tener inversa debe ser necesariamente cuadrada.

Definición 2.2. Una matriz cuyo determinante es no nulo ($det(A) \neq 0$) se denomina *matriz no singular*.

Definición 2.3. Una matriz A cuadrada tiene inversa, es decir, existe A^{-1}, si A es una matriz *no singular*.

A partir de esta última definición podemos decir que un sistema de ecuaciones lineales tiene solución única si la matriz A del sistema $Ax = B$ es *cuadrada* y *no singular*.

Definición 2.4. Se denomina *rango de un matriz* al número de filas que son linealmente indepediente.

Por lo tanto, el rango de una matriz cualquiera siempre es menor o igual al número de filas ($rango(A) \leq$ número de filas). De esto último se puede inferir que una matriz A de $n \times n$ dimensiones es no singular si su rango es n ($rango(A) = n$). Si el vector B se puede escribir como combinación lineal de las columnas de la matriz A y la matriz A es singular, entonces existen infinitas soluciones para el sistema.

Definición 2.5. Una norma vectorial en $\Re^n$ es una función, $\|\cdot\|$, de $\Re^n$ en $\Re$, con las siguientes propiedades:

1. $\|x\| > 0$ para todo $x \in \Re^n$;
2. $\|x\| = 0$ si y sólo si $x \equiv 0$ ($x = [0; 0; \ldots; 0]^T$);
3. $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$ para todo $\alpha \in \Re$ y $x \in \Re^n$, y;
4. $\|x + y\| \leq \|x\| + \|y\|$ para todo $x; y \in \Re^n$.

Definición 2.6. Las normas l_2 y l_∞ de un vector están definidas por:

1. $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$ (también llamada norma euclídea);
2. $\|x\|_\infty = \max\limits_{1 \leq i \leq n} |x_i|$.

Definición 2.7. Una norma matricial sobre un conjunto de todas las matrices $n \times n$ es una función de valor real, $\|\cdot\|$, definida en este conjunto y que satisface para todas las matrices A y B de $n \times n$ y todos los números reales α:

1. $\|A\| > 0$;
2. $\|A\| = 0$ si y sólo si $A \equiv 0$, es decir, A es la matriz nula;
3. $\|\alpha \cdot A\| = |\alpha| \cdot \|A\|$;
4. $\|A + B\| \leq \|A\| + \|B\|$;
5. $\|A \cdot B\| \leq \|A\| \cdot \|B\|$.

2.3. Matrices triangulares

Una matriz triangular es aquella que sólo tiene coeficientes no nulos en la diagonal principal y por encima o por debajo de ella. Hay dos tipos: la matriz triangular superior, generalmente denominada U, y la matriz triangular inferior, denominada L. Estas matrices son muy convenientes cuando se deben resolver sistemas de ecuaciones lineales puesto que permiten una rápida obtención de los resultados sin la necesidad de invertir la matriz de coeficientes A. Estos dos tipos de matrices dan lugar a dos métodos muy utilizados: la sustición inversa, para matrices U, y la sutitución directa, para matrices L.

Por ejemplo, para el primer caso, una matriz U de 4×4 tiene la siguiente forma:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Para resolver un sistema $Ux = B$ basta con empezar por la última fila para obtener x_4 y luego ir reemplazando este valor en las ecuaciones anteriores, es decir, hacer:

$$\begin{aligned} x_4 &= \frac{b_4}{u_{44}} \\ x_3 &= \frac{b_3 - u_{34} \cdot x_4}{u_{33}} \\ &\vdots \\ x_i &= \frac{b_i - \sum_{j=i+1}^{n} u_{ij} \cdot x_j}{u_{ii}} \end{aligned}$$

Esta forma de resolver el sistema de ecuaciones lineales se denomnina *sustitución inversa*.

Cuando la matriz es triangular inferior el procedimiento para rsolver $Lx = B$ es:

$$\begin{aligned} x_1 &= \frac{b_1}{l_{11}} \\ x_2 &= \frac{b_2 - l_{21} \cdot x_1}{l_{22}} \\ &\vdots \\ x_i &= \frac{b_i - \sum_{j=1}^{i-1} l_{ij} \cdot x_j}{l_{ii}} \end{aligned}$$

En este caso, el método se denomina *sustitución directa*.

Cualquiera de estos métodos es sencillo de aplicar y evita tener que invertir la matriz de coeficiente de un sistema de ecuaciones lineales, lo que facilita la resolución del mismo. En consecuencia, los métodos directos se basan en transformar la matriz de coeficientes original no triangular, en una nueva matriz de coeficientes triangular.

2.4. Eliminación de Gauss y sustitución inversa

El método de eliminación de Gauss es un método directo muy efectivo que transforma la matriz de coeficientes original en una matriz triangular superior y luego aplica el método de sustitución inversa para obtener la solución del sistema dado. Para ello se basa en la propiedad que tienen las matrices de que la misma no cambia si se reemplaza una de las filas por una combinación lineal de las restantes filas. El procedimiento en líneas generales es:

- Se fija la primera fila de la matriz A.
- Se tranforman las filas siguientes de manera de que el coeficiente a_{i1} se anule, es decir, se utiliza el coeficiente a_{11} de la diagonal principal como *pivote*.
- Se fija la siguiente fila, se fija el pivote en la diagonal principal y se repite el paso anterior.
- Se continúa hasta que la matriz queda transformada en una matriz triangular superior.

- Se aplica la sustitución inversa para hallar los x_i.

Por ejemplo, supongamos que tenemos la siguiente matriz A de dimensiones $n = 4$, con su vector independiente B, generamos la matriz ampliada:

$$A = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{array}\right].$$

Para obtener la nueva segunda fila operamos de la siguiente manera:

1. Calculamos el coeficiente m_{21}:
$$m_{21} = \frac{a_{21}}{a_{11}}$$

2. Luego calculamos los coeficientes a^*_{2i} y b^*_2:
$$\begin{aligned} a^*_{22} &= a_{22} - m_{21} \times a_{12} \\ a^*_{23} &= a_{23} - m_{21} \times a_{13} \\ a^*_{24} &= a_{24} - m_{21} \times a_{14} \\ b^*_2 &= b_2 - m_{21} \times b_1 \end{aligned}$$
y así sucesivamente para el resto de las filas, con lo que otenemos la nueva matriz ampliada:
$$A = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a^*_{22} & a^*_{23} & a^*_{24} & b^*_2 \\ 0 & a^*_{32} & a^*_{33} & a^*_{34} & b^*_3 \\ 0 & a^*_{42} & a^*_{43} & a^*_{44} & b^*_4 \end{array}\right],$$
y los correspondientes m_{31} y m_{41}.

3. El siguiente paso es repetir los pasos 1 y 2, es decir, calcular un nuevo coeficiente m, el m_{32} y los nuevos coeficientes. Operando sucesivamente de esta forma obtendremos finalmente la siguiente matriz ampliada :
$$A = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a^*_{23} & a^*_{23} & a^*_{24} & b^*_2 \\ 0 & 0 & a^{\#}_{33} & a^{\#}_{34} & b^{\#}_3 \\ 0 & 0 & 0 & a^{+}_{44} & b^{+}_4 \end{array}\right],$$
y los correspondientes m_{42} y m_{43}.

4. Finalmente, para obtener el vector x debemos hacer:
$$\begin{aligned} x_4 &= \frac{b^+_4}{a^+_{44}} \\ x_3 &= \frac{b^{\#}_3 - a^{\#}_{34} x_4}{a^{\#}_{33}} \\ x_2 &= \frac{b^*_2 - a^*_{23} x_3 - a^*_{24} x_4}{a^*_{22}} \\ x_1 &= \frac{b_1 - a_{12} x_2 - a_{13} x_3 - a_{14} x_4}{a_{11}} \end{aligned}$$

La expresión general para la transformación de la matriz es la siguiente:

$$a_{ij}^* = a_{ij} - m_{il} \times a_{lj},$$

para los coeficientes de la matriz A, y:

$$b_i^* = b_i - m_{il} \times b_l$$

para los coeficientes del vector de términos independientes (B), con $m_{il} = \frac{a_{il}}{a_{ll}}$.

Este procedimiento es muy útil puesto que se conoce exactamente la cantidad de pasos que deben efectuarse, es decir, el método tiene un cantidad *finita* de pasos, inclusive si el sistema a resolver cuenta con varios vectores B. En ese caso, basta con transformarlos conjuntamente con la matriz A.

Con este procedimiento, es posible conocer el «costo computacional» del método, es decir, establecer cuanto tiempo lleva todo el proceso. Una forma de estimar este costo de transformación de la matriz en triangular superior es mediante la siguiente expresión que cuenta las operaciones realizadas (sumas, restas, multiplicaciones y divisiones). Para la transformación de la matriz A ampliada con el vector B en una matriz triangular superior tenemos la siguiente cantidad de operaciones:

$$\sum_{k=1}^{n-1} \left[(n-k) + 2\,(n-k)\,(n-k+1) \right] = \frac{2}{3}n^3 + \frac{n^2}{2} - \frac{7}{6}n.$$

A su vez, para la sustición inversa tenemos esta cantidad de operaciones:

$$1 + \sum_{k=1}^{n-1} \left[2\,(n-k) + 1 \right] = n^2.$$

En consecuencia, si se suman ambos valores, tenemos que el costo de efectuar la eliminación de Gauss es:

$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n;$$

es decir, proporcional a n^3.

Conviene tener presente que esta estimación es aproximada, pues no se han tenido en cuenta otros «costos» difíciles de evaluar como son el manejo de las prioridades de memoria, la forma de guardar los datos, etc. Sin embargo, esta estimación sirve para establecer que a medida que la dimensión de la matriz aumenta, el costo es proporcional al cubo de la misma, es decir, el aumento del tiempo empleado en resolver el sistema completo (el «costo computacional») es potencial y no lineal. Es por ello que resolver un sistema de 1.000×1.000 insume un costo proporcional a 1.000.000.000 operaciones.

Un problema que puede surgir en este método es si alguno de los elementos de la diagonal principal al ser transformados se anulan. Si esto ocurriera, de acuerdo con el algoritmo anterior, el procedimiento se detendría y en consecuencia no podría obtenerse solución alguna. En estos casos se aplican versiones más desarrolladas, denominadas *Eliminación de Gauss con Pivoteo Parcial* (EGPP) o *Eliminación de Gauss con Pivoteo Total* (EGPT).

En el primer caso, lo que se hace es primero intercambiar las filas, reordenándolas de manera tal que el coeficiente nulo quede fuera de la diagonal principal, y luego se continúa con el algoritmo tradicional. Veamos un ejemplo. Supongamos el siguiente sistema:

$$\begin{aligned} x_1 + x_2 - x_3 &= 1 \\ x_1 + x_2 + 4x_3 &= 2 \\ 2x_1 - x_2 + 2x_3 &= 3. \end{aligned}$$

Armemos el sistema ampliado para aplicar el método de Eliminación de Gauss. Entonces nos queda:

$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 1 \\ 1 & 1 & 4 & 2 \\ 2 & -1 & 2 & 3 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 1 & -1 & 1 \\ 0 & 0 & 5 & 1 \\ 0 & -3 & 4 & 1 \end{array}\right].$$

Como vemos, la transformación de la matriz nos deja nulo el coeficiente a_{22}^* de la segunda fila, lo que nos impide seguir operando. Para poder seguir debemos intercambiar las filas dos y tres, en consecuencia tendremos:

$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 1 \\ 0 & -3 & 4 & 1 \\ 0 & 0 & 5 & 1 \end{array}\right] \Rightarrow \left\{\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right\} = \left[\begin{array}{c} 1,2667 \\ -0,0667 \\ 0,2000 \end{array}\right].$$

El intercambio entre las filas 2 y 3 evitó que el procedimiento se detuviera. Pero también es posible que valores muy chicos en los coeficientes de la diagonal principal generen un problema en la mecánica del sistema. Por ejemplo, consideremos el siguiente sistema:

$$\begin{aligned} 0,03x_1 + 58,9x_2 &= 59,2 \\ 5,31x_1 - 6,10x_2 &= 47,0; \end{aligned}$$

que debe ser resuelto con una precisión de solamente tres dígitos y aplicando corte en vez de redondeo. Si aplicamos Eliminación de Gauss tendremos:

$$\left[\begin{array}{cc|c} 0,03 & 58,9 & 59,2 \\ 0 & -10400 & -10300 \end{array}\right];$$

pues al hacer los cálculos obtenemos que:

$$m_{21} = \frac{5,31}{0,03} = 177 \Rightarrow a_{22}^* = -6,10 - 177 \times 58,9 \approx -6,10 - 10400 \approx -10400$$

$$b_2^* = 47,0 - 177 \times 59,2 \approx 47,0 - 10400 \approx -10300.$$

Así, la solución del sistema es:

$$\left\{\begin{array}{c} x_1 \\ x_2 \end{array}\right\} = \left[\begin{array}{c} 30,0 \\ 0,990 \end{array}\right],$$

Pero si resolvemos el sistema anterior con precisión «infinita», el resultado que obtenemos es:

$$\left\{\begin{array}{c} x_1 \\ x_2 \end{array}\right\} = \left[\begin{array}{c} 10 \\ 1 \end{array}\right],$$

lo que nos indica que el resultado anterior es incorrecto. Esta diferencia está dada por el coeficiente $0,03$ en la diagonal principal. Si reordenamos el sistema original tenemos:

$$\begin{aligned} 5,31x_1 - 6,10x_2 &= 47,0 \\ 0,03x_1 + 58,9x_2 &= 59,2; \end{aligned}$$

y si utilizamos la misma precisión, resulta:

$$\left[\begin{array}{cc|c} 5,31 & -6,10 & 47,0 \\ 0 & 58,9 & 58,9 \end{array}\right];$$

puesto que al hacer los cálculos obtenemos:

$$m_{21} = \frac{0,03}{5,31} = 0,005649 \approx 0,005 \Rightarrow a_{22}^* = 58,9 - 0,005 \times (-6,10) \approx 58,9 + 0,030 \approx 58,9$$

$$b_2^* = 59,2 - 0.005 \times 47,0 = 59,2 - 0,235 \approx 58,9.$$

La solución del sistema es:

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} 10 \\ 1 \end{bmatrix},$$

resultado que coincide con el obtenido con precisión «infinita».

Es por eso que el método de Eliminación de Gauss con Pivoteo Parcial (EGPP) se usa también cuando alguno de los coeficientes de la diagonal principal es muy chico con respecto a los demás coeficientes de la matriz.

En el caso del pivoteo total se efectúa no sólo un reordenamiento de las filas sino también de las columnas, lo que complica aún más el procedimiento.

Ambos casos insumen un mayor costo computacional que resulta muy difícil estimar puesto que no se trata de contar operaciones aritméticas como en la estimación anterior, si bien se considera que una comparación es equivalente a una suma/resta.

2.5. Factorización LU

El método de eliminación de Gauss es un método muy potente. Sin embargo, no siempre es conveniente su utilización. Supongamos por un momento que para resolver un determinado problema debemos resolver el sistema de ecuaciones en forma anidada. Es decir, cada nueva solución depende del resultado obtenido en un paso anterior, o sea, cada vector B depende de la solución anterior ($B^{<i>} = f\left(x^{<i-1>}\right)$).

Si queremos resolver estos sistemas nos encontraremos con la desventaja de que en cada paso tendremos que recalcular la matriz triangular superior, lo que significa un costo computacional muy grande, tal como vimos en el punto anterior. Por lo tanto, deberíamos buscar un método que nos evite repetir dichos cálculos.

Un método muy eficiente para estos casos es la *descomposición o factorización LU*. Ésta consiste en descomponer la matriz A original en el producto de dos matriz: una triangular inferior (L) y una triangular superior (U), para armar el siguiente sistema:

$$Ax = LUx = B \text{ con } A = LU.$$

De esta forma obtenemos dos sistemas de ecuaciones:

$$\begin{aligned} Ly &= B \\ Ux &= y \end{aligned}$$

En el primer caso, para obtener la solución intermedia y, aplicamos la sustitución directa, y en el segundo, la sustitución inversa. Vemos que en este método el vector B no es transformado en ninguno de los sistemas resueltos, que es lo que estábamos buscando. ¿Pero cómo se obtienen las dos matrices triangulares?

En el caso de la matriz triangular superior, la forma más sencilla de obtenerla es aplicar el mismo algoritmo que el utilizado para eliminación de Gauss, lo que significa que el costo computacional es similar (pero no igual, puesto que no debe transformarse al vector B). Nos falta la matriz L. Pero esta matriz es muy sencilla de obtener. Planteemos el esquema para obtener los coeficientes de la matriz L partiendo que los elementos de la diagonal principal son

iguales a 1 ($l_{ii} = 1$):

$$\begin{aligned}
u_{11} &= a_{11} \\
l_{21}u_{11} &= a_{21} \Rightarrow l_{21} = \frac{a_{21}}{u_{11}} = \frac{a_{21}}{a_{11}} = m_{21} \\
l_{31}u_{11} &= a_{31} \Rightarrow l_{31} = \frac{a_{31}}{u_{11}} = \frac{a_{31}}{a_{11}} = m_{31} \\
\cdots \quad &\cdots \quad \cdots \\
l_{31}u_{12} + l_{32}u_{22} &= a_{32} \Rightarrow l_{32}u_{22} = a_{32} - l_{31}u_{12} = \underbrace{a_{32} - m_{31}a_{12}}_{a^*_{32}} = a^*_{32} \\
l_{32} &= \frac{a^*_{32}}{u_{22}} = \frac{a^*_{32}}{a^*_{22}} = m_{32}
\end{aligned}$$

Como vemos, la matriz L está compuesta por los coeficientes de la diagonal principal iguales a 1 ($l_{ii} = 1$), en tanto que los coeficientes por debajo de la diagonal principal iguales a los coeficientes m_{ij} del método de eliminación de Gauss ($l_{ij} = m_{ij}$). Es decir, las matrices tienen la siguiente forma:

$$L = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ m_{21} & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ m_{n-1,1} & \cdots & m_{n-1,n-2} & 1 & 0 \\ m_{n1} & \dots & m_{n,n-2} & m_{n,n-1} & 1 \end{bmatrix}$$

y

$$U = \begin{bmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a^*_{22} & a^*_{23} & \dots & a^*_{2n} \\ 0 & 0 & a^*_{33} & \cdots & a^*_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & a^*_{nn} \end{bmatrix}$$

donde los a^*_{ij} son los coeficientes transformados del método de Eliminación de Gauss.

Obtenidas L y U, la solución del sistema la obtenemos aplicando, primero, la sustitución directa para hallar el vector y y luego, sustitución inversa para hallar x. Para el primer caso, aplicamos el siguiente algoritmo:

$$\begin{aligned}
y_1 &= b_1 \\
y_2 &= b_2 - l_{21}y_1 \\
&\vdots \\
y_i &= b_i - \sum_{j=1}^{i-1} l_{ij}y_j
\end{aligned}$$

puesto que los coeficientes l_{ii} son iguales a uno ($l_{ii} = 1$).

Como dijimos, obtenido y, se aplica la sustitución inversa para obtener el vector x solución del sistema. El algoritmo es:

$$\begin{aligned}
x_n &= \frac{y_n}{u_{nn}} \\
x_{n-1} &= \frac{y_{n-1} - u_{n-1,n}\, y_n}{u_{n-1,n-1}} \\
&\vdots \\
x_i &= \frac{y_i - \sum_{j=i+1}^{n} u_{ij}x_j}{u_{ii}}
\end{aligned}$$

Como vemos, en ningún caso hemos modificado o transformado al vector B, por lo que una vez que obtenemos las matrices U y L, podemos resolver los distintos sistemas aplicando sustitución directa primero e inversa después. Este método se conoce como *Método de Doolittle*.

Ahora nos quedaría analizar el costo computacional del método. Sin embargo, dado que hemos utilizado el método de eliminación de Gauss para obtener las matrices U y L, el costo para este método es muy similar al de dicho método. En consecuencia, la ventaja está principalmente en no tener que repetir la triangulación de la matriz A para cada sistema con un B distinto.

Al obtener la matriz U mediante Eliminación de Gauss podemos tener el mismo problema ya visto: que un coeficiente de la diagonal principal se haga nulo en los pasos intermedios. En ese sentido, valen las mismas aclaraciones respecto al Pivoteo Parcial y al Pivoteo Total. Es por eso que suele decirse que existe un par de matrices L y U que cumplen con:

$$PA = LU,$$

donde P es una *matriz de permutación*.

2.6. Método de Cholesky

2.6.1. Matrices simétricas y definidas positivas

Antes de analizar un caso particular de factorización de matrices conviene recordar la definición de un algunos tipos de matrices. En primer lugar, se dice que una matriz es simétrica cuando dicha matriz es igual a su transpuesta, es decir:

$$A = A^T.$$

Otro tipo de matriz es la conocida como definida positiva[1]. En este caso se debe cumplir que:

$$x^T Ax > 0 \text{ para todo } x \neq 0.$$

Es de notar que lo que se impone para que una matriz sea definida positiva es que el escalar resultante de la operación $x^T Ax$ sea no nulo y mayor que cero. En general demostrar esto resulta muy engorroso, por lo que suelen utilizarse algunos procedimientos alternativos. Para ello veamos los siguiente conceptos.

Definición 2.8. Una *primera submatriz principal* de una matriz A es la que tiene la forma:

$$A_k = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kn} \end{bmatrix}$$

para alguna $1 \leq k \leq n$.

Teorema 2.1. Una matriz simétrica A es *definida positiva* si y sólo si sus primeras submatrices principales tienen determinante positivo.

Teorema 2.2. La matriz simétrica A es *definida positiva* si y sólo si la eliminación de Gauss sin pivoteo puede efectuarse en el sistema $Ax = B$ con todos los pivotes positivos.

Corolario 2.2.1. La matriz simétrica A es *definida positiva* si y sólo si A puede factorizarse en la forma LDL^T, donde L es una matriz triangular inferior con coeficientes iguales a uno en la diagonal principal ($l_{ii} = 1$) y D es una matriz diagonal con coeficientes positivos ($d_{ii} > 0$).

Corolario 2.2.2. La matriz A es *simétrica y definida positiva* si y sólo si A puede factorizarse en la forma LL^T donde L es una matriz triangular inferior con elementos no nulos en su diagonal.

[1] Algunos autores exigen que A sea simétrica y definida positiva. Sin embargo, en principio, se puede decir que no es necesario que una matriz sea simétrica para que sea definida positiva.

2.6.2. Algoritmo de Cholesky

Con el último corolario se puede efectuar una factorización de la matriz A conocida como método o algoritmo de **Cholesky**. En efecto, si la matriz A es simétrica definida positiva, es posible obtener una matriz S que cumpla:

$$SS^T = A.$$

Veamos como podemos obtener esta natriz a partir de la factoriziación LU. De acuerdo con el corolario 2.2.1, la matriz simétrica A puede ser factorizada como LDL^T. Si además es definida positiva, entonces los coeficientes de D son positivos. En consecuencia, podemos obtener sin problemas $\sqrt{D}$, con lo cual tenemos $A = L\sqrt{D}\sqrt{D}L^T$. Así nuestra matriz A puede ser expresada como:

$$A = \underbrace{L\sqrt{D}}_{S} \cdot \underbrace{\sqrt{D}L^T}_{S^T} = SS^T.$$

Finalmente, las expresiones para obtener esta matriz S son:

$$s_{ii} = \left[a_{ii} - \sum_{k=1}^{i-1} s_{ik}^2 \right]^{1/2} \text{ y } s_{ji} = \frac{1}{s_{ii}} \left[a_{ji} - \sum_{k=1}^{i-1} s_{jk} s_{ik} \right],$$

con $j > i$.

Este método es mucho más eficiente puesto que sólo debemos calcular y guardar una sola matriz, a diferencia de la factorización LU en la que debo calcular y guardar dos matrices, si bien algunos algortimos permiten guardar ambas matrices en una sola. Además, el método Cholesky no aumenta considerablemente el «costo computacional» que analizamos en los puntos anteriores, por más que deban extraerse n raíces cuadradas.

Este método es muy aplicado en programas estructurales que aplican el método de los elementos finitos, dado que la matriz de coeficientes es una matriz simétrica y definida positiva. De todos modos, tiene las mismas desventajas vistas para los otros métodos cuando la dimensión de la matriz es cada vez más grande.

2.7. Condición de una matriz

Uno de los puntos a tener en cuenta es qué error cometemos al resolver un sistema de ecuaciones lineales mediante un método directo. Una forma de conocer el error de nuestro vector solución x sería analizar el algoritmo utilizado con ayuda de la gráfica de proceso. Este procedimiento resulta un tanto engorroso y largo, además de poco práctico. Una segunda manera podría ser analizar lo siguiente: puesto que nuestro sistema se puede expresar como $Ax = B$, una forma alternativa es $x = A^{-1}B$. Si definimos a $N = A^{-1}$, nos queda $x = NB$. Si desarrollamos esta expresión para cada componente de x nos queda:

$$x_i = \sum_{j=1}^{n} n_{ij} b_j.$$

Armemos un algoritmo que tenga la siguiente forma:

$$\begin{aligned} s_j &= n_{ij} b_j; \\ x_i &= \sum_{j=1}^{n} s_j. \end{aligned}$$

Analicemos los errores en cada paso. Para el primero tenemos:

$$\begin{aligned} e_{s_j} &= b_j e_{n_{ij}} + n_{ij} e_{b_j} = b_j e_{ij} + n_{ij} e_j \\ er_{s_j} &= \frac{b_j e_{n_{ij}} + n_{ij} e_{b_j}}{n_{ij} b_j} + \mu_j = er_{n_{ij}} + er_{b_j} + \mu_j = er_{ij} + er_j + \mu_j. \end{aligned}$$

Con el error relativo podemos recalcular el error total de s_j:

$$e_{s_j} = b_j e_{ij} + n_{ij} e_j + n_{ij} b_j \mu_j.$$

Para el segundo paso tendremos:

$$\begin{aligned} e_{x_i} &= \sum_{j=1}^{n} e_{s_j} = \sum_{j=1}^{n} (b_j e_{ij} + n_{ij} e_j) + \sum_{j=1}^{n} n_{ij} b_j \mu_j, \\ er_{x_i} &= \sum_{j=1}^{n} \frac{e_{s_j}}{x_i} + \sum_{k=2}^{n} \mu_k = \sum_{j=1}^{n} \frac{(b_j e_{ij} + n_{ij} e_j)}{x_i} + \sum_{j=1}^{n} \frac{n_{ij} b_j \mu_j}{x_i} + \sum_{k=2}^{n} \mu_k. \end{aligned}$$

Esta última expresión la podemos escribir también como:

$$er_{x_i} = \sum_{j=1}^{n} \frac{n_{ij} b_j \left(er_{ij} + er_j + \mu_j\right)}{x_i} + \sum_{k=2}^{n} \mu_k.$$

Si reordenamos los términos tendremos:

$$er_{x_i} = \sum_{j=1}^{n} \frac{n_{ij} b_j}{x_i} \left(er_{ij} + er_j\right) + \sum_{j=1}^{n} \frac{n_{ij} b_j}{x_i} \mu_j + \sum_{k=2}^{n} \mu_k.$$

De esta forma se puede decir que el error relativo de x_i es:

$$er_{x_i} \approx \sum_{j=1}^{n} Cp_j \left(er_{ij} + er_j\right) + \sum_{j=1}^{n} Te_j \mu_j,$$

con

$$Cp_j = \frac{2 n_{ij} b_j}{x_i} \text{ y } Te_j \approx \frac{n_{ij} b_j}{x_i} + 1,$$

si tomamos que $er_{ij}; er_j < r$ y que $\mu_j < \varepsilon$.

Hemos encontrado para cada x_i la expresión del error relativo, o mejor dicho, una idea aproximada del error. Pero, en la práctica, ¿sirve esto? Todos los cálculos son engorrosos y además hemos partido de un algoritmo no del todo práctico, pues hemos dicho que invertir la matriz no es conveniente [2]. Entonces, ¿qué hacemos?

Supongamos que hemos resuelto nuestro sistema $Ax = B$ con un algoritmo cualquiera y que en consecuencia hemos obtenido una solución $\hat{x}$. Lo que nos interesa conocer es una cota del error absoluto, $\|x - \hat{x}\|$, o del error relativo, $\dfrac{\|x - \hat{x}\|}{\|x\|}$, en alguna norma, por ejemplo, la norma infinito.

Como, en principio, no conocemos el resultado exacto de x, lo que podemos hacer es calcular lo siguiente:

$$R = B - A\hat{x},$$

donde R lo denominamos *residuo*. Si nuestra solución $\hat{x}$ fuera la solución exacta, entonces nuestro vector R debería ser nulo. Sin embargo, en la práctica, siempre obtendremos un vector R no nulo,

[2] Esta deducción es interesante, pues nos muestra que tanto el C_p como el T_e dependen de la matriz A, dado que la matriz N no es otra cosa que A^{-1}. De ahí que la solución de cualquier sistema de ecuaciones lineales mediante la inversión de la matriz es potencialmente inestable.

debido a la propagación de los errores de redondeo o de los errores inherentes y de redondeo. ¿Qué conclusiones podemos sacar conociendo R? Veamos el siguiente ejemplo.

Supongamos la matriz A y el vector B dados a continuación:

$$A = \begin{bmatrix} 1,2969 & 0,8648 \\ 0,2161 & 0,1441 \end{bmatrix}; B = \begin{bmatrix} 0,8642 \\ 0,1440 \end{bmatrix}.$$

Supongamos también que usando un determinado algoritmo hemos obtenido las siguientes soluciones:

$$\hat{x}_1 = \begin{bmatrix} 0,9911 \\ -0,4870 \end{bmatrix};\ \hat{x}_2 = \begin{bmatrix} -0,0126 \\ 1,0182 \end{bmatrix}.$$

Entonces, tendremos:

$$R = B - A\hat{x}_i \approx \begin{bmatrix} -10^{-7} \\ 10^{-7} \end{bmatrix}.$$

Por lo tanto, tendremos que $\|R\|_\infty = 10^{-7}$. Podemos decir que el residuo es muy chico. Sin embargo, la solución correcta es:

$$x = \begin{bmatrix} 2 \\ -2 \end{bmatrix}.$$

Es decir, ¡el error cometido es del mismo orden de la solución, o sea, 10^7 veces el residuo!

Es importante tener en cuenta que cualquiera sea el algoritmo utilizado, no podemos esperar sino un residuo pequeño o muy pequeño, lo que significa que este residuo R por sí solo no nos sirve de mucho para estimar el error que hemos cometido al obtener $\hat{x}$.

¿Cómo se relaciona, entonces, este residuo con el error en $\hat{x}$? Veamos. Escribamos el residuo como:

$$R = B - A\hat{x} = Ax - A\hat{x} = A\left(x - \hat{x}\right).$$

es decir:

$$x - \hat{x} = A^{-1}R.$$

Elijamos cualquier norma vectorial, por ejemplo, la infinita. Entonces tendremos:

$$\|x - \hat{x}\|_\infty = \left\|A^{-1}R\right\|_\infty \leq \left\|A^{-1}\right\|_\infty \|R\|_\infty.$$

Esto nos da una cota del error absoluto en términos de A^{-1}. Usualmente el error relativo es más significativo que el absoluto. Como $\|B\| \leq \|A\|\,\|x\|$ implica que $\frac{1}{\|x\|} \leq \frac{\|A\|}{\|B\|}$, tendremos que:

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \left\|A^{-1}\right\| \|R\| \frac{\|A\|}{\|B\|} = \left\|A^{-1}\right\| \|A\| \frac{\|R\|}{\|B\|}.$$

Esta expresión nos permite establecer que el residuo por sí mismo no nos alcanza para estimar el error de nuestro vector solución $\hat{x}$, sino que también debemos conocer algunas características de la matriz A. En particular, vemos que el error relativo de $\hat{x}$ depende de $\left\|A^{-1}\right\| \|A\|$. A este número lo denominaremos *condición de A* y lo expresaremos como $\text{cond}_\infty(A)$ o $\kappa(A)$[3]. Tanto $\left\|A^{-1}\right\|$ como $\|A\|$ son números reales (son normas de las matrices) por lo tanto para que el error relativo de $\hat{x}$ no sea muy grande, el producto de $\left\|A^{-1}\right\| \|A\|$ debiera ser cercano a uno, es decir:

$$\left\|A^{-1}\right\| \|A\| = \kappa(A) > / > 1.$$

Si la matriz es no singular debe cumplirse que:

$$1 = \|I\| = \left\|A^{-1}A\right\| \leq \kappa(A),$$

que puede considerarse el límite inferior, en tanto que si la matriz A es singular (no existe A^{-1}), $\kappa(A) = \infty$, que puede ser considerado el límite superior. Así, puede decirse que el número de

[3]En este caso hemos utilizado la norma infinito. Podría haberse usado la norma euclídea y obtener el $\text{cond}_2(A)$.

condición da una idea de cuan cerca está la matriz de ser *singular*, o lo que es lo mismo, de que el sistema no tenga solución o que sean infinitas.

Una conclusión interesante es que si la matriz A del sistema está mal condicionada, pequeños desvíos en el residuo R pueden llevar a grandes desvíos en $\hat{x}$, es decir, si definimos que $\|\Delta x\| = \|x - \hat{x}\|$, entonces puede darse que $\|\Delta x\| >> 1$, algo que no es aceptable.

2.8. Refinamiento iterativo de la solución

Hemos visto en los puntos anteriores que los métodos directos pueden resolver muy bien un sistema de ecuaciones lineales, con excepción de un sistema con la matriz de coeficientes A mal condicionada. Aún así, existe la posibilidad de obtener una solución aceptable, dentro de cierto rango. Al analizar el error cometido, introdujimos el concepto del vector «residuo», que denominamos R, y que obtuvimos de la siguiente manera:

$$R = B - A\hat{x}.$$

Como vimos, con ese vector residuo podemos calcular el error de nuestra aproximación $\hat{x}$ respecto de nuestra solución «exacta» x, pues tenemos que:

$$B - A\hat{x} = Ax - A\hat{x} = A\underbrace{(x - \hat{x})}_{\delta} = A\delta = R,$$

y, en consecuencia, resolviendo este nuevo sistema de ecuaciones podemos obtener nuestro valor δ. Dado que hemos definido que $\delta = x - \hat{x}$, entonces podemos decir que:

$$x = \hat{x} + \delta,$$

y con ello hemos obtenido nuestra solución «exacta». Sin embargo, esto no suele ocurrir al primer intento, de manera que lo que obtendremos en realidad es una nueva aproximación de nuestra solución, que llamaremos $\tilde{x}$. Para sistematizar esto, digamos que

$$\hat{x} = x_1;\ R_1 = B - Ax_1;\ A\delta_1 = R_1,$$

por lo que tendremos:

$$\tilde{x} = x_2 = x_1 + \delta_1.$$

El paso siguiente es obtener R_2 y δ_2, en forma análoga a δ_1. En consecuencia, tendremos que

$$x_3 = x_2 + \delta_2 = x_1 + \delta_1 + \delta_2 = \hat{x} + \delta_1 + \delta_2.$$

Si generalizamos, tenemos que la solución «exacta» se puede obtener con la expresión

$$x = \hat{x} + \sum_{i=1}^{n\to\infty} \delta_i,$$

es decir, que a la solución aproximada le sumamos todos los «errores» para obtener la solución «exacta». Por supuesto, es imposible efectuar infinitas interaciones, por lo que es imprescindible establecer algún criterio de corte. Un criterio puede ser cortar las iteraciones cuando $\|R_k\| \leq Tol$, pero vimos que esto no asegura que el error sea pequeño. Otro crietrio, tal vez más acertado, es interrumpir las iteraciones o cálculos cuando $\|\delta_k\| \leq Tol$, que tiene en cuenta el error de $\hat{x}$.

Este procedimiento que obtiene la solución de nuestro sistema sumando los errores, se conoce como *método del refinamiento iterativo de la solución* y ha cobrado gran desarrollo en los últimos años, pues pueden obtenerse buenos resultados con matrices mal condicionadas. Suele decirse que para obtener una buena solución, los sistemas $A\delta_i = R_i$ deben resolverse con mayor precisión que el sistema original. Si hemos resuelto el sistema $Ax = B$ en simple precisión,

entonces debe usarse doble precisión para resolver cada uno de estos sistemas. Esto no es del todo cierto, ya que pueden obtenerse buenos resultados usando la misma precisión, tal como ha demostrado N. Higham (véase [6]). Pero existe otra cuestión. ¿Cuando conviene aplicar este método?

Supongamos (una vez más) que obtenemos la aproximación $\hat{x}$. Con esta solución, podemos obtener el vector residuo mediante

$$R_1 = B - A\hat{x}.$$

Si realizamos los cálculos utilizando una precisión de t dígitos, podemos demostrar que

$$\|R_1\| \approx 10^{-t} \|A\| \|\hat{x}\|.$$

Para saber si el método es convergente, podemos obtener una aproximación o estimación del *número de condición* de A. Para ello vamos a obtener el vector δ_1 según vimos arriba, es decir, haciendo

$$A\delta_1 = R_1.$$

Entonces, podemos escribir lo siguiente:

$$\|\delta_1\| \approx \|x - \hat{x}\| = \left\|A^{-1}R_1\right\| \leq \left\|A^{-1}\right\| \|R_1\| \approx \left\|A^{-1}\right\| \left(10^{-t} \|A\| \|\hat{x}\|\right) = 10^{-t} \|\hat{x}\| \kappa(A),$$

con lo cual podemos estimar $\kappa(A)$ mediante

$$\kappa(A) \approx \frac{\|\delta_1\|}{\|\hat{x}\|} 10^t.$$

Como hemos dicho, este método permite obtener buenos resultados inclusive con matrices mal condiciondas. Sin embargo, si $\kappa(A) >> 10^t$, el sistema está tan mal condicionado que debe modificarse la precisión original usada en la obtención de $\hat{x}$ para obtener un resultado aproximado aceptable.

2.9. Errores de los métodos directos

Hemos visto que el hecho de obtener un vector residuo pequeño no es garantía para inferir que el resultado obtenido tiene un error también pequeño. Analicemos el sistema en una forma más detallada. Supongamos ahora que tanto la matriz A como el vector B tienen pequeñas perturbaciones que llamaremos δA y δB respectivamente, y que nuestra solución sea $\hat{x}$. Entonces tendremos:

$$(A + \delta A)\hat{x} = B + \delta B.$$

Podemos escribir que

$$A\hat{x} + \delta A\hat{x} = B + \delta B.$$

Sabemos que $x = \hat{x} + \delta x$, por lo tanto podemos escribir:

$$\begin{aligned} A(x - \delta x) + \delta A(x - \delta x) &= B + \delta B, \\ Ax - A\delta x + \delta Ax - \delta A\delta x &= B + \delta B. \end{aligned}$$

Si despreciamos $\delta A\delta x$, tendremos

$$\begin{aligned} Ax + A\delta x - \delta Ax &= B + \delta B, \\ A\delta x - \delta Ax &= \delta B, \\ A\delta x &= \delta B + \delta Ax, \\ \delta x &= A^{-1}\delta B + A^{-1}\delta Ax. \end{aligned}$$

Si tomamos normas a ambos lados tendremos:

$$\|\delta x\| \leq \left\|A^{-1}\right\| \|\delta B\| + \left\|A^{-1}\right\| \|\delta A\| \|x\|,$$

y como además tenemos que $\|B\| \leq \|A\| \|x\|$, entonces podemos dividir todo de manera de obtener:

$$\begin{aligned} \frac{\|\delta x\|}{\|A\| \|x\|} &\leq \frac{\left\|A^{-1}\right\| \|\delta B\|}{\|B\|} + \frac{\left\|A^{-1}\right\| \|\delta A\| \|x\|}{\|A\| \|x\|}, \\ &\leq \frac{\left\|A^{-1}\right\| \|\delta B\|}{\|B\|} + \frac{\left\|A^{-1}\right\| \|\delta A\|}{\|A\|}. \end{aligned}$$

Si multiplicamos por $\|A\|$ tendremos que:

$$\begin{aligned} \frac{\|\delta x\|}{\|x\|} &\leq \frac{\left\|A^{-1}\right\| \|A\|}{\|B\|} \|\delta B\| + \left\|A^{-1}\right\| \|\delta A\| \frac{\|x\| \|A\|}{\|B\|} \\ \frac{\|\delta x\|}{\|x\|} &\leq \frac{\left\|A^{-1}\right\| \|A\|}{\|B\|} \|\delta B\| + \left\|A^{-1}\right\| \|\delta A\| \\ \frac{\|\delta x\|}{\|x\|} &\leq \underbrace{\|A\| \left\|A^{-1}\right\|}_{\text{cond}(A)} \left(\frac{\|\delta B\|}{\|B\|} + \frac{\|\delta A\|}{\|A\|} \right). \end{aligned}$$

Podemos ver que para que los errores de x sean pequeños no basta con que δB y δA sean pequeños (es decir, que los errores inherentes sean pequeños), sino que es necesario que el número de condición de A (cond(A)) sea cercano a 1.

Analicemos ahora los errores de redondeo. Vamos a buscar una cota de estos errores. Supongamos que aplicamos el método de factorización LU para resolver el sistema. Si suponemos que solamente se producen errores de redondeo, entonces tendremos en realidad que

$$LU = A + \delta A,$$

donde δA son las perturbaciones producidas por los errores de redondeo al obtener L y U. Entonces, nuestro sistema queda como:

$$\begin{aligned} (A + \delta A)(x - \delta x) &= B \\ Ax - A\delta x + \delta A x - \delta A \delta x &= B \\ A\delta x &= -\delta A(x - \delta x). \end{aligned}$$

por lo tanto,

$$\delta x = -A^{-1}\delta A(x - \delta x).$$

Si tomamos la norma tenemos

$$\begin{aligned} \|\delta x\| &\leq \left\|A^{-1}\right\| \|\delta A\| \|x - \delta x\| \\ &\leq \left\|A^{-1}\right\| \|A\| \|x - \delta x\| \frac{\|\delta A\|}{\|A\|} \\ \|\delta x\| &\leq \kappa(A) \|x - \delta x\| \frac{\|\delta A\|}{\|A\|}. \end{aligned}$$

Se puede demostrar que $\|\delta A\| \leq 1,01(n^3 + 3n^2)\rho \|A\| \mu$, donde $\rho = \text{máx} \dfrac{\left|a_{ij}^k\right|}{\|A\|}$, n es la dimensión de la matriz A y μ es la unidad de máquina; entonces tenemos que

$$\frac{\|\delta x\|}{\|x - \delta x\|} \approx \frac{\|\delta x\|}{\|x\|} \leq \kappa(A) 1,01(n^3 + 3n^2)\rho\mu,$$

y podemos definir el error total para los métodos directos como

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \left[\frac{\|\delta B\|}{\|B\|} + \frac{\|\delta A\|}{\|A\|} + 1,01(n^3 + 3n^2)\rho\,\mu \right].$$

Podemos ver que si la matriz es de grandes dimensiones, comienzan a tener gran incidencia los errores de redondeo, con lo cual el sistema puede volverse inestable si $\kappa(A) >> 1$, además de mal condicionado.

2.10. Métodos iterativos

Hasta ahora hemos estudiado los llamados *métodos directos* para resolver sistemas de ecuaciones lineales. Son llamados de esta forma porque el algoritmo tiene una cantidad conocida de pasos («finita») y los resultados que obtenemos al aplicarlos deberían ser exactos, salvo por el error de redondeo, aunque vimos que esto no siempre es así. Estos métodos se suelen usar con matrices densas o casi llenas, como por ejemplo las surgidas del análisis matricial de estructuras planas, las cuales tienen muchos coeficientes distintos de cero ($a_{ij} \neq 0$).

Pero existen muchos otros problemas en los cuales el sistema de ecuaciones tiene una matriz A que no es densa, sino por el contrario, es *rala*, es decir, tiene muchos coeficientes nulos, como es el caso del análisis estructural en tres dimensiones. Entonces trabajar con los métodos directos se vuelve muy poco práctico, pues debemos hacer muchas operaciones con coeficientes nulos y, lo que es peor, muchas veces transformar un coeficiente nulo en otro no nulo, incorporando un error que antes no existía. Es por eso que se han desarrollado métodos que tienen en cuenta este tipo de matrices. Son los métodos denominados *iterativos*.

En estos métodos, la solución la obtenemos a partir de una solución inicial, la cual se va *corrigiendo* en sucesivas iteraciones hasta obtener la solución «correcta», de ahí el nombre de iterativos. En principio, podenos suponer que la cantidad de iteraciones es «infinita», es decir, que la solución exacta la obtenemos luego de infinitas iteraciones. Como efectuar esto es imposible, lo que se hace es iterar hasta que la solución esté dentro de las tolerancias impuestas.

Para analizar estos métodos partamos de definirlos en forma matricial. Sabemos que nuestro sistema se expresa como

$$Ax = B,$$

o, lo que es lo mismo, como

$$B - Ax = 0.$$

En consecuencia, podemos sumar en ambos miembros Px sin cambiar la igualdad. Nos queda que

$$Px = Px - Ax + B \Rightarrow Px = (P - A)\,x + B.$$

Si despejamos x de la expresión anterior, nos queda:

$$x = P^{-1}\,(P - A)\,x + P^{-1}B,$$

que puede escribirse como

$$x = \left(P^{-1}P - P^{-1}A\right)x + P^{-1}B = \left(I - P^{-1}A\right)x + P^{-1}B,$$

a partir del cual se puede obtener el método iterativo para resolver un sistema de ecuaciones, que toma la siguiente forma:

$$x^{(n+1)} = \left(I - P^{-1}A\right)x^{(n)} + P^{-1}B,$$

donde n es la iteración.

La expresión anterior puede escribirse en forma general como

$$x^{(n+1)} = Tx^{(n)} + C,$$

donde

$$T = I - P^{-1}A \text{ y } C = P^{-1}B.$$

Con esta última expresión podemos defnir dos tipos de métodos iterativos: los estacionarios, aquellos en los que T y C no sufren modificaciones durante las iteraciones, y los no estacionarios, aquellos en los que los valores de T y C dependen de la iteración.

2.10.1. Métodos estacionarios

Como hemos visto, los métodos iterativos estacionarios son aquellos en los que T y C son *invariantes*, es decir, permanecen constantes en las sucesivas iteraciones necesarias para hallar la solución.

Supongamos por un momento que conocemos nuestra solución «exacta» x. Entonces podemos decir que:

$$\begin{aligned} x &= x^{(n+1)} + e^{(n+1)} \Rightarrow \\ x^{(n+1)} + e^{(n+1)} &= T\left(x^{(n)} + e^{(n)}\right) + C \\ &= \underbrace{Tx^{(n)} + C}_{x^{(n+1)}} + Te^{(n)} \\ &= x^{(n+1)} + Te^{(n)} \Rightarrow \\ e^{(n+1)} &= Te^{(n)}. \end{aligned}$$

De la última expresión podemos deducir que:

$$e^{(n+1)} = Te^{(n)} = TTe^{(n-1)} = T^2e^{(n-1)} = \ldots = T^{n+1}e^{(0)},$$

expresión que nos indica que para que un método iterativo estacionario sea convergente se debe cumplir que $\|T\| < 1$, y que $\|T\| << 1$ para que la convergencia sea rápida.

Método de Jacobi

El método estacionario más sencillo es el *Método de Jacobi*. Si definimos que $A = L + D + U$, este método es aquél que define $P = D$. Por lo tanto, podemos escribir que:

$$\begin{aligned} x^{(n+1)} &= \left(I - D^{-1}A\right)x^{(n)} + D^{-1}B \\ &= \left[I - D^{-1}(L + D + U)\right]x^{(n)} + D^{-1}B \\ &= [I - \underbrace{D^{-1}D}_{I} - D^{-1}(L + U)]x^{(n)} + D^{-1}B \\ &= D^{-1}\left[B - (L + U)x^{(n)}\right], \end{aligned}$$

donde L, D y U tienen la siguiente forma:

$$L = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ a_{21} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & \ldots & a_{m\,m-1} & 0 \end{bmatrix}, D = \begin{bmatrix} a_{11} & 0 & \ldots & 0 \\ 0 & a_{22} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & a_{mm} \end{bmatrix} \text{ y } U = \begin{bmatrix} 0 & a_{12} & \ldots & a_{1m} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & a_{m-1\,m} \\ 0 & \ldots & 0 & 0 \end{bmatrix}.$$

En su forma tradicional este método se expresa como:

$$x_i^{(n+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}.$$

En sí, el método consiste en suponer una solución inicial, generalmente el vector nulo ($x = [0]$), e iterar hasta obtener la solución, usando siempre el vector obtenido en el paso anterior. Para analizar la convergencia, debemos recordar algunas definiciones.

Definición 2.9. Una matriz A se denomina *diagonal dominante* si se cumple que

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\,.$$

Definición 2.10. Una matriz A se denomina *estrictamente diagonal dominante* si se cumple que

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

Definición 2.11. Una matriz A se denomina *diagonal dominante en forma irreductible* si se cumple que

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\,,$$

para $i = 1; 2; \ldots; n$ y en al menos una fila que

$$|a_{kk}| > \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}|\,.$$

El método de Jacobi converge rápidamente si la matriz A es *estrictamente diagonal dominante*, como se verá más adelante. En cambio, la convergencia es lenta si la matriz A es cualquiera de las otras dos. Finalmente, si la matriz A no cumple con ninguna de las definiciones anteriores, el método de Jacobi no converge.

Método de Gauss-Seidel

Cuando el método de Jacobi es convergente, esta convergencia es muy lenta. Para mejorar esta velocidad de convergencia, imaginemos que usamos parte de los resultados ya obtenidos en el obtención de los siguientes, es decir, obtener el x_i aprovechando los x_j para $j < i$. Este método se conoce como *método de Gauss-Seidel* y resulta de definir $P = D + L$. Desarrollemos la expresión final sabiendo que $Px^{(n+1)} = Px^{(n)} - Ax^{(n)} + B$:

$$\begin{aligned}
(D+L)x^{(n+1)} &= [(D+L) - A]\,x^{(n)} + B \\
&= [(D+L) - (L+D+U)]\,x^{(n)} + B \\
&= [D+L-L-D-U]\,x^{(n)} + B \\
&= B - Ux^{(n)} \Rightarrow \\
Dx^{(n+1)} &= B - Lx^{(n+1)} - Ux^{(n)} \Rightarrow \\
x^{(n+1)} &= D^{-1}\left[B - Lx^{(n+1)} - Ux^{(n)}\right].
\end{aligned}$$

En su forma tradicional el método se escribe de la siguiente manera:

$$x_i^{(n+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)}}{a_{ii}}.$$

Este método converge para las mismas condiciones impuestas al método de Jacobi. Se puede asegurar que si el método de Gauss-Seidel converge, también lo hace el de Jacobi, pero la inversa no siempre se cumple.

Método de las sobrerrelajaciones sucesivas (SOR)

Si bien Gauss-Seidel es más rápido que Jacobi, la velocidad de convergencia no es muy alta. Busquemos algún método que nos mejore esta velocidad. Partamos nuevamente de la expresión general $Px^{(n+1)} = Px^{(n)} - Ax^{(n)} + B$. Si reordenamos un poco la expresión tenemos:

$$Px^{(n+1)} = Px^{(n)} + \underbrace{B - Ax^{(n)}}_{R^{(n)}} = Px^{(n)} + R^{(n)},$$

que podemos escribir también como

$$x^{(n+1)} = \underbrace{P^{-1}P}_{I} x^{(n)} + P^{-1}R^{(n)} = x^{(n)} + P^{-1}R^{(n)}.$$

La idea es buscar una matriz P que nos mejore la velocidad de convergencia. Supongamos, entonces, que tomamos $P = L + \frac{1}{\omega}D$. Si partimos de la expresión conocida tenemos que:

$$\begin{aligned}
\left(\frac{1}{\omega}D + L\right) x^{(n+1)} &= \left[\left(\frac{1}{\omega}D + L\right) - A\right] x^{(n)} + B \\
&= \left[\left(\frac{1}{\omega}D + L\right) - (L + D + U)\right] x^{(n)} + B \\
&= \left[\frac{1}{\omega}D + L - L - D - U\right] x^{(n)} + B \\
&= B - \left(1 - \frac{1}{\omega}\right) Dx^{(n)} - Ux^{(n)} \Rightarrow \\
\frac{1}{\omega}Dx^{(n+1)} &= B - Lx^{(n+1)} - \left(1 - \frac{1}{\omega}\right) Dx^{(n)} - Ux^{(n)} \Rightarrow \\
x^{(n+1)} &= -\omega\left(1 - \frac{1}{\omega}\right) \underbrace{D^{-1}D}_{I} x^{(n)} + \omega D^{-1}\left[B - Lx^{(n+1)} - Ux^{(n)}\right] \\
&= (1-\omega)x^{(n)} + \omega D^{-1}\left[B - Lx^{(n+1)} - Ux^{(n)}\right] \\
&= (1-\omega)x^{(n)} + \omega\, x_{GS}^{(n+1)}.
\end{aligned}$$

Este método se conoce como *Método de las sobrerrelajaciones sucesivas* (o SOR por sus siglas en inglés), y pondera el $x^{(n)}$ con el $x^{(n+1)}$ obtenido con el método de Gauss-Seidel, tomando como factor de ponderación el coeficiente ω. En su forma tradicional se suele escribir como:

$$x_i^{(n+1)} = (1-\omega)x^{(n)} + \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(n)}}{a_{ii}}.$$

En este método la velocidad de convergencia está dada por el ω. Se puede asegurar que existe un valor que hace máxima la velocidad de convergencia para un sistema dado, que puede ser estimado conociendo el radio espectral de la matriz de Jacobi. Si observamos con detenimiento veremos que el método de Gauss-Seidel es un caso especial del SOR, pues surge de tomar $\omega = 1$. En efecto, si $\omega = 1$ tenemos:

$$\begin{aligned} x_i^{(n+1)} &= (1-1)x^{(n)} + \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}}{a_{ii}} \\ &= \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(n+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(n)}}{a_{ii}}, \end{aligned}$$

que es el método de Gauss-Seidel.

En realidad, al imponer que $0 < \omega < 2$ existen dos métodos: cuando $0 < \omega < 1$, estamos en presencia de un método de *subrelajación*, también conocido como *método de Jacobi modificado*, en tanto que cuando $1 < \omega < 2$, se trata de un método de *sobrerrelajación* propiamente dicho. En general, estos métodos convergen mucho más rápido que los otros dos, y puede decirse que cuando Gauss-Seidel no converge, utilizando un $\omega < 1$ se logra una mejor convergencia que con el método de Jacobi.

Criterios de interrupción

Hasta acá hemos visto los distintos métodos iterativos estacionarios más tradicionales que se aplican para resolver sistems de ecuaciones lineales. Pero no hemos analizado los criterios para interrumpir dichas iteraciones. Dado que los métodos convergen a una solución cuando $n \to \infty$, es decir, que se debe dar que $x - x^{(n)} = 0$ cuando $n \to \infty$, entonces podemos tomar como criterios para interrumpir las iteraciones, que $x - x^{(n)} < Tol$, siendo Tol una valor definido arbitrariamente, generalmente relacionado con la precisión utilizada (μ). Existen varios criterios que pueden aplicarse. Estos son:

1. Que la norma infinita del vector $r^{(n)}$ sea menor a la tolerancia, esto es:
$$\left\| r^{(n)} \right\|_\infty < Tol.$$

2. Que la norma infinita del error absoluto entre dos soluciones sucesivas de x sea menor a la tolerancia, es decir, que:
$$\left\| x^{(n)} - x^{(n-1)} \right\|_\infty < Tol.$$

3. Que la norma infinita del error relativo entre dos soluciones sucesivas sea menor a la tolerancia, o sea:
$$\frac{\left\| x^{(n)} - x^{(n-1)} \right\|_\infty}{\left\| x^{(n)} \right\|_\infty} < Tol.$$

El mejor de los criterios es último, pues hemos visto que es el error relativo el que mejor representa la incidencia del error en los resultados.

Sin embargo, debemos recordar del análisis de la cota de error para los métodos directos que

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \left\|A^{-1}\right\| \|A\| \frac{\|R\|}{\|B\|}$$

expresión que puede ampliarse al caso de un método iterativo como

$$\begin{aligned} \frac{\left\|x^{<k+1>} - x^{<k>}\right\|}{\|x^{<k+1>}\|} &\leq \left\|A^{-1}\right\| \|A\| \frac{\|R\|}{\|B\|} \\ \frac{\left\|x^{<k+1>} - x^{<k>}\right\|}{\|x^{<k+1>}\|} &\leq \kappa(A) \frac{\|R\|}{\|B\|} \approx \kappa(A)\, Tol. \end{aligned}$$

con lo cual debemos cuidarnos al momento de elegir la tolerancia cuando aplicamos un método iterativo. Queda evidente que cuando la matriz tiende a ser mal condicionada, la tolerancia debe ser más chica.

2.10.2. Convergencia de los métodos estacionarios

Hemos dicho que los métodos de Jacobi y Gauss-Seidel convergen para matrices A estrictamente diagonal dominantes. Los siguientes teoremas aseguran la convergencia de ambos métodos.

Teorema 2.3. Si A es una matriz de $n \times n$, entonces se cumple que:

1. $\|A\|_2 = \left[\rho\left(A^T A\right)\right]^{1/2}$.
2. $\rho(A) \leq \|A\|$, para toda norma natural.

Teorema 2.4. Si la matriz A es *estrictamente diagonal dominante*, entonces con cualquier elección de $x^{(0)}$, tanto el método de Jacobi como el de Gauss-Seidel dan las sucesiones $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ que convergen a una única solución del sistema $Ax = B$.

Teorema 2.5. Si $a_{ij} \leq 0$ para cada $i \neq j$, y si $a_{ii} > 0$ para cada $i = 1; 2; \ldots; n$, entonces será válida una y sólo una de las siguientes afirmaciones:

1. $0 \leq \rho(T_G) < \rho(T_J) < 1$;
2. $1 < \rho(T_J) < \rho(T_G)$;
3. $\rho(T_J) = \rho(T_G) = 0$;
4. $\rho(T_J) = \rho(T_G) = 1$;

donde T_J es la matriz de Jacobi, y T_G es la matriz de Gauss-Seidel.

Para analizar la convergencia del método de las sobrerrelajaciones sucesivas se deben tener en cuenta estos otros teoremas.

Teorema 2.6. Para cualquier $x^{(0)} \in \Re^n$, la sucesión $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ definida por

$$x^{(k+1)} = Tx^{(k)} + C, \text{ para cada } k \geq 1,$$

converge en la solución única de $x = Tx + C$ si y sólo si $\rho(T) < 1$.

Este teorema nos dice que cualquier método iterativo converge cuando el radio espectral de la matriz T es menor a 1, tal como vimos al comenzar. Recordemos que la definición del radio espectral de una matriz A cualquiera es

$$\rho(A) = max\,|\lambda|\,,$$

donde λ es un autovalor de A. En efecto, habíamos dicho que para que cualquier método iterativo sea convergente, se debía cumplir que $\|T\| < 1$. Como $\rho(T) \leq \|T\| < 1$, si los módulos de los autovalores de T son menores que 1, entonces los método convergen a la solución buscada.

Teorema 2.7. Si A es una matriz definida positiva y si $0 < \omega < 2$, entonces el método SOR converge para cualquier elección del vector aproximado $x^{(0)}$.

Este teorema lo podemos aplicar también al método de Gauss-Seidel. Efectivamente, puesto que cuando $\omega = 1$, el método SOR resulta ser el de Gauss-Seidel, y como el teorema 2.7 asegura la convergencia del método SOR para cualquier vector inicial cuando $0 < \omega < 2$, entonces asegura también la convergencia del método de Gauss-Seidel cuando la matriz A es definida positiva. Este teorema puede ampliarse a matrices simétricas definidas positivas.

Teorema 2.8. Si A es una matriz definida positiva y tridiagonal, entonces $\rho(T_G) = [\rho(T_J)]^2 < 1$, y la elección óptima de ω para el método SOR es

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_J)]^2}}.$$

Este último vincula los autovalores de la matriz T_J, es decir la matriz T del método de Jacobi, con el valor de ω. Aunque se refiere a una matriz tridiagonal, es posible ver que cuanto menor sea el valor de $\rho(T_J)$ más se acerca ω a 1. (Si $\rho(T_J)^2$ es mayor que uno, entonces no hay un ω real que haga convergente al método.)

2.10.3. Métodos no estacionarios

Vimos en el punto anterior los métodos estacionarios, aquellos en los cuales las matrices T y C se mantienen invariantes en las sucesivas iteraciones. Existen otros métodos en los cuales estas dos matrices sí se van modificando en las sucesivas iteraciones. Son los llamados *métodos no estacionarios*.

Supongamos que en nuestra expresión general, definimos que $P = \frac{1}{\alpha} I$. Si reemplazamos obtenemos:

$$\begin{aligned} x^{(i+1} &= (I - \alpha I A)\, x^{(i)} + \alpha I B, \\ &= x^{(i)} + \alpha \left(B - A x^{(i)} \right), \\ &= x^{(i)} + \alpha r^{(i)}. \end{aligned}$$

Tenemos ahora un método iterativo que depende de un parámetro α para ir corrigiendo el vector solución. Nos falta definir ese parámetro. Pero también depende de otro vector, el ya visto *residuo*. Por lo tanto tenemos dos elementos que podemos manejar para obtener una mejor aproximación. Veremos a continuación algunos de los métodos no estacioanrios más sencillos que han servido de base para el desarrollo de los más modernos y complejos.

Método de los residuos mínimos

Una primera aproximación para esta expresión es buscar que el vector $r^{(i+1)}$ sea mínimo en cada iteración. De esta manera siempre tenderemos a la solución del sistema, pues el ideal es que sea nulo. Una forma de obtener el mínimo es minimizar la norma euclídea, es decir, el módulo de $r^{(i+1)}$. Partamos precisamente de la definición del módulo:

$$\left\| r^{(i+1)} \right\|_2 = \left\| B - A x^{(i+1)} \right\|_2.$$

Si lo elevamos al cuadrado tenemos

$$\begin{aligned} \left\| r^{(i+1)} \right\|_2^2 &= \left\| B - A x^{(i+1)} \right\|_2^2 \\ \left\| r^{(i+1)} \right\|_2^2 &= \left\| B - A \left(x^{(i)} + \alpha r^{(i)} \right) \right\|_2^2 \\ r^{(i+1)^T} \cdot r^{(i+1)} &= \left[B - A \left(x^{(i)} + \alpha r^{(i)} \right) \right]^T \cdot \left[B - A \left(x^{(i)} + \alpha r^{(i)} \right) \right]. \end{aligned}$$

Como queremos minimizar el módulo de $r^{(i+1)}$, lo mismo es minimizar el cuadrado del módulo. Para ello vamos a derivar la útlima expresión respecto de α, que es nuestro parámetro, y lo igualaremos a cero. Así tenemos que:

$$\begin{aligned} -2\left(Ar^{(i)}\right)^T \cdot \left[B - Ax^{(i)} - \alpha Ar^{(i)}\right] &= 0 \\ \left(Ar^{(i)}\right)^T \cdot \left[r^{(i)} - \alpha Ar^{(i)}\right] &= 0 \\ \left(Ar^{(i)}\right)^T \cdot r^{(i)} &= \alpha \left(Ar^{(i)}\right)^T \cdot Ar^{(i)} \Rightarrow \\ \alpha_i &= \frac{\left(Ar^{(i)}\right)^T \cdot r^{(i)}}{\left(Ar^{(i)}\right)^T \cdot Ar^{(i)}}. \end{aligned}$$

Este coeficiente α_i nos asegura que el residuo sea mínimo. Así nuestro esquema iterativo queda de la siguiente forma:

$$\begin{aligned} r^{(i)} &= B - Ax^{(i)} \\ \alpha_i &= \frac{\left(Ar^{(i)}\right)^T \cdot r^{(i)}}{\left(Ar^{(i)}\right)^T \cdot Ar^{(i)}} \\ x^{(i+1)} &= x^{(i)} + \alpha_i \; r^{(i)}. \end{aligned}$$

Este método sólo es convergente si se cumple que la matriz A es simétrica y definida positiva, pues de lo contrario no obtendremos un mínimo. (Una demostración de esto puede verse en [10].)

Existe un segundo algoritmo que tiene la siguiente forma:

$$\begin{aligned} r^{(0)} &= B - Ax^{(0)} \\ \alpha_i &= \frac{\left(Ar^{(i)}\right)^T \cdot r^{(i)}}{\left(Ar^{(i)}\right)^T \cdot Ar^{(i)}} \\ x^{(i+1)} &= x^{(i)} + \alpha_i \; r^{(i)} \\ r^{(i+1)} &= r^{(i)} - \alpha_i \; Ar^{(i)}. \end{aligned}$$

En ambos algoritmos las iteraciones finalizan cuando $r^{(i+1)} < Tol$, pues $r^{(n)} = 0$ para $n \to \infty$.

Método del descenso más rápido

Un segundo método no estacionario es el denominado *método del descenso más rápido*. Este método mejora la aproximación obtenida en el punto anterior. Para poder deducirlo antes necesitamos saber qué es una forma cuadrática.

Forma cuadrática: Es una función vectorial que se expresa como:

$$f(x) = \frac{1}{2} \; x^T Ax - B^T x + C,$$

similar a una ecuación de segundo grado en el campo escalar, donde A es una matriz, x y B son vectores y C es una constante (escalar).

Supongamos ahora que queremos hallar el mínimo (o máximo) de esta función. Entonces debemos obtener su derivada e igualarla a cero, es decir, hacer que:

$$\frac{d \; f(x)}{dx} = \frac{1}{2} A^T x + \frac{1}{2} Ax - B = 0.$$

Si A es una matriz simétrica entonces $A = A^T$, y podemos escribir:

$$\frac{d\ f(x)}{dx} = Ax - B = 0,$$

que no es otra cosa que nuestro sistema de ecuaciones lineales original. Si además A es definida positiva, nos aseguramos que la solución que se obtenga haga mínima a la forma cuadrática. En consecuencia, para aplicar este método, la matriz A también debe ser *simétrica definida positiva*.

Recordemos también qué es el gradiente de una función vectorial. Para una función $f(x)$ el gradiente se expresa como:

$$\frac{d\ f(x)}{dx} = f'(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

El gradiente nos da una idea de la «pendiente» o del crecimiento de la forma cuadrática. Si queremos hallar el valor mínimo de la función $f(x)$ partiendo de una solución inicial, lo ideal sería utilizar estas direcciones de mayor crecimiento pero en sentido inverso, es decir, usar $-f'(x)$, que puede escribirse como:

$$-f'(x) = B - Ax.$$

Pero como estamos iterando, tenemos en realidad que:

$$-f'(x^{(i)}) = B - Ax^{(i)} = r^{(i)},$$

que resulta ser el *residuo*. En consecuencia, el residuo no es otra cosa que la dirección descendente más empinada para llegar al mínimo, o sea, la del *descenso más rápido*. Como partimos de un vector inicial, lo que nos interesa es obtener un coeficiente α que optimice cada paso utilizando la dirección más empinada y así obtener una aproximación $i+1$ más cercana a la solución «exacta». Para ello partamos de la expresión general

$$x^{(i+1)} = x^{(i)} + \alpha r^{(i)}.$$

Para obtener el α, minimizaremos la forma cuadrática. Así tenemos que:

$$\frac{df(x^{(i+1)})}{d\alpha} = f'(x^{(i+1)})^T\ \frac{dx^{(i+1)}}{d\alpha} = f'(x^{(i+1)})^T \cdot r^{(i)} = 0,$$

lo que equivale a decir que el residuo y el gradiente son *ortogonales*. Como además sabemos que $r^{(i+1)} = -f'(x^{(i+1)}$, entonces tenemos:

$$\begin{aligned}
-r^{(i+1)^T} \cdot r^{(i)} &= 0 \\
\left(B - Ax^{(i+1)}\right)^T \cdot r^{(i)} &= 0 \\
\left[B - A\left(x^{(i)} + \alpha_i \cdot r^{(i)}\right)\right]^T \cdot r^{(i)} &= 0 \\
\left(B - Ax^{(i)}\right)^T \cdot r^{(i)} - \alpha_i \left(Ar^{(i)}\right)^T \cdot r^{(i)} &= 0 \\
\left(B - Ax^{(i)}\right)^T \cdot r^{(i)} &= \alpha_i \left(Ar^{(i)}\right)^T \cdot r^{(i)} \\
r^{(i)^T} \cdot r^{(i)} &= \alpha_i r^{(i)^T} \cdot Ar^{(i)} \\
\alpha_i &= \frac{r^{(i)^T} \cdot r^{(i)}}{r^{(i)^T} \cdot Ar^{(i)}}.
\end{aligned}$$

Así, nuestro nuevo algoritmo es:

$$\begin{aligned} r^{(0)} &= B - Ax^{(0)} \\ \alpha_i &= \frac{{r^{(i)}}^T \cdot r^{(i)}}{{r^{(i)}}^T \cdot Ar^{(i)}} \\ x^{(i+1)} &= x^{(i)} + \alpha_i r^{(i)} \\ r^{(i+1)} &= r^{(i)} - \alpha_i A r^{(i)}. \end{aligned}$$

El criterio para interrupir las iteraciones es el mismo que el aplicado para el método de los residuos mínimos.

Método de los gradientes conjugados

El método anterior es una mejora notable al método de los residuos mínimos. No sólo mejora la velocidad de convergencia sino que reduce la cantidad de operaciones. Sin embargo tiene una desventaja importante: suele usar varias veces la misma dirección de acercamiento. Esto significa que no utiliza bien las direcciones más empinadas. Veamos por qué.

Vimos que el vector residuo es el gradiente de nuestra forma cuadrática. Supongamos que ésta sea solamente de dos variables, es decir, un paraboloide de revolución. El gradiente será un plano que pasa por un punto cuya «inclinación» nos da una idea del crecimiento (decrecimiento) en ese punto. Pero en realidad lo que tenemos son varias direcciones posibles que descienden rápidamente hacia el mínimo. El método anterior sólo exige que los residuos sean ortogonales, pero no se ocupa de las direcciones con las cuales se aproxima al siguiente resultado, con lo cual puede repetir cualquier dirección en el proceso iterativo hasta obtener la solución. Así pierde eficiencia.

La forma más rápida de llegar sería usar direcciones que no se repitan durante el proceso de descenso. ¿Cuál sería el conjunto de direcciones que harían más rápido ese descenso? La respuesta es: tomemos un conjunto de direcciones $d^{(0)}; d^{(1)}; \ldots; d^{(n-1)}$, tales que sean ortogonales entre sí, o sea que se cumpla que:

$$\begin{aligned} d^{(0)} \cdot d^{(1)} &= 0; \\ d^{(1)} \cdot d^{(2)} &= 0; \\ \cdots &\quad \cdots \\ d^{(i)} \cdot d^{(j)} &= 0, \text{ para } i \neq j. \end{aligned}$$

Entonces nuestra expresión inicial será

$$x^{(i+1)} = x^{(i)} + \alpha\, d^{(i)},$$

en lugar de la vista para el resto de los métodos.

Como hemos definido que las direcciones que aproximan nuestra solución son ortogonales, entonces también el error $e^{(i+1)}$ debería ser ortogonal, es decir, se debería cumplir que:

$$\begin{aligned} {d^{(i)}}^T \cdot e^{(i+1)} &= 0 \\ {d^{(i)}}^T \cdot \left(e^{(i)} + \alpha_i\, d^{(i)}\right) &= 0 \\ {d^{(i)}}^T \cdot e^{(i)} + \alpha_i\, {d^{(i)}}^T \cdot d^{(i)} &= 0 \\ {d^{(i)}}^T \cdot e^{(i)} &= -\alpha_i\, {d^{(i)}}^T \cdot d^{(i)} \\ \alpha_i &= -\frac{{d^{(i)}}^T \cdot e^{(i)}}{{d^{(i)}}^T \cdot d^{(i)}}. \end{aligned}$$

Sin embargo, este algoritmo no es muy útil pues debemos conocer el error que estamos cometiendo para obtener el coeficiente α_i. Y si conocemos $e^{(i+1)}$, conocemos la solución y no tendría sentido obtener el coeficiente α.

En lugar de proponer que el error sea ortogonal a la dirección, vamos a proponer que las direcciones sean *conjugadas*, también llamadas direcciones *ortogonales por A*. ¿Qué significa esto? Supongamos por un momento que trabajamos sobre una superficie esférica similar a un globo, y dibujamos sobre ésta dos líneas que sean ortogonales, como lo son, un meridiano y un paralelo. Si deformamos nuestro globo de manera que deje de ser esférico y se convierta en un elipsoide de revolución, las dos líneas se cortarán, pero *no serán ortogonales*. Si volvemos a transformar ese globo defomado en una esfera otra vez, dichas líneas volverán a ser ortogonales. Las direcciones en el elipsoide se denominan *conjugadas*.

La idea del método es partir de la situación del elipsoide, transformar los vectores de forma de llevarlos a la esfera, obtener allí las direcciones ortogonales y luego trabajar nuevamente en el elipsoide. De esa forma, las direcciones serán ortogonales en la superficie de la esfera, y conjugadas en el elipsoide. (Otro ejemplo en ese mismo sentido sería proyectar la esfera sobre un plano, práctica común de la *cartografía*.)

Para obtener nuestro nuevo algoritmo, vamos a proponer que la dirección $d^{(i)}$ sea ortogonal a $r^{(i+1)}$:

$$\begin{aligned}
-{d^{(i)}}^T \cdot r^{(i+1)} &= 0 \\
{d^{(i)}}^T \cdot Ae^{(i+1)} &= 0 \\
{d^{(i)}}^T \cdot A\left(e^{(i)} + \alpha_i\, d^{(i)}\right) &= 0 \\
{d^{(i)}}^T \cdot Ae^{(i)} + \alpha_i\, {d^{(i)}}^T \cdot Ad^{(i)} &= 0 \\
{d^{(i)}}^T \cdot Ae^{(i)} &= -\alpha_i\, {d^{(i)}}^T \cdot Ad^{(i)} \\
\alpha_i &= -\frac{{d^{(i)}}^T \cdot Ae^{(i)}}{{d^{(i)}}^T \cdot Ad^{(i)}} \\
\alpha_i &= \frac{{d^{(i)}}^T \cdot r^{(i)}}{{d^{(i)}}^T \cdot Ad^{(i)}}.
\end{aligned}$$

Con este coeficiente α_i nos aseguramos que nuestro método va aproximando la solución mediante direcciones conjugadas. Pero nos faltan hallar estas direcciones. ¿Cómo las obtenemos? La forma más sencilla es aplicar el método de Gram-Schmidt para ortogonalizar vectores. En este caso lo que haremos es obtener vectores conjugados a partir de un vector inicial, por lo que la fórmula de Gram-Schmidt queda de la siguiente forma:

$$d^{(i)} = u^{(i)} + \sum_{j=0}^{i-1} \beta_{ij} Ad^{(j)},$$

y el coeficiente β_{ij} lo obtenemos mediante:

$$\beta_{ij} = -\frac{{u^{(i)}}^T \cdot Ad^{(j)}}{{d^{(j)}}^T \cdot Ad^{(j)}},$$

siendo $u^{(i)}$ el vector a partir del cual obtenemos las direcciones conjugadas (ortogonales por A). (Véase [11].)

Nos falta definir el vector $u^{(i)}$. Si proponemos al vector $r^{(i)}$ tendremos que:

$${d^{(i)}}^T \cdot r^{(i)} = {r^{(i)}}^T \cdot r^{(i)},$$

y entonces, obtendremos lo siguiente:

$$\beta_{ij} = -\frac{{r^{(i)}}^T \cdot Ad^{(j)}}{{d^{(j)}}^T \cdot Ad^{(j)}}.$$

Ahora vamos a obtener el β_{ij} para poder encontrar nuestras direcciones conjugadas. Así, tenemos que:

$$\begin{aligned}
{r^{(i)}}^T \cdot r^{(j+1)} &= {r^{(i)}}^T \cdot r^{(j)} - \alpha_j {r^{(i)}}^T \cdot A r^{(j)} \\
\alpha_j {r^{(i)}}^T \cdot A r^{(j)} &= {r^{(i)}}^T \cdot r^{(j)} - {r^{(i)}}^T \cdot r^{(j+1)} \\
{r^{(i)}}^T \cdot A r^{(j)} &= \begin{cases} \dfrac{1}{\alpha_j} {r^{(j)}}^T \cdot r^{(j)} & \text{si } i = j \\ -\dfrac{1}{\alpha_j} {r^{(j+1)}}^T \cdot r^{(j+1)} & \text{si } i = j+1 \end{cases} \\
\beta_{j+1\,j} &= \frac{1}{\alpha_j} \frac{{r^{(j+1)}}^T r^{(j+1)}}{{d^{(j)}}^T A d^{(j)}}
\end{aligned}$$

Antes hemos obtenido que:

$$\alpha_j = \frac{{d^{(j)}}^T \cdot r^{(j)}}{{d^{(j)}}^T \cdot A d^{(j)}} \Rightarrow \frac{1}{\alpha_j} = \frac{{d^{(j)}}^T \cdot A d^{(j)}}{{d^{(j)}}^T \cdot r^{(j)}},$$

por lo tanto, finalmente tendremos que:

$$\beta_{j+1\,j} = \frac{{d^{(j)}}^T \cdot A d^{(j)}}{{d^{(j)}}^T \cdot r^{(j)}} \frac{{r^{(j+1)}}^T \cdot r^{(j+1)}}{{d^{(j)}}^T \cdot A d^{(j)}} = \frac{{r^{(j+1)}}^T \cdot r^{(j+1)}}{{d^{(j)}}^T \cdot r^{(j)}} = \frac{{r^{(j+1)}}^T \cdot r^{(j+1)}}{{r^{(j)}}^T \cdot r^{(j)}},$$

pues al $d^{(j)}$ lo obtenemos a partir del $r^{(j)}$. Simplificando la notación tenemos:

$$\beta_{j+1} = \frac{{r^{(j+1)}}^T r^{(j+1)}}{{r^{(j)}}^T \cdot r^{(j)}}.$$

Con este último coeficiente tenemos el algoritmo para el *método de los gradientes conjugados*, que resulta ser:

$$\begin{aligned}
d^{(0)} = r^{(0)} &= B - A x^{(0)} \\
\alpha_i &= \frac{{r^{(i)}}^T \cdot r^{(i)}}{{d^{(i)}}^T \cdot A d^{(i)}} \\
x^{(i+1)} &= x^{(i)} + \alpha_i d^{(i)} \\
r^{(i+1)} &= r^{(i)} - \alpha_i A d^{(i)} \\
\beta_{i+1} &= \frac{{r^{(i+1)}}^T \cdot r^{(i+1)}}{{r^{(i)}}^T \cdot r^{(i)}} \\
d^{(i+1)} &= r^{(i+1)} + \beta_{i+1} d^{(i)}.
\end{aligned}$$

2.10.4. Convergencia de los métodos no estacionarios

Analizaremos brevemente la convergencia de los métodos no estacionarios. En primer lugar nos ocuparemos rápidamente del método de los residuos mínimos, y luego de los otros dos métodos.

Método de los residuos mínimos

Ya habíamos dicho que para garantizar la convergencia de este método, la matriz A debe ser definida positiva. El siguiente teorema demuestra esta afirmación.

Teorema 2.9. Sea A una matriz definida positiva y sea

$$\mu = \lambda_{\text{mín}}\left(\frac{A+A^T}{2}\right);\ \sigma = \|A\|_2\,,$$

entonces el vector $r^{(i+1)}$ generado por el método de los residuos mínimos satisface la relación

$$\left\|r^{(i+1)}\right\|_2 \leq \left(1-\frac{\mu^2}{\sigma^2}\right)^{1/2}\left\|r^{(i)}\right\|_2,$$

y el algoritmo correspondiente converge para cualquier valor inicial de $x^{(0)}$.

La demostración de este teorema puede verse en [9].

Método del descenso más rápido

Para el análisis de la convergencia de este método (y el de los gradientes conjugados) nos basaremos en el estudio de los autovalores y autovectores de la matrioz A.

Supongamos que el vector $e^{(i)}$ sea un autovector asociado a un autovalor λ_e. Entonces el residuo se puede escribir como:

$$r^{(i)} = -Ae^{(i)} = -\lambda_e e^{(i)},$$

por lo tanto, es también un autovector.

De la misma forma podemos obtener $e^{(i+1)}$, pues es:

$$\begin{aligned} e^{(i+1)} &= e^{(i)} + \frac{{r^{(i)}}^T\cdot r^{(i)}}{{r^{(i)}}^T\cdot Ar^{(i)}}r^{(i)} \\ &= e^{(i)} + \frac{{r^{(i)}}^T\cdot r^{(i)}}{\lambda_e {r^{(i)}}^T\cdot r^{(i)}}\left(-\lambda_e e^{(i)}\right) \\ &= 0. \end{aligned}$$

Si uno elige $\alpha_i = \lambda_e$, ¡basta con una iteración para obtener el resultado «exacto»! Pero en realidad, debemos expresar $e^{(i)}$ como una combinación lineal de autovectores, es decir,

$$e^{(i)} = \sum_{j=1}^{n}\xi_j v^{(j)},$$

donde los $v^{(j)}$ son vectores ortonormales (elegidos así por conveniencia), y los ξ_j son las longitudes de cada vector. Entonces nos queda

$$\begin{aligned} r^{(i)} &= -Ae^{(i)} = -\sum_{j=1}^{n}\xi_j\lambda_j v^{(j)} \\ \left\|e^{(i)}\right\|^2 &= {e^{(i)}}^T\cdot e^{(i)} = \sum_j {\xi_j}^2 \\ {e^{(i)}}^T\cdot Ae^{(i)} &= \left[\sum_j \xi_j {v^{(j)}}^T\right]\left[\sum_j \xi_j\lambda_j v^{(j)}\right] \\ &= \sum_j {\xi_j}^2\lambda_j \\ \left\|r^{(i)}\right\|^2 &= {r^{(i)}}^T\cdot r^{(i)} = \sum_j {\xi_j}^2{\lambda_j}^2 \\ {r^{(i)}}^T\cdot Ar^{(i)} &= \sum_j {\xi_j}^2{\lambda_j}^3 \end{aligned}$$

Esta última expresión la obtenemos al tener en cuenta que el $r^{(i)}$ también se puede expresar como la combinación lineal de autovectores, y que su longitud es $-\xi_j\lambda_j$. Si volvemos a la expresión del vector $e^{(i+1)}$ tenemos:

$$\begin{aligned} e^{(i+1)} &= e^{(i)} + \frac{r^{(i)^T}\cdot r^{(i)}}{r^{(i)^T}\cdot Ar^{(i)}} r^{(i)} \\ &= e^{(i)} + \frac{\sum_j \xi_j{}^2\lambda_j{}^2}{\sum_j \xi_j{}^2\lambda_j{}^3} r^{(i)}, \end{aligned}$$

que nos muestra que α_i es un promedio ponderado de $\dfrac{1}{\lambda_j}$.

Para analizar la convergencia en forma más general vamos a definir primero la *norma energética* $\|e\|_A = \left(e^T\cdot Ae\right)^{1/2}$. Con esta norma tenemos:

$$\begin{aligned} \left\|e^{(i+1)}\right\|_A^2 &= e^{(i+1)^T}\cdot Ae^{(i+1)} \\ &= \left(e^{(i)^T} + \alpha_i r^{(i)^T}\right) A \left(e^{(i)} + \alpha_i r^{(i)}\right) \\ &= e^{(i)^T}\cdot Ae^{(i)} + 2\alpha_i r^{(i)^T}\cdot Ae^{(i)} + \alpha_i{}^2 r^{(i)^T}\cdot Ar^{(i)} \\ &= \left\|e^{(i)}\right\|_A^2 + 2\frac{r^{(i)^T}\cdot r^{(i)}}{r^{(i)^T}\cdot Ar^{(i)}}\left[-r^{(i)^T}\cdot r^{(i)}\right] + \left[\frac{r^{(i)^T}\cdot r^{(i)}}{r^{(i)^T}\cdot Ar^{(i)}}\right]^2 r^{(i)^T}\cdot Ar^{(i)} \\ &= \left\|e^{(i)}\right\|_A^2 - \frac{\left[r^{(i)^T}\cdot r^{(i)}\right]^2}{r^{(i)^T}\cdot Ar^{(i)}} \\ &= \left\|e^{(i)}\right\|_A^2 \left[1 - \frac{\left(\sum \xi_j^2\lambda_j^2\right)^2}{\sum \xi_j^2\lambda_j^3 \sum \xi_j^2\lambda_j}\right] \\ &= \left\|e^{(i)}\right\|_A^2 \omega^2 \quad \text{con } \omega^2 = 1 - \frac{\left(\sum \xi_j^2\lambda_j^2\right)^2}{\sum \xi_j^2\lambda_j^3 \sum \xi_j^2\lambda_j} \end{aligned}$$

Esto quiere decir que el error de la iteración $i+1$ es función de los autovalores de A. Como lo que interesa es un límite superior del error, y no el error en si mismo, si definimos que

$$\kappa = \frac{\lambda_{\text{máx}}}{\lambda_{\text{mín}}},$$

se puede demostrar que

$$\omega = \frac{\kappa - 1}{\kappa + 1},$$

con lo cual tenemos que

$$\left\|e^{(i)}\right\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^i \left\|e^{(0)}\right\|_A.$$

(La demostración indicada se puede ver en [11].)

Método de los gradientes conjugados

Para el método de los gradientes conjugados vale el mismo desarrollo hecho para el descenso más rápido, pero con una leve modificación se llega a que

$$\left\|e^{(i)}\right\|_A \leq 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^i \left\|e^{(0)}\right\|_A.$$

Podemos decir que el método converge más rápido que el método del descenso más rápido, pues en el primero la convergencia depende de $\sqrt{\kappa}$, mientras que en el segundo depende de κ. Puesto que κ es equivalente a la condición de A, finalmente tenemos que una matriz bien condicionada converge rápidamente a la solución, en tanto que no lo hace si está mal condicionada. Por esta razón, este método rara vez se aplica directamente sobre el sistema $Ax = B$, sino que se *precondiciona* a la matriz A con una matriz M, formando el sistema $M^TAx = M^TB$ de manera tal que M^TA suele tener un número de condición mucho menor que A.

Por otra parte, si la matriz está bien condicionada, el método de los gradientes conjugados converge luego de n iteraciones. Es más, si no hubieran problemas derivados de la representación numérica en las computadoras, el método convergería después de k iteraciones, siendo k el número de autovalores *no repetidos* de A.

2.10.5. Aspectos computacionales

En general, obtener una solución eficiente de un sistema de ecuaciones lineales por medio de métodos iterativos depende fuertemente de la elección del método. Si bien podemos esperar una menor eficiencia de estos métodos respecto de los métodos directos, los métodos iterativos suelen ser más fáciles de implementar y, como no hay que factorizar la matriz, permiten resolver sistemas mucho más grandes que los directos.

Como resumen de los métodos vistos, tenemos lo siguiente:

1. **Método de Jacobi**: Muy fácil de usar, pero sólo converge si la matriz es estrictamente diagonal dominante. Actualmente sólo se lo considera como una forma de introducción a los métodos iterativos.

2. **Método de Gauss-Seidel**: Converge más rápido que el de Jacobi, pero no puede competir con los métodos no estacionarios. Tiene la ventaja de que también converge si la matriz del sistema es simétrica y definida positiva.

3. **Método de las sobrerrelajaciones sucesivas**: Converge más rápido que Gauss-Seidel si $\omega > 1$, y suele converger con $\omega < 1$ cuando Gauss-Seidel no converge. Como vimos, la velocidad de convergencia depende de ω, valor que no es fácil de obtener en forma analítica. Obtener ese valor puede llevar a perder parte de esa ventaja.

4. **Método de los residuos mínimos**: Converge si la matriz A del sistema es definida positiva y mejora si además es simétrica. Es más fácil de programar pues hay que hacer operaciones matriciales (vectoriales). La convergencia puede ser lenta, similar a Jacobi.

5. **Método del descenso más rápido**: Se aplica a sistemas con matrices simétricas definidas positivas. Converge más rápido que el anterior pero si la matriz no está bien condicionada, no converge. Es más fácil de programar que el anterior porque reduce la cantidad de operaciones matriciales. Es equivalente a Gauss-Seidel.

6. **Método de los gradientes conjugados**: Se aplica a matrices simétricas definidas positivas. Cuando la matriz está bien condicionada y además tiene p autovalores repetidos y bien distribuidos, converge para $k = n - p$ iteraciones (convergencia supralineal). Por este motivo, suele usarse precondicionado para conseguir convergencias supralineales. Es más fácil de implementar que los anteriores métodos no estacionarios, pero suele tener problemas con el error de redondeo.

2.11. Errores de los métodos iterativos

En este punto analizaremos fundamentalmente los errores de los métodos iterativos estacionarios, pues son conceptualmente más fáciles de entender. Empezaremos por el error de truncamiento.

Supongamos que x sea la solución de nuestro sistema de ecuaciones y $x^{(k+1)}$ el resultado luego de $k+1$ iteraciones. Entonces podemos definir el error como

$$x^{(k+1)} - x = T\left(x^{(k)} - x\right).$$

Si sumamos y restamos $Tx^{(k+1)}$ tenemos

$$\begin{aligned} x^{(k+1)} - x &= T\left(x^{(k)} - x\right) + Tx^{(k+1)} - Tx^{(k+1)} \\ &= T\left(x^{(k)} - x^{(k+1)}\right) + T\left(x^{(k+1)} - x\right). \end{aligned}$$

Si tomamos las normas tenemos que

$$\begin{aligned} \left\|x^{(k+1)} - x\right\| &\leq \|T\| \left\|x^{(k+1)} - x^{(k)}\right\| + \|T\| \left\|x^{(k+1)} - x\right\| \\ (1 - \|T\|) \left\|x^{(k+1)} - x\right\| &\leq \|T\| \left\|x^{(k+1)} - x^{(k)}\right\| \\ \left\|x^{(k+1)} - x\right\| &\leq \frac{\|T\|}{(1 - \|T\|)} \left\|x^{(k+1)} - x^{(k)}\right\|. \end{aligned}$$

Por lo tanto, el error de truncamiento está dado por

$$E_T \cong \frac{\|T\|}{(1 - \|T\|)} \left\|x^{(k+1)} - x^{(k)}\right\|.$$

Para el caso del error inherente partimos de

$$x^{(k+1)} = Tx^{(k)} + C.$$

Si consideramos los errores inherentes del sistema, el resultado que obtendremos será en realidad $\bar{x}^{(k+1)}$. Supongamos que desechamos todos los errores de los pasos anteriores, es decir, que $x^{(k)} \equiv \bar{x}^{(k)}$, entonces tenemos que

$$\bar{x}^{(k+1)} = x^{(k)} - \delta x^{(k)} = (T - \delta T)x^{(k)} + (C - \delta C).$$

Como $x = Tx + C$, podemos hacer lo siguiente:

$$\begin{aligned} \bar{x}^{(k+1)} - x &= (T - \delta T)x^{(k)} + (C - \delta C) - Tx - C \\ &= T\left(x^{(k)} - x\right) - \delta T x^{(k)} - \delta C \\ &= T\left(x^{(k)} - x\right) - \delta T x^{(k)} - \delta C + Tx^{(k+1)} - Tx^{(k+1)} \\ &= T\left(x^{(k+1)-x}\right) + T\left(x^{(k)} - x^{(k+1)}\right) - \delta T x^{(k)} - \delta C. \end{aligned}$$

Si nuevamente tomamos las normas, obtenemos

$$\begin{aligned} \left\|x^{(k+1)} - x\right\| &\leq \|T\| \left\|x^{(k+1)} - x\right\| + \|T\| \left\|x^{(k+1)} - x^{(k)}\right\| + \|\delta T\| \left\|x^{(k)}\right\| + \|\delta C\| \\ (1 - \|T\|) \left\|x^{(k+1)} - x\right\| &\leq \|T\| \left\|x^{(k+1)} - x^{(k)}\right\| + \|\delta T\| \left\|x^{(k)}\right\| + \|\delta C\| \\ \left\|x^{(k+1)} - x\right\| &\leq \frac{\|T\|}{1 - \|T\|} \left\|x^{(k+1)} - x^{(k)}\right\| + \frac{\|\delta T\|}{1 - \|T\|} \left\|x^{(k)}\right\| + \frac{\|\delta C\|}{1 - \|T\|}. \end{aligned}$$

Si analizamos en detalle esta última expresión, vemos que se repite el error de truncamiento (primer término de la derecha). En consecuencia, el error inherente está dado por

$$E_I \cong \frac{\|\delta T\|}{1 - \|T\|} \left\|x^{(k)}\right\| + \frac{\|\delta C\|}{1 - \|T\|}.$$

Finalmente, analicemos el error de redondeo. Una vez más, partamos de la expresión

$$x^{(k+1)} = Tx^{(k)} + C,$$

y nuevamente supongamos que lo que obtenemos es en realidad es $\bar{x}^{(k+1)}$ y que $x^{(k)} \equiv \bar{x}^{(k)}$. Entonces nos queda:

$$\bar{x}^{(k+1)} = Tx^{(k)} + C.$$

Para cada componente de $x^{(k+1)}$ tenemos

$$x_i^{(k+1)} - \delta x_i^{(k+1)} = \left[\text{fl}\left(\sum_{j=1}^{n} t_{ij} x_j^{(k)} \right) + c_i \right] (1 - \delta_i).$$

Si hacemos un análisis retrospectivo del error («backward error»), y asumimos que $n\mu \leq 0,01$, nos queda que

$$\delta x_i^{(k+1)} = \left[\sum_j t_{ij} x_j^{(k)} 1,01(n+2-j)\mu\theta_j \right] (1 + \delta_i) + x_i^{(k+1)} \delta_i,$$

con $|\theta_j| \leq 1$ y $|\delta_i| \leq \mu$.

Consideremos ahora el hecho de que generalmente las matrices de los sistemas son ralas. Entonces podemos definir que

$$\begin{aligned} p &= \max_{1 \leq i \leq n} \{p_i\}, \text{ con } p_i: \text{ cantidad de elementos no nulos en una fila.} \\ q &= \max_{1 \leq i,j \leq n} \{|t_{ij}|\}, \end{aligned}$$

entonces, si tomamos normas nos queda

$$\left\| \delta x_i^{(k+1)} \right\| \leq q \left\| x^{(k)} \right\| 1,01 \left[\sum_{j=1}^{p} (p+2-j) \right] \mu + \left\| x_i^{(k+1)} \right\| \mu,$$

y como $x^{(k)} \approx x^{(k+1)}$, podemos escribir que

$$\frac{\left\| \delta x_i^{(k+1)} \right\|}{\left\| x^{(k)} \right\|} \leq \left(q\, 1,01 \frac{p^2 + 3p}{2} + 1 \right) \mu.$$

Ahora estimemos la diferencia $\bar{x}^{(k+1)} - x$. Sabemos que

$$\bar{x}^{(k+1)} = Tx^{(k)} + C - \delta x^{(k+1)},$$

entonces

$$\begin{aligned} \bar{x}^{(k+1)} - x &= Tx^{(k)} + C - \delta x^{(k+1)} - Tx - C \\ &= T\left(x^{(k)} - x \right) - \delta x^{(k+1)}. \end{aligned}$$

Si nuevamente sumamos y restamos $Tx^{(k+1)}$, obtenemos

$$\bar{x}^{(k+1)} - x = T\left(x^{(k)} - \bar{x}^{(k+1)} \right) + T\left(\bar{x}^{(k+1)} - x \right) - \delta x^{(k+1)}.$$

Una vez más, tomemos las normas, con lo cual nos queda

$$\begin{aligned}
\left\|\bar{x}^{(k+1)} - x\right\| &= \|T\| \left\|x^{(k+1)} - \bar{x}^{(k)}\right\| + \|T\| \left\|\bar{x}^{(k+1)} - x\right\| + \left\|\delta x^{(k+1)}\right\| \\
(1 - \|T\|) \left\|\bar{x}^{(k+1)} - x\right\| &= \|T\| \left\|x^{(k+1)} - \bar{x}^{(k)}\right\| + \left\|\delta x^{(k+1)}\right\| + \\
\left\|\bar{x}^{(k+1)} - x\right\| &= \frac{\|T\|}{1 - \|T\|} \left\|x^{(k+1)} + \bar{x}^{(k)}\right\| - \frac{\left\|\delta x^{(k+1)}\right\|}{1 - \|T\|}
\end{aligned}$$

Puesto que $\left\|\delta x_i^{(k+1)}\right\| \leq \left(q\, 1,01 \frac{p^2+3p}{2} + 1\right) \left\|x^{(k)}\right\| \mu$ y como el primer término corresponde al error de truncamiento, nos queda que

$$E_R \leq \frac{\left\|x^{(k)}\right\|}{1 - \|T\|} \left(q\, 1,01 \frac{p^2+3p}{2} + 1\right) \mu.$$

Finalmente, el error total al aplicar un método iterativo estacionario es la suma de todos los errores, es decir,

$$\begin{aligned}
\left\|x^{(k+1)} - x\right\| \leq & E_T + E_I + E_R \\
\leq & \frac{\|T\|}{(1 - \|T\|)} \left\|x^{(k+1)} - x^{(k)}\right\| + \frac{\|\delta T\|}{1 - \|T\|} \left\|x^{(k)}\right\| + \frac{\|\delta C\|}{1 - \|T\|} \\
& + \frac{\left\|x^{(k)}\right\|}{1 - \|T\|} \left(q\, 1,01 \frac{p^2+3p}{2} + 1\right) \mu \\
\left\|x^{(k+1)} - x\right\| \leq & \frac{1}{1 - \|T\|} \Bigg[\|T\| \left\|x^{(k+1)} - x^{(k)}\right\| + \|\delta T\| \left\|x^{(k)}\right\| + \|\delta C\| \\
& + \left\|x^{(k)}\right\| \left(q\, 1,01 \frac{p^2+3p}{2} + 1\right) \mu \Bigg].
\end{aligned}$$

Como hemos visto en el capítulo 1, siempre es conveniente que los errores de truncamiento e inherentes predominen respecto al de redondeo. En consecuencia, siempre debemos tratar que $E_R < E_T < E_I$, es decir, que el error de redondeo sea el de menor incidencia, y si es posible, despreciable [4].

2.12. Notas finales

Los métodos vistos no son los únicos disponibles para resolver sistemas de ecuaciones lineales. Dentro de los métodos directos también están el método QR y el de la descomposición por el valor singular, método muy usado con matrices muy mal condicionadas, aunque algunos autores sostienen que debería ser un método básico, igual que eliminación de Gauss.

Algo similar ocurre con los métodos iterativos, particularmente con los no estacionarios. Además de los tres que hemos visto, están el método de los residuos mínimos generalizado, el método de los gradientes biconjugados y el de los gradientes conjugados cuadrado, el método por iteraciones de Chebichev, más otros derivados fundamentalmente a partir del método del gradiente conjugado y de los residuos mínimos.

La existencia de varios métodos refleja que la elección de uno en particular depende fundamentalmente de las propiedades de la matriz de coeficientes del sistema. Es por esto que cada vez es más importante saber qué problema (o fenómeno físico) está siendo representado con el sistema a resolver. Si buscamos información sobre la utilización de cada método, veremos que están muy ligados al tipo de problema que se estudia y resuelve.

[4] González, en su libro, dice que $E_T < E_R$ pero eso se contrapone con lo que afirman otros autores. La razón principal es que el error de redondeo tiene un comportamiento «errático», lo que hace difícil acotarlo. (Ver ejemplo en el capítulo 1 con el error de discretización.)

En muchos campos de la ingeniería, los sistemas de ecuaciones lineales están directamente relacionados con la resolución de ecuaciones diferenciales en derivadas parciales, por eso es que métodos para resolver este tipo de problemas, como el de las diferencias finitas o de los elementos finitos, hayan impulsado el desarrollo de métodos más potentes y más precisos, dado que mayormente trabajan con matrices de dimensiones muy grandes que además suelen ser ralas.

Finalmente, quien quiera adentrarse en los métodos iterativos no estacionarios, el libro de Y. Saad es una muestra muy interesante de cómo la necesidad de contar con algoritmos cada vez más veloces y con capacidad de resolver grandes sistemas de ecuaciones, disparan el desarrollo y la investigación de la matemática aplicada.

Capítulo 3

Ecuaciones no Lineales

3.1. Introducción

En el capítulo anterior vimos como resolver sistemas de ecuaciones lineales (de la forma $Ax = B$ o $Ax - B = 0$), sistemas cuya solución es única. Pero existe una gran cantidad de problemas que no pueden representarse mediante ecuaciones lineales. Muchas cuestiones que debe enfrentar la ingeniería no tienen solución única o no se pueden obtener en forma algebraica.

Tomemos el siguiente caso: supongamos que queremos desarrollar una mejora en la costa y para ello necesitamos un recinto cerrado, el cual vamos a rellenar con arena. Para conseguir ese recinto necesitamos una pared de contención que construiremos con tablestacas. Para diseñar las tablestacas debemos resolver una ecuación del tipo $a_0 + a_1\ x + a_2\ x^2 + a_3\ x^3 = 0$, donde x es la longitud de hinca, también conocida como «ficha». Esta ecuación tiene tres soluciones posibles (tres raíces). Si bien existe una solución algebraica para obtener las raíces de una ecuación de tercer grado, en general, es mucho más práctico resolverla mediante algún método iterativo, y obtener aquella solución (raíz) que sea compatible con el problema.

Como dijimos, están también aquellas ecuaciones que no tienen solución algebraica y que, por lo tanto, sólo podrán resolverse mediante aproximaciones. Tenemos como ejemplo, calcular la longitud de onda de una ola marítima en aguas poco profundas. La expresión para esto es:

$$L = L_0 \tanh\left(\frac{2\pi}{L}x\right),$$

donde L_0 es la longitud de onda en aguas profundas ($x \geq \dfrac{L}{2}$) y x es la profundidad del mar. Esta expresión es válida para $0 \leq x \leq \dfrac{L}{2}$. Como podemos ver, esta ecuación no tiene solución algebraica, y, en consecuencia, el único modo de obtenerla es mediante un método iterativo. (Cuando $x > \frac{L}{2}$, entonces $\tanh\left(\frac{2\pi}{L}x\right) \cong 1$, y $L = L_0$.)

Dado que este tipo de problemas son regularmente comunes en la ingeniería, en este capítulo nos ocuparemos de estudiar los distintos métodos para resolver ecuaciones no lineales, de manera de obtener resultados muy precisos.

Como repaso, recordemos los teorema del valor medio y del valor intermedio.

Teorema 3.1. *(Teorema del valor medio.)* Si $f \in \mathrm{C}[a;b]$ y f es diferenciable en $(a;b)$, entonces existirá un número c en $(a;b)$ tal que

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$

Teorema 3.2. *(Teorema del valor intermedio.)* Si $f \in \mathrm{C}[a;b]$ y M es un número cualquiera entre $f(a)$ y $f(b)$, existirá un número c en $(a;b)$ para el cual $f(c) = M$.

3.2. Método de la bisección

Supongamos que tenemos una función cualquiera $f(x)$ y debemos hallar el valor de $\bar{x}$, tal que $f(\bar{x}) = 0$. Asumamos que $\bar{x}$ está incluido en el intervalo $(a; b)$, con $b > a$. Para que esto sea cierto, generalmente se verifica que $f(a) \cdot f(b) < 0$. (Sin embargo, hay casos en que $f(\bar{x}) = 0$, $\bar{x} \in (a; b)$ y no se cumple que $f(a) \cdot f(b) < 0$.)

Puesto que $\bar{x} \in (a; b)$, calculemos nuestra primera aproximación tomando el valor medio del intervalo, es decir,

$$x_1 = a + \frac{b-a}{2} = \frac{b+a}{2}.$$

Para saber si es o no solución debemos verificar que $f(x_1) = 0$. Si no lo es, debemos volver a obtener una aproximación mediante un esquema similar. Para ello, verifiquemos que $f(a) \cdot f(\bar{x}) < 0$. Si es cierto, entonces nuestro nuevo intervalo será $(a; \bar{x})$, si no lo es, nuestro intervalo será $(\bar{x}; b)$. Supongamos, por simplicidad, que $f(a) \cdot f(\bar{x}) < 0$ y que nuestro nuevo intervalo es $(a; \bar{x})$. Con el mismo método usado antes, nuestra nueva aproximación es

$$x_2 = \frac{x_1 + a}{2} = \frac{\frac{b+a}{2} + a}{2} = \frac{b+a}{4} + \frac{a}{2},$$

que también puede escribirse como

$$x_2 = x_1 - \frac{b-a}{4}.$$

Nuevamente verificamos si $f(x_2) = 0$. Si seguimos iterando hasta obtener x_n, tenemos que

$$x_n = x_{n-1} - \frac{b-a}{2^n},$$

por lo que también podemos decir que

$$|\bar{x} - x_n| \leq \frac{b-a}{2^n}.$$

Si queremos hallar el valor exacto de $\bar{x}$ deberíamos iterar hasta que $|\bar{x} - x_n| = 0$. Pero si establecemos una tolerancia de modo que $|\bar{x} - x_n| < \varepsilon$, entonces tendremos que la cantidad aproximada de iteraciones para obtener este resultado es

$$\begin{aligned} |\bar{x} - x_n| &\leq \frac{b-a}{2^n} < \varepsilon \\ \frac{b-a}{2^n} &< \varepsilon \\ n &> \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln(2)}. \end{aligned}$$

Por ejemplo, si nuestra tolerancia la expresamos como $\varepsilon = 10^{-t}$, para que el método de la bisección converja se necesitarán $n > \dfrac{\ln(b-a) - \ln(10^{-t})}{\ln(2)}$ iteraciones. Si lo desarrollamos un poco más, tenemos:

$$\begin{aligned} n &> \frac{\ln(b-a)}{\ln(2)} + t \cdot \frac{\ln(10)}{\ln(2)} \\ n &> \frac{1}{\ln(2)} \left[\ln(b-a) + t \cdot \ln(10)\right], \end{aligned}$$

y podemos escribir que

$$\begin{aligned} n &> \frac{1}{0,69} \left[\ln(b-a) + t \cdot 2,30\right] \\ n &> 1,44 \cdot \ln(b-a) + 3,32 \cdot t, \end{aligned}$$

con lo cual la cantidad de iteraciones depende mucho más de la tolerancia que del intervalo.

Este método de aproximación de las soluciones se conoce como *método de la bisección*. Es muy sencillo y tiene la ventaja de que siempre converge, pues nada exige a la función a la cual se le quiere calcular la raíz, salvo que se cumpla que $f(x_{k-1}) \cdot f(x_k) < 0$, mientras se está iterando. Hemos estimado cuantas iteraciones son necesarias para encontrar una solución aceptable a partir del error absoluto, pero en realidad los criterios para detener el procedimiento pueden ser los siguientes:

$$\begin{aligned} |x_n - x_{n-1}| &\leq \varepsilon, \\ \frac{|x_n - x_{n-1}|}{|x_n|} &\leq \varepsilon, \\ |f(x_n|) &\leq \varepsilon. \end{aligned}$$

donde ε es la tolerancia que ya mencionamos.

Cualquiera de los tres criterios es bueno para detener el proceso, pero el segundo es el más efectivo, pues se basa en el error relativo, y nos da una idea aproximada de la cantidad de cifras o dígitos significativos que tiene el resultado obtenido, pero a costa de hacer todavía mucho más lenta la convergencia, por lo ya visto al estimar n. En cambio, el último es el menos confiable, pues por definición $f(x_n)$ tiende a cero, con lo cual siempre es pequeño.

Si bien no tiene problemas por la convergencia, hemos visto que el método resulta muy lento para alcanzar un resultado aceptable. Además, según sea el criterio de interrupción aplicado, en muchas ocasiones puede despreciar un resultado intermedio más preciso. Es por eso que no suele utilizarse como único método para alcanzar la solución.

3.3. Método de la falsa posición o «regula falsi»

Hay otro método que también se basa en ir «achicando» el intervalo en el que se encuentra la solución. Se trata del *método de la falsa posición o «regula falsi»*. Consiste en trazar la cuerda que une los puntos $f(a)$ y $f(b)$ de la función dada e ir reduciendo el intervalo hasta obtener el valor de $\bar{x}$ tal que $f(\bar{x}) = 0$. Al igual que para el método de la bisección, debemos empezar por calcular x_1. Existen dos métodos equivalentes para obtenerlo:

$$\begin{aligned} x_1 &= a - \frac{f(a)\,(b-a)}{f(b) - f(a)}, \text{ o} \\ x_1 &= b - \frac{f(b)\,(b-a)}{f(b) - f(a)}. \end{aligned}$$

La forma de aplicar el método es la siguiente. Utilicemos la primera expresión para obtener x_1; entonces verifiquemos que $f(x_1) \cdot f(a) < 0$. Si esto es cierto, para obtener nuestra segunda aproximación tendremos la siguiente expresión:

$$x_2 = a - \frac{f(a)\,(x_1 - a)}{f(x_1) - f(a)},$$

caso contrario, nos queda esta otra expresión:

$$x_2 = x_1 - \frac{f(x_1)\,(b - x_1)}{f(b) - f(x_1)}.$$

Algo similar ocurre si partimos de la segunda. Si $f(x_1) \cdot f(b) < 0$, nos queda

$$x_2 = b - \frac{f(b)\,(b - x_1)}{f(b) - f(x_1)},$$

y si no

$$x_2 = x_1 - \frac{f(x_1)\,(x_1 - a)}{f(x_1) - f(a)}.$$

Este método no es una gran mejora al método de la bisección, aunque al trazar las cuerdas, hace uso de la función y generalmente suele converger un poco más rápido. Un punto importante a tener en cuenta es que al igual que en el método de la bisección, los sucesivos x_k se encuentran siempre en el intervalo de análisis (el intervalo $[x_{k-2}; x_{k-1}]$)y, por lo tanto, en el intervalo $[a; b]$ inicial. Y análogamente al método de la bisección, es posible que desprecie soluciones más precisas obtenidas en pasos intermedios.

3.4. Método de las aproximaciones sucesivas o punto fijo

Puesto que los métodos anteriores no tienen una convergencia rápida, no son muy prácticos para resolver problemas de gran complejidad. Veremos a continuación un método mucho más poderoso y efectivo.

Supongamos que nuestro problema a resolver, $f(x) = 0$, lo escribimos de una manera levemente diferente:

$$f(x) = x - g(x) = 0.$$

Es evidente que podemos despejar x de esta ecuación sin problemas, por lo que finalmente nos queda:

$$x = g(x),$$

es decir, nuestro problema se resume a encontrar una función $g(x)$. Pero como estamos resolviéndolo en forma iterativa, la expresión que nos queda es:

$$x_{k+1} = g(x_k),$$

para $k = 1; 2; \ldots; n$. El esquema entonces es sencillo: partiendo de una solución inicial, por ejemplo x_0, luego de efectuar n iteraciones tendremos nuestra solución aproximada x_n, que estará mucho más cerca del resultado «exacto» que nuestro valor inicial.

Veamos entonces un ejemplo de cómo aplicar el método. Supongamos que nuestra función es

$$x^4 - 4x^3 - x^2 + 16x - 12 = 0.$$

Sabemos que existe un método algebraico para obtener las raíces de este polinomio, pero hagamos uso del método para obtener la raíz en el intervalo $(0,5; 1,5)$. Para ello, propongamos la siguiente función $g(x)$:

$$g_1(x) = 4x^3 - x^4 + x^2 - 15x + 12,$$

y resolvamos en forma iterativa tomando $x_0 = 0,5$. Calculemos $g_1(x_0)$ y obtengamos x_1, luego x_2 y así sucesivamente:

$$\begin{aligned} x_1 = g_1(x_0) &= 4(0,5)^3 - 0,5^4 + 0,5^2 - 15(0,5) + 12 = 5,188 \\ x_2 = g_1(x_1) &= 4(5,188)^3 - 5,188^4 + 5,188^2 - 15(5,188) + 12 = -204. \end{aligned}$$

Es fácil notar que tenemos un problema. Como dijimos, la raíz buscada se encuentra en el intervalo $(0,5; 1,5)$, pero ambos resultados nos dieron fuera de dicho intervalo, y en el caso de x_2, de signo opuesto. Evidentemente, esta función $g_1(x)$ no nos sirve.

Cambiemos la función y volvamos a intentarlo. Probemos con la siguiente función:

$$g_2(x) = \sqrt[4]{4x^3 + x^2 - 16x + 12},$$

e iniciemos el proceso con el mismo x_0. En este caso al hacer diez iteraciones obtenemos el siguiente resultado: $x_{10} = 1,0005$, valor que está dentro del intervalo. Verifiquemos si este valor es «correcto» calculando $f(x_{10})$:

$$f(1,0005) = 1,0005^4 - 4(1,0005)^3 - 1,0005^2 + 16(1,0005) - 12 = 0,0033.$$

Este valor puede considerarse cercano cero y por lo tanto, hemos podido encontrar la raíz buscada.

¿Pero por qué fallamos al usar la primera función? Para entender esto veamos los siguientes teoremas.

Teorema 3.3. Si $g(x) \in \mathrm{C}(a;b)$ y $g(x) \in [a;b]$ para todo $x \in [a;b]$, entonces $g(x)$ tiene un punto fijo en $[a;b]$.

Teorema 3.4. Si $g'(x)$ existe en $[a;b]$, y existe una constante $m < 1$, tal que

$$\left|g'(x)\right| \leq m, \text{ para toda } x \in [a;b],$$

entonces, el punto fijo en en $[a;b]$ es único.

La demostración de estos teoremas puede verse en [1].

Estos teoremas son *suficientes* pero no necesarios, es decir, pueden no cumplirse y existir dicho punto, tal como vimos en el ejemplo anterior. La función $g_1(x)$ no cumple con los teoremas antes expuestos, sin embargo el punto fijo existe[1].

Si miramos la función $g_1(x)$ rápidamente notamos que la función $g_1(x)$ en $0,5$ no se «mapea» en el intervalo dado, por lo tanto, no se puede asegurar que exista un punto fijo. Y si hallamos la primera derivada en ese punto tenemos que $|g_1'(0,5)| = |-11,5| = 11,5 > 1$, con lo cual si existiera el punto fijo, no podríamos asegurar que dicho punto fijo sea único. No ocurre lo mismo con la función $g_2(x)$ puesto que $g_2(0,5) = 1,476 \in [0,5;1,5]$ y $|g_2'(0,5)| = |-0,932| = 0,932 < 1$, con lo cual el punto fijo es único. En realidad, deberíamos haber verificado ambas funciones g para varios puntos del intervalo, pero al comprobar que el punto de partida no cumple con las condiciones de ambos teoremas (función $g_1(x)$), nos indica que esta función no es convergente.

Verificado que la función $g(x)$ es convergente, nos falta definir el o los criterios de interrupción. Como en los casos anteriores, éstos son similares a los ya vistos, es decir,

$$\begin{aligned} |x_n - x_{n-1}| &\leq \varepsilon, \\ \frac{|x_n - x_{n-1}|}{|x_n|} &\leq \varepsilon, \\ |f(x_n| &\leq \varepsilon. \end{aligned}$$

Con el mismo ejemplo tenemos una pregunta: ¿cómo podemos obtener una solución por aproximaciones sucesivas (o punto fijo) que tenga una convergencia rápida? Para ello tenemos el siguiente teorema.

Teorema 3.5. Sea $g(x) \in \mathrm{C}[a;b]$ tal que $g(x) \in [a;b]$ para toda x en $[a;b]$, que existe $g'(x)$ en $[a;b]$ y que una constante $k < 1$ cuando

$$\left|g'(x)\right| \leq k, \text{ para toda } x \in (a;b).$$

Entonces, para cualquier número $x_0 \in [a;b]$, la sucesión definida por

$$x_n = g(x_{n-1}), \ n \geq 1,$$

converge en el único punto fijo $\bar{x}$ en $[a;b]$.

[1]Para $x = 1$ se tiene $g_1(x) = 1$, por lo tanto, el punto fijo existe.

Demostración El teorema 3.5 implica que existe un punto fijo en $[a;b]$. Como $g(x)$ «mapea» $[a;b]$ en sí mismo, la sucesión $\{x_n\}_{n=0}^{\infty}$ se define para todo $n \geq 0$ y $x_n \in [a;b]$ para todo n. Dado que $|g'(x)| \leq k$, si aplicamos el teorema del valor medio, tenemos

$$|x_n - \bar{x}| = |g(x_{n-1}) - g(\bar{x})| = \left|g'(\xi)\right| |x_{n-1} - \bar{x}| \leq k \, |x_{n-1} - \bar{x}| \,,$$

donde $\xi \in (a;b)$. En forma inductiva obtenemos

$$|x_n - \bar{x}| \leq k \, |x_{n-1} - \bar{x}| \leq k^2 \, |x_{n-2} - \bar{x}| \leq \ldots \leq k^n \, |x_0 - \bar{x}| \,.$$

Como $k < 1$, entonces

$$\lim_{n\to\infty} |x_n - \bar{x}| \leq \lim_{n\to\infty} k^n \, |x_0 - \bar{x}| = 0,$$

y la sucesión $\{x_n\}_{n=0}^{\infty}$ converge a $\bar{x}$.

Corolario 3.5.1. Si g satisface las hipótesis de teorema 3.5, las cotas de error que supone utilizar x_n para aproximar $\bar{x}$ están dadas por

$$|x_n - \bar{x}| \leq k^n \, \text{máx}\{x_0 - a; b - x_0\},$$

y por

$$|x_n - \bar{x}| \leq \frac{k^n}{1-k} \, |x_1 - x_0| \,, \text{ para toda } n \geq 1.$$

Demostración La primera cota viene de:

$$|x_n - \bar{x}| \leq k^n \, |x_o - \bar{x}| \leq k^n \, \text{máx}\{x_0 - a; b - x_0\},$$

porque $x \in (a;b)$.

Con $x \geq 1$, la demostración del teorema 3.5 implica que

$$|x_{n+1} - \bar{x}| = |g(x_n) - g(x_{n-1})| \leq |x_n - x_{n-1}| \leq \ldots \leq k^n \, |x_1 - x_0| \,.$$

En consecuencia, cuando $m > n \geq 1$,

$$\begin{aligned} |x_m - x_n| &= |x_m - x_{m-1} + x_{m-1} - x_{m-2} + \ldots + x_{n+1} - x_n| \\ &\leq |x_m - x_{m-1}| + |x_{m-1} - x_{m-2}| + \ldots + |x_{n+1} - x_n| \\ &\leq k^{m-1} \, |x_1 - x_0| + k^{m-2} \, |x_1 - x_0| + \ldots + k^n \, |x_1 - x_0| \\ &= k^n \left(1 + k + k^2 + \ldots + k^{m-n-1}\right) |x_1 - x_0| \,. \end{aligned}$$

Por el mismo teorema, tenemos que $\lim_{n\to\infty} x_m = \bar{x}$, por lo tanto

$$|\bar{x} - x_n| = \lim_{n\to\infty} \leq k^n \, |x_1 - x_0| \sum_{i=0}^{\infty} k^i.$$

Pero $\sum_{i=0}^{\infty} k^i$ es una serie geométrica con razón k. Como $0 < k < 1$, esta sucesión converge a $\dfrac{1}{1-k}$, por lo que nos queda que

$$|\bar{x} - x_n| \leq \frac{k^n}{1-k} \, |x_1 - x_0| \,.$$

Podemos ver que como $|g'(x)| \leq k$, la convergencia depende de la primera derivada de $g(x)$. Cuanto más chico sea k, más rápida será convergencia.

3.5. Método de Newton-Raphson

Este método es uno de los más poderosos que se conocen para resolver ecuaciones de la forma $f(x) = 0$. Una primera aproximación al método es partir del método de la falsa posición, y en vez de trazar una cuerda entre los dos extremos del intervalo, trazamos una tangente, que pase por un punto. Supongamos que para el mismo intervalo $[a; b]$ trazamos la tangente que pasa por $f(b)$. La ecuación de la recta tangente será

$$t(x) = f'(b)(x - b) + f(b).$$

Cuando se cumpla que $f(x) = 0$ se debará cumplir que $t(x) = 0$. Por lo tanto podríamos haller un valor x_1 tal que $t(x_1) = 0$ para ir aproximando nuestra raíz. Así obtenemos

$$\begin{aligned} t(x_1) &= 0 = f'(b)(x_1 - b) + f(b) \\ x_1 &= b - \frac{f(b)}{f'(b)}. \end{aligned}$$

Si $f(x_1) \neq 0$, podmeos repetir el procedimiento otra vez para obtener un x_2. En definitiva, podemos crear una aproximación iterativa de la siguiente forma:

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

Existe forma de deducirlo a través de la serie de Taylor. Supongamos que $f(x) \in \mathrm{C}^2[a, b]$, y sea $\hat{x}$ una aproximación de $\bar{x}$ tal que $f(\bar{x}) = 0$. También que $f'(\hat{x}) \neq 0$ y $|\hat{x} - \bar{x}|$ sea pequeño. Desarrollemos el primer polinomio de Taylor para $f(\hat{x})$ expandida alrededor de x,

$$f(x) = f(\hat{x}) + f'(\hat{x})(x - \hat{x}) + f''(\xi(x))\frac{(x - \hat{x})^2}{2},$$

donde $\xi(x)$ está entre x y $\hat{x}$. Puesto que $f(\bar{x}) = 0$, entonces para $x = \bar{x}$ tenemos

$$0 = f(\hat{x}) + f'(\hat{x})(\bar{x} - \hat{x}) + f''(\xi(\bar{x}))\frac{(\bar{x} - \hat{x})^2}{2}.$$

Al suponer que $|\hat{x} - \bar{x}|$ es pequeño, podemos despreciar $(\bar{x} - \hat{x})^2$, con lo que nos queda

$$0 = f(\hat{x}) + f'(\hat{x})(\bar{x} - \hat{x}),$$

y despejando $\bar{x}$ de la ecuación nos queda

$$\bar{x} = \hat{x} - \frac{f(\hat{x})}{f'(\hat{x})}.$$

Y si en lugar de aproximar con $\hat{x}$ lo hacemos con x_0, entonces generamos una sucesión $\{x_n\}$ definida por

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

que es la misma expresión que ya vimos.

De este desarrollo podemos ver que el error cometido es proporcional a $(\bar{x} - x_n)^2$ o a $f''(x_n)$ (puesto que cuando $x_n \approx \bar{x}$ podemos suponer que $\xi(x_n) \approx x_n$). De ahí que podemos aplicar los mismos criterios de interrupción que en los otros métodos.

También podemos observar que si no elegimos un x_0 lo suficientemente cerca, el método puede no converger. Para esto tenemos el siguiente teorema.

Teorema 3.6. Sea $f \in \mathrm{C}^2[a; b]$; si $\bar{x} \in [a; b]$ es tal que $f(\bar{x}) = 0$ y $f'(\bar{x}) \neq 0$, entonces existe un $\delta > 0$ tal que el método de Newton-Raphson genera una sucesión $\{x_n\}_{n=1}^{\infty}$ que converge a $\bar{x}$ para cualquier aproximación inicial $x_0 \in [\bar{x} - \delta; \bar{x} + \delta]$.

Demostración La demostración se basa en analizar el método de Newton-Raphson como si fuera el método de las aproximaciones sucesivas, tomando que $x_n = g(x_{n-1})$, $n \geq 1$, y que

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Entonces, sea k un número cualquiera en $(0;1)$. En primer lugar debemos encontrar un intervalo $[\bar{x}-\delta;\bar{x}+\delta]$ que g «mapee» en sí mismo y en el que $|g'(x)| \leq k$ para toda $x \in [\bar{x}-\delta;\bar{x}+\delta]$.

Como $f'(\bar{x}) \neq 0$ y $f'(\bar{x})$ es continua, existe $\delta_1 > 0$ tal que $f'(x) \neq 0$ para $x \in [\bar{x}-\delta_1;\bar{x}+\delta_1] \subset [a;b]$. Por lo tanto, g está definida y es continua en $[\bar{x}-\delta_1;\bar{x}+\delta_1]$. Por otro lado tenemos que

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2},$$

para $x \in [\bar{x}-\delta_1;\bar{x}+\delta_1]$ y como $f \in \mathrm{C}^2[a;b]$, tendremos que $g \in \mathrm{C}^1[a;b]$.

Como hemos supuesto que $f(\bar{x}) = 0$, entonces

$$g'(\bar{x}) = \frac{f(\bar{x})f''(\bar{x})}{[f'(\bar{x})]^2} = 0.$$

Además g' es continua y k es tal que $0 < k < 1$, entonces existe un δ, tal que $0 < \delta < \delta_1$, y

$$\left|g'(x)\right| \leq k \text{ para toda } x \in [\bar{x}-\delta;\bar{x}+\delta].$$

Nos falta todavía demostrar que $g : [\bar{x}-\delta;\bar{x}+\delta] \to [\bar{x}-\delta;\bar{x}+\delta]$. Si $x \in \bar{x}-\delta;\bar{x}+\delta]$. El teorema del valor medio implica que existe un número ξ entre x y $\bar{x}$ para el que se cumple

$$|g(x) - g(\bar{x})| = \left|g'(\xi)\right| |x - \bar{x}| .$$

Por lo tanto, se cumple que

$$|g(x) - \bar{x}| = |g(x) - g(\bar{x})| = \left|g'(\xi)\right| |x - \bar{x}| \leq k\,|x - \bar{x}| < |x - \bar{x}| .$$

Como $x \in \bar{x}-\delta;\bar{x}+\delta]$, podemos deducir que $|x-\bar{x}| < \delta$ y que $|g(x)-\bar{x}| < \delta$. Este último resultado nos muestra que $g : [\bar{x}-\delta;\bar{x}+\delta] \to [\bar{x}-\delta;\bar{x}+\delta]$.

En consecuencia, la función $g(x) = x - f(x)/f'(x)$ satisface todas las hipótesis del teorema 3.5, de modo que la sucesión $\{x_n\}_{n=1}^{\infty}$ definida por

$$x_n = g(x_{n-1}) = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \text{ para } n \geq 1,$$

converge a $\bar{x}$ para cualquier $x_0 \in [\bar{x}-\delta;\bar{x}+\delta]$.

Como vimos, este método es una variante del método de las aproximaciones sucesivas. Si la función $f(x)$ no tiene derivada en el entorno $[a;b]$ no es posible aplicarlo, pero si resulta difícil calcularla o evaluarla, existe un método alternativo denominado *método de la secante*, el cual reemplaza $f'(x_{n-1})$ por su aproximación discreta, es decir,

$$f'(x_{n-1}) = \frac{f(x_{n-1}) - f(x_{n-2})}{x_{n-1} - x_{n-2}}.$$

Si reemplazamos esto último en la fórmula de Newton-Raphson tenemos

$$x_n = x_{n-1} - \frac{f(x_{n-1})(x_{n-1} - x_{n-2})}{f(x_{n-1}) - f(x_{n-2})},$$

que también podemos escribir como

$$x_n = \frac{f(x_{n-1})x_{n-2} - f(x_{n-2})x_{n-1}}{f(x_{n-1}) - f(x_{n-2})}.$$

3.6. Análisis del error

En este punto analizaremos la convergencia de los métodos iterativos vistos. Nos basaremos en la siguiente definición.

Definición 3.1. Un sucesión $\{x_n\}_{n=0}^{\infty}$ convergirá a $\bar{x}$ de orden α con una constante asintótica λ si se cumple que

$$\lim_{n\to\infty} \frac{|x_{n+1} - \bar{x}|}{|x_n - \bar{x}|^{\alpha}} = \lambda,$$

con $x_n \neq \bar{x}$ para toda n, y α y λ son dos constantes positivas.

En consecuencia, tenemos que la convergencia puede ser **lineal** ($\alpha = 1$), **cuadrática** ($\alpha = 2$), **cúbica** ($\alpha = 3$), etc. Dado que obtener un procedimiento con convergencia mayor a la cuadrática no es sencillo, nos ocuparemos de analizar solamente los dos primeros casos.

Enunciaremos dos teoremas que se refieren a la convergencia lineal y a la cuadrática, que están basados en el método de las aproximaciones sucesivas.

Teorema 3.7. *(Convergencia lineal.)* Sea $g \in \mathrm{C}[a;b]$ tal que $g \in [a;b]$ para toda $x \in [a;b]$. Si g' es continua en $(a;b)$ y existe una constante $k < 1$ tal que

$$\left|g'(x)\right| \leq k, \text{ para todo } x \in (a;b),$$

y si $g'(\bar{x}) \neq 0$, entonces para cualquier $x_0 \in [a;b]$ la sucesión

$$x_n = g(x_{n-1}), \text{ para } n \geq 1,$$

converge sólo linealmente al punto fijo $\bar{x} \in [a;b]$.

Teorema 3.8. *(Convergencia cuadrática.)* Sea $\bar{x}$ la solución de la ecuación $x = g(x)$. Si $g'(\bar{x}) = 0$ y g'' es continua y está estrictamente acotada por una constante M en un intervalo abierto I que contiene a $\bar{x}$, entonces existirá un $\delta > 0$ tal que, para $x_0 \in [\bar{x} - \delta; \bar{x} + \delta]$, la sucesión definida por $x_n = g(x_{n-1})$ cuando $n \geq 1$, converge al menos cuadráticamentea $\bar{x}$. Además para valores suficientemente grandes de n, se tiene

$$|x_{n+1} - \bar{x}| < \frac{M}{2} |x_n - \bar{x}|^2 .$$

Las demostraciones de ambos teoremas pueden verse en [1].

El primer teorema nos dice que para que la convergencia sea cuadrática o superior, se debe cumplir que $g'(\bar{x}) = 0$, en tanto que el segundo, nos da las condiciones que aseguran que la convergencia sea al menos cuadrática. Este teorema nos indica que el método de las aproximaciones sucesivas nos puede llevar a desarrollar métodos con orden de convergencia cuadrática o superior. En efecto, si partimos de

$$x_n = g(x_{n-1}),$$

podemos suponer que $g(x)$ se puede escribir como

$$g(x) = x - \phi(x) f(x).$$

De acuerdo con el segundo teorema, para obtener una convergencia al menos cuadrática debemos plantear que $g'(\bar{x}) = 0$. Dado que:

$$g'(x) = 1 - \phi'(x) f(x) - \phi(x) f'(x),$$

entonces

$$g'(\bar{x}) = 1 - \phi(\bar{x}) f'(\bar{x}),$$

pues $f(\bar{x}) = 0$, entonces $g'(\bar{x}) = 0$ si y sólo si $\phi(\bar{x}) = 1/f'(\bar{x})$. Si reemplazamos esto en la función original nos queda

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

que no es otra cosa que el método de Newton-Raphson.

Del segundo teorema, obtenemos además que $M = |g''(x)|$. De ahí que si $|g''(x)| = 0$, la convergencia podría ser cúbica, es decir, si $f''(x)$ se anula en el intervalo, la convergencia será superior a la cuadrática [2].

3.7. Métodos de convergencia acelerada

Si bien hemos visto que el método de Newton-Raphson es de convergencia cuadrática, no siempre es posible utilizarlo. La principal razón es que debemos conocer la derivada de la función. Aunque vimos un método alternativo, el método de la secante, éste no resulta ser un método de convergencia cuadrática. Veremos ahora un procedimiento para obtener convergencia cuadrática a partir de un método linealmente convergente.

Supongamos que tenemos la sucesión $\{x_n\}_{n=0}^{\infty}$ que converge linealmente y que los signos de $x_n - \bar{x}$, $x_{n+1} - \bar{x}$ y $x_{n+2} - \bar{x}$ son iguales y que n es suficientemente grande. Para construir una nueva sucesión $\{\tilde{x}_n\}_{n=0}^{\infty}$ que converja más rápido que la anterior vamos a plantear que

$$\frac{x_{n+1} - \bar{x}}{x_n - \bar{x}} \approx \frac{x_{n+2} - \bar{x}}{x_{n+1} - \bar{x}},$$

con lo cual nos queda

$$(x_{n+1} - \bar{x})^2 \approx (x_{n+2} - \bar{x})(x_n - \bar{x}).$$

Si la desarrollamos nos queda

$$x_{n+1}^2 - 2x_{n+1}\bar{x} + \bar{x}^2 \approx x_{n+2}x_n - (x_{n+2} + x_n)\bar{x} + \bar{x}^2,$$

y

$$(x_{n+2} + x_n - 2x_{n+1})\bar{x} \approx x_{n+2}x_n - x_{n+1}^2.$$

Si despejamos $\bar{x}$ nos queda

$$\bar{x} \approx \frac{x_{n+2}x_n - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n}.$$

Si ahora sumamos y restamos x_n^2 y $2x_nx_{n+1}$ en el numerador, tenemos

$$\begin{aligned}
\bar{x} &\approx \frac{x_n^2 + x_{n+2}x_n - 2x_nx_{n+1} - x_n^2 + 2x_nx_{n+1} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n} \\
&\approx \frac{x_n(x_{n+2} - 2x_{n+1} + x_n) - (x_n^2 - 2x_nx_{n+1} + x_{n+1}^2)}{x_{n+2} - 2x_{n+1} + x_n} \\
&\approx x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n}.
\end{aligned}$$

Si definimos la nueva sucesión $\{\tilde{x}_n\}_{n=0}^{\infty}$ como

$$\tilde{x}_n = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n},$$

obtenemos una técnica denominada **método Δ^2 de Aitken**, que supone que la sucesión $\{\tilde{x}_n\}_{n=0}^{\infty}$ converge más rápidamente a $\bar{x}$ que la sucesión $\{x_n\}_{n=0}^{\infty}$.

La notación Δ asociada a esta técnica está dada por:

[2] Si se desarrolla $g''(x)$ se tiene que la derivada que más incide en la convergencia es $f''(x)$.

Definición 3.2. Dada la sucesión $\{x_n\}_{n=0}^{\infty}$, la *diferencia progresiva* Δx_n está definida por

$$\Delta x_n = x_{n+1} - x_n, \text{ para } n \geq 0.$$

Las potencias más altas $\Delta^k x_n$ se definen por medio de

$$\Delta^k x_n = \Delta(\Delta^{k-1} x_n), \text{ para } k \geq 2.$$

A partir de estas definiciones tenemos que $\Delta^2 x_n$ se expresa como

$$\begin{aligned} \Delta^2 x_n &= \Delta(\Delta^1 x_n) = \Delta(x_{n+1} - x_n) \\ &= \Delta x_{n+1} - \Delta x_n = (x_{n+2} - x_{n+1}) - (x_{n+1} - x_n) \\ &= x_{n+2} - 2x_{n+1} + x_n, \end{aligned}$$

por lo que el método Δ^2 de Aitken puede escribirse como

$$\tilde{x}_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}.$$

Para analizar la convergencia de este método tenemos el siguiente teorema.

Teorema 3.9. Sea la sucesión $\{x_n\}_{n=0}^{\infty}$ que converge linealmente a $\bar{x}$, y que para valores suficientemente grandes de n, se cumpla que $(x_n - \bar{x})(x_{n+1} - \bar{x}) > 0$. Entonces la sucesión $\{\tilde{x}_n\}_{n=0}^{\infty}$ converge a $\bar{x}$ con mayor rapidez que $\{x_n\}_{n=0}^{\infty}$ en el sentido de que

$$\lim_{n\to\infty} \frac{\tilde{x}_n - \bar{x}}{x_n - \bar{x}} = 0.$$

Si aplicamos el método Δ^2 de Aitken a una sucesión cuya convergencia sea lineal, podemos acelerar la convergencia a cuadrática. Podemos entonces desarrollar otros métodos a partir de esta técnica.

3.8. Método de Steffensen

Si aplicamos esta técnica a una sucesión obtenida por el método de las aproximaciones sucesivas tendremos el método conocido como *método de Steffensen*. Este método, en realidad, tiene una leve modificación al método Δ^2 de Aitken.

Al aplicar el método Δ^2 de Aitken a una sucesión linealmente convergente, la nueva sucesión convergente cuadráticamente se construye mediante los siguientes términos:

$$x_0;\ x_1 = g(x_0);\ x_2 = g(x_1);\ \tilde{x}_0 = \{\Delta^2\}(x_0);\ x_3 = g(x_2);\ \tilde{x}_1 = \{\Delta^2\}(x_1);\ \ldots.$$

En cambio, el método de Steffensen calcula los cuatro primeros términos de la forma indicada pero introduce una leve modificación al calcular el término x_3. La secuencia queda entonces como:

$$x_0^{(0)};\ x_1^{(0)} = g(x_0^{(0)});\ x_2^{(0)} = g(x_1^{(0)});\ x_0^{(1)} = \{\Delta^2\}(x_0^{(0)});$$
$$x_1^{(1)} = g(x_0^{(1)});\ x_2^{(1)} = g(x_1^{(1)});\ x_0^{(2)} = \{\Delta^2\}(x_0^{(1)});\ \ldots$$

De esta manera, el método se asegura una convergencia cuadrática y mejora notablemente la precisión en los resultados obtenidos por el método de las aproximaciones sucesivas. En el siguiente ejemplo podemos ver la diferencia en la convergencia.

Supongamos que para aplicar el método de las aproximaciones sucesivas tenemos la expresión

$$x_{k+1} = \frac{2 - e^{x_k} + x_k^2}{3}, x_0 = 0,50.$$

Tabla 3.1: Método de Steffensen

$\mathbf{i}$	$\mathbf{x_i}$	$\mathbf{k}$	$\mathbf{i}$	$\mathbf{x_i^{(k)}}$
0	0,50000	0	0	0,50000
1	0,20043		1	0,20043
2	0,27275		2	0,27275
3	0,25361	1	0	0,25868
4	0,25855		1	0,25723
5	0,25727		2	0,25761
6	0,25760	2	0	0,25753
7	0,25751			
8	0,25753			

Para ver la eficacia del método y poder comparar, obtendremos la raíz por el método de las aproximaciones sucesivas primero, y por el método de Steffensen, después.

En la tabla 3.1 podemos ver los resultados obtenidos al aplicar ambos métodos. En la segunda columna están los obtenidos con aproximaciones sucesivas y en la última, los obtenidos con Steffensen. Observemos que el método de Steffensen alcanzó más rápidamente el resultado «correcto» que el método de las aproximaciones sucesivas. Mientras este último necesitó ocho iteraciones, el de Steffensen requirió solamente seis.

3.9. Notas finales

Hasta aquí hemos visto seis métodos iterativos para obtener las raíces de una ecuación del tipo $f(x) = 0$. Los dos primeros, el de la bisección y el de la posición falsa («regula falsi») son métodos que aseguran la convergencia pero que son muy lentos. Suelen usarse como una primera aproximación cuando no se tiene información más detallada del punto $\bar{x}$, de ahí que son conocidos como *métodos de arranque*. Sirven para acotar el intervalo en el caul se encuentra la raíz buscada. Los otros cuatro, el de las aproximaciones sucesivas, el Newton-Raphson, el de la secante y el de Steffensen son mucho más potentes y en el caso de Newton-Raphson y Steffensen, con una rápidez de convergencia cuadrática. De los cuatro métodos métodos, los más usuales para programar son el de las aproximaciones sucesivas y el de la secante, puesto que son sencillo y no requieren conocer la derivada primera. Es común, además, que cuando no se tiene un intervalo los suficientemente acotado para trabajar con los métodos de refinamiento, se comience con el método de la bisección, y así, disminuir el «costo computacional».

Sin embargo, cuando la ecuación $f(x) = 0$ tiene multiplicidad de ceros (ejemplo, la función $\text{sen}(x)$), ninguno de estos métodos puede distinguir rápidamente esta situación. Es por eso que existen otros métodos para resolver este tipo de problemas (ver [1]).

Capítulo 4

Interpolación de curvas

4.1. Introducción

En este capítulo nos concentraremos en el estudio de los métodos de interpolación de curvas. Es usual que los ingenieros trabajen con datos extraídos de mediciones, relevamientos, ensayos de laboratorio, etc., los cuales no siempre entregan el valor necesitado para el problema que se está tratando de resolver. Un ejemplo típico de interpolación sencilla utilizado por cualquier profesional de la ingeniería es la interpolación lineal en una tabla de datos (por ejemplo, de estadísticas) para obtener un valor entre dos puntos dados. Este tipo de interpolación lineal era muy usado cuando no existían las calculadoras científicas de bolsillo (ni hablar de computadoras) y debían usarse las famosas **Tablas de logaritmos** para obtener logaritmos, senos, cosenos y cualquier otra función trigonométrica o trascendente.

Un ejempolo de interpolación muy interesante es la función «spline» del AutoCAD, que permite dibujar curvas que pasen por puntos determiandos en el dibujo, y que no pocos usuarios no saben usar en forma eficiente.

Otro ejemplo de interpolación más avanzado es la utilización de polinomios interpolantes en la resolución de estructuras cuando se utilizan programas de análisis estructural que aplican el *método de los elementos finitos*. Allí es de fundamental importancia entender los tipos de polinomios que se pueden usar y los datos necesarios para poder obtener estos polinomios.

Puesto que hay muchos métodos y formas de interpolar, nos ocuparemos de los métodos clásicos y veremos algunas mejoras que se han desarrollado a estos métodos. En particular, gracias al artículo de L.N. Trefethen y J. P. Berrut (véase [14]), analizaremos una mejora al método de Lagrange básico, denominada *Interpolación Baricéntrica de Lagrange*.

4.2. Método de Lagrange

Supongamos que tenemos una lista con datos ordenados de a pares como la de la siguiente tabla:

Tabla 4.1: Datos ordenados de a pares

x	**y**
x_0	y_0
x_1	y_1
x_2	y_2
x_3	y_3

Y supongamos que necesitamos conocer el valor de $y(x_A)$ para un x_A entre x_1 y x_2. La forma sencilla de obtener este valor es graficar estos puntos y trazar un segmento de recta que una y_1 e y_2, ubicar x_A en las abscisas y trazar por él una línea recta paralela al eje de ordenadas que corte el segmento ya mencionado. Finalmente, desde este punto, trazamos una línea recta paralela al eje de abscisas hasta cortar el eje de ordenadas, con lo cual hemos obtenido el valor de $y(x_A)$.

Queda muy evidente que este procedimiento es muy engorroso si se quiere hacerlo en forma metódica. Sin embargo, es la forma más sencilla de interpolación polinomial, la interpolación lineal. Efectivamente, si tomamos los dos puntos en cuestión podemos armar una recta mediante el siguiente sistema:

$$\begin{aligned} y_1 &= m\ x_1 + n \\ y_2 &= m\ x_2 + n \end{aligned}$$

Si restamos y_1 a y_2 obtenemos m:

$$y_2 - y_1 = m(x_2 - x_1) \Rightarrow\ m = \frac{y_2 - y_1}{x_2 - x_1}.$$

Si ahora reemplazamos m en la primera ecuación obtenemos n:

$$y_1 = \frac{y_2 - y_1}{x_2 - x_1} x_1 + n\ \Rightarrow\ n = y_1 - \frac{y_2 - y_1}{x_2 - x_1} x_1.$$

Finalmente la ecuación de la recta que pasa por y_1 e y_2 es:

$$y(x) = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1,$$

que también puede escribirse como

$$y(x) = y_1 \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_1}{x_2 - x_1}.$$

Para hallar $y(x_A)$ basta con reemplazar x_A en cualquiera de las expresiones anteriores.

Lo hecho anteriormente es equivalente al procedimiento gráfico. ¿Pero que pasa si queremos usar más de dos puntos? Supongamos que necesitamos usar los cuatro puntos de la tabla 4.1 para interpolar un punto cualquiera entre x_0 y x_3. En ese caso, el polinomio de mayor grado posible es un polinomio cúbico, porque tiene cuatro coeficientes, y se puede expresar así:

$$y(x) = a_0 + a_1\ x + a_2\ x^2 + a_3\ x^3.$$

Si reemplazamos los cuatro puntos en esta ecuación obtenemos el siguiente sistema de ecuaciones lineales:

$$\begin{aligned} y_0 &= a_0 + a_1\ x_0 + a_2\ x_0^2 + a_3\ x_0^3 \\ y_1 &= a_0 + a_1\ x_1 + a_2\ x_1^2 + a_3\ x_1^3 \\ y_2 &= a_0 + a_1\ x_2 + a_2\ x_2^2 + a_3\ x_2^3 \\ y_3 &= a_0 + a_1\ x_3 + a_2\ x_3^2 + a_3\ x_3^3 \end{aligned}$$

Basta con resolver este sistema de ecuaciones lineales para obtener los coeficientes a_i. Analicemos el sistema escribiéndolo en forma matricial:

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{bmatrix}}_{A} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Se trata de una matriz especial, que se conoce como *matriz de VanderMonde*. Tiene la particularidad de ser mal condicionada, por lo que cualquier método que usemos para resolver este sistema puede traernos algún problema.

La interpolación de Lagrange es una forma sencilla sistemática de resolver el sistema de ecuaciones lineales anterior. El polinomio interpolador lo obtenemos síguiendo los pasos descriptos a continuación:

1. Calculamos los $n+1$ polinomios $L_{n;i}(x)$ relacionados cada uno con cada dato x_i, donde n es el grado del polinomio e i indica el punto considerado, mediante la expresión:

$$L_{n;i}(x) = \frac{\prod_{\substack{j=0 \\ j\neq i}}^{n}(x - x_j)}{\prod_{\substack{j=0 \\ j\neq i}}^{n}(x_i - x_j)} = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

   con $i = 0; 1; \ldots; n$, $j = 0; 1; \ldots; n$, y x_i y x_j refieren a los datos a interpolar.Estos polinomios cumplen con la particularidad de que:

$$L_{n;i}(x) = \begin{cases} 1 & \text{si } x = x_i \\ 0 & \text{si } x = x_j conj \neq i. \end{cases}$$

2. El polinomio interpolador lo obtenemos mediante la expresión:

$$P_n(x) = \sum_{i=0}^{n} y_i \; L_{n;i}(x).$$

Por ejemplo, podemos armar una interpolación lineal mediante los polinomios de Lagrange entre los puntos x_1 y x_2. Al aplicar el método obtenemos:

$$\begin{aligned} L_{1;0} &= \frac{x - x_2}{x_1 - x_2} \\ L_{1;1} &= \frac{x - x_1}{x_2 - x_1} \\ P_1(x) &= y_1 \; L_{1;0}(x) + y_2 \; L_{1;1}(x) \\ P_1(x) &= y_1 \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_1}{x_2 - x_1}, \end{aligned}$$

que es la ecuación de la recta que obtuvimos antes.

Para obtener el polinomio de tercer grado tendremos:

$$\begin{aligned} L_{3;0}(x) &= \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} \\ L_{3;1}(x) &= \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} \\ L_{3;2}(x) &= \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} \\ L_{3;3}(x) &= \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \\ P_3(x) &= y_0 L_{3;0}(x) + y_1 L_{3;1}(x) + y_2 L_{3;3}(x) + y_3 L_{3;3}(x) \end{aligned}$$

Como hemos utilizado todos lo puntos de los datos, es evidente que no podemos crear un polinomio de mayor grado que el cúbico. Por lo tanto, existe un sólo polinomio posible de construir con todos los datos disponibles. El siguiente teorema define a este único polinomio.

Teorema 4.1. Sean x_0, x_1, ..., x_n, $n+1$ números diferentes, y sea f una función tal que sus valores se obtengan a partir de los números dados $(f(x_0); f(x_1), \ldots, f(x_n))$, entonces existe un único polinomio $P_n(x)$ de grado n, que cumple con la propiedad

$$f(x_k) = P(x_k) \text{ para cada } k = 0; 1; \ldots; n;$$

y este polinomio está dado por la siguiente expresión

$$P_n(x) = f(x_0)L_{n;0}(x) + f(x_1)L_{n;1}(x) + \ldots + f(x_n)L_{n;n}(x) = \sum_{i=0}^{n} f(x_i)L_{n;i}(x),$$

donde

$$L_{n;i}(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j},$$

para $i = 0; 1; \ldots; n$.

Sin embargo, podemos crear varios polinomios de grados menores a n. Así, con los datos de la tabla 4.1 estamos en condiciones construir al menos tres polinomios de grado 1 y dos polinomios de grado 2. (En realidad, hay más polinomios para ambos grados, pero no siempre son de utilidad práctica.)

Obtenido el polinomio interpolante nos queda un punto por definir: ¿cuál es el error que estamos cometiendo al interpolar mediante un polinomio respecto de la función original? Para ello tenemos el siguiente teorema.

Teorema 4.2. Sean x_0, x_1, x_2,... ,x_n, números distintos en el intervalo $[a; b]$ y sea $f \in \mathrm{C}^{n+1}[a; b]$. Entonces, para cualquier $x \in [a; b]$ existe un número $\xi(x) \in [a; b]$ para que el se cumple que

$$f(x) = P_n(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n}(x - x_i),$$

donde $P_n(x)$ es el máximo polinomio interpolante.

Demostración Si $x = x_i$ para $i = 0; 1; 2; \ldots; n$ entonces $f(x_i) = P_n(x_i)$ y para cualquier $\xi(x_i) \in [a; b]$ se cumple lo expresado en el teorema. En cambio, si $x \neq x_i$ para $i = 0; 1; 2; \ldots; n$, se puede definir la siguiente función $g(u)$ para $u \in [a; b]$:

$$g(u) = f(u) - P_n(u) - [f(x) - P_n(x)] \prod_{i=0}^{n} \frac{(u - x_i)}{(x - x_i)}.$$

Como $f \in \mathrm{C}^{n+1}[a; b]$, $P_n \in \mathrm{C}^{\infty}[a; b]$, y $x \neq x_i$ para cualquier i, entonces $g \in \mathrm{C}^{n+1}[a; b]$. Si $u = x_j$ tendremos que

$$g(x_j) = f(x_j) - P_n(x_j) - [f(x) - P_n(x)] \prod_{i=0}^{n} \frac{(x_j - x_i)}{(x - x_i)} = 0 - [f(x) - P_n(x)]0 = 0.$$

También tenemos que $g(x) = 0$, pues

$$g(x) = f(x) - P_n(x) - [f(x) - P_n(x)] \prod_{i=0}^{n} \frac{(x - x_i)}{(x - x_i)} = f(x) - P_n(x) - [f(x) - P_n(x)] = 0,$$

y en consecuencia, $g \in \mathrm{C}^{n+1}[a; b]$ y se anula para $x; x_0; x_1; \ldots; x_n$, es decir, para $n+2$ números distintos. De acuerdo con el Teorema de Rolle, existe entonces un $\xi \in (a, b)$ tal que $g^{n+1}(\xi) = 0$. Así tendremos que

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - P_n^{(n+1)}(\xi) - [f(x) - P_n(x)] \frac{d^{n+1}}{du^{n+1}} \left[\prod_{i=0}^{n} \frac{(u - x_i)}{(x - x_i)} \right]_{u=\xi}.$$

Como $P_n(u)$ es un polinomio de grado n, entonces $P_n^{(n+1)}(u) = 0$. A su vez, $\prod_{i=0}^{n} \frac{(u-x_i)}{(x-x_i)}$ es un polinomio de grado $n+1$, entonces su derivada de orden $n+1$ será

$$\frac{d^{n+1}}{du^{n+1}} \left[\prod_{i=0}^{n} \frac{(u-x_i)}{(x-x_i)} \right] = \frac{(n+1)!}{\prod_{i=0}^{n}(x-x_i)}.$$

Si reempazamos, tendremos que

$$0 = f^{(n+1)}(\xi) - 0 - [f(x) - P_n(x)] \frac{(n+1)!}{\prod_{i=0}^{n}(x-x_i)}.$$

Al despejar $f(x)$ de la ecuación anterior nos queda

$$f(x) = P_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n}(x-x_i).$$

Desde el punto de vista teórico, esta expresión del error es muy importante porque muchas de las técnicas de derivación e integración numérica se derivan de aplicar la interpolación por el método de Lagrange. Sin embargo, para otros casos, no debemos olvidarnos que no conocemos $f(x)$ (y por lo tanto, tampoco $f^{(n+1)}(x)$), por lo que el error calculado es sólo una aproximación.

Finalmente, podemos ver que el método tiene algunas desventajas:

1. Cada evaluación del polinomio $P_n(x)$ requiere $O(n^2)$ operaciones aritméticas.
2. Agregar un par de datos $x_{n+1}, f(x_{n+1})$ requiere rehacer todos lo polinomios $L_{n,i}(x)$.
3. Es numéricamente inestable.

4.3. Método de Newton

Una forma alternativa de plantear la construcción del polinomio interpolador es la siguiente. Supongamos que queremos usar solamente los primeros tres puntos de nuestra tabla. Entonces planteemos el siguiente sistema de ecuaciones:

$$\begin{aligned} y_0 &= a_0 + a_1\ x_0 + a_2\ x_0^2 \\ y_1 &= a_0 + a_1\ x_1 + a_2\ x_1^2 \\ y_2 &= a_0 + a_1\ x_2 + a_2\ x_2^2. \end{aligned}$$

Al eliminar a_0 tenemos este nuevo sistema

$$\begin{aligned} y_1 - y_0 &= a_1(x_1 - x_0) + a_2(x_1^2 - x_0^2) \\ y_2 - y_1 &= a_1(x_2 - x_1) + a_2(x_2^2 - x_1^2), \end{aligned}$$

que puede escrbirse como

$$\begin{aligned} \frac{y_1 - y_0}{x_1 - x_0} &= a_1 + a_2(x_1 + x_0) \\ \frac{y_2 - y_1}{x_2 - x_1} &= a_1 + a_2(x_2 + x_1). \end{aligned}$$

Si ahora eliminamos a_1 obtenemos el coeficiente a_2 que resulta ser

$$\begin{aligned} a_2(x_2 - x_0) &= \frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0} \\ a_2 &= \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}. \end{aligned}$$

Ahora reemplacemos a_2 en una de las ecuaciones anteriores para obtener a_1

$$\begin{aligned} \frac{y_1 - y_0}{x_1 - x_0} &= a_1 + \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}(x_1 + x_0) \\ a_1 &= \frac{y_1 - y_0}{x_1 - x_0} - \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}(x_1 + x_0). \end{aligned}$$

Ahora reemplacemos a_1 y a_2 en la primera ecuación de todas para obtener a_0:

$$\begin{aligned} y_0 &= a_0 + \left[\frac{y_1 - y_0}{x_1 - x_0} - \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}(x_1 + x_0)\right] x_0 + \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} x_0^2 \\ a_0 &= y_0 - \left(\frac{y_1 - y_0}{x_1 - x_0} - \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} x_1\right) x_0. \end{aligned}$$

Armemos finalmente el polinomio interpolante reemplazando a_0, a_1 y a_2

$$\begin{aligned} P(x) &= y_0 - \left(\frac{y_1 - y_0}{x_1 - x_0} - \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} x_1\right) x_0 + \\ &\quad + \left[\frac{y_1 - y_0}{x_1 - x_0} - \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}(x_1 + x_0)\right] x + \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0} x^2 \\ &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}\left[x^2 - (x_0 + x_1)x + x_0 x_1\right] \\ &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}(x - x_0)(x - x_1). \end{aligned}$$

Esta forma de armar el polinomio se denomina *método de las diferencias divididas de Newton*, y podemos sistematizarla para que sea muy sencillo de realizar. En primer lugar, podemos decir que $f(x_i) = y_i$. Seguidamente vamos a definir que:

$$\begin{aligned} f(x_0; x_1) &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ f(x_1; x_2) &= \frac{f(x_2) - f(x_1)}{x_2 - x_1}, \end{aligned}$$

y generalizando

$$f(x_i; x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

Análogamente tenemos que:

$$f(x_0; x_1; x_2) = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0} =, \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0},$$

y si generalizamos nuevamente tenemos

$$f(x_i; x_{i+1}; x_{i+2}) = \frac{f(x_{i+1}; x_{i+2}) - f(x_i; x_{i+1})}{x_{i+2} - x_i}.$$

Finalmente podemos generalizar totalmente las expresiones anteriores a la siguiente expresión:

$$f(x_k; x_{k+1}; \ldots; x_{n-1}; x_n) = \frac{f(x_{k+1}; x_{k+2}; \ldots; x_n) - f(x_k; x_{k+1}; \ldots; x_{n-1})}{x_n - x_k}.$$

Si utilizamos esta notación para el polinomio que hallamos más arriba nos queda:

$$P(x) = f(x_0) + f(x_0; x_1) \cdot (x - x_0) + f(x_0; x_1; x_2) \cdot (x - x_0) \cdot (x - x_1).$$

Esta forma nos permite agregar un punto más y aumentar el grado del polinomio en forma sencilla. Efectivamente, si queremos agregar x_3, solamente debemos agregar al polinomio anterior el término $f(x_0; x_1; x_2; x_3)(x - x_0)(x - x_1)(x - x_2)$, con lo cual nos queda

$$\begin{aligned} P(x) &= f(x_0) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \\ &\quad + f(x_0; x_1; x_2; x_3)(x - x_0)(x - x_1)(x - x_2). \end{aligned}$$

Esta forma de armar los polinomios facilita notablemente el aumentar la cantidad de puntos para obtener un polinomio interpolante, pues permite usar el polinomio anterior. En la tabla siguiente se puede ver un esquema de cómo operar.

Tabla 4.2: Método de Newton

$\mathbf{x}$	$\mathbf{f(x)}$	$\mathbf{f(x_i; x_{i+1})}$	$\mathbf{f(x_i; x_{i+1}; x_{i+2})}$	$\mathbf{f(x_i; x_{i+1}; x_{i+2}; x_{i+3})}$
x_0	$f(x_0)$			
		$f(x_0; x_1)$		
x_1	$f(x_1)$		$f(x_0; x_1; x_2)$	
		$f(x_1; x_2)$		$f(x_0; x_1; x_2; x_3)$
x_2	$f(x_2)$		$f(x_1; x_2; x_3)$	
		$f(x_2; x_3)$		
x_3	$f(x_3)$			

Observemos que podemos armar dos polinomios a partir del método de Newton. Uno es el que obtuvimos antes, por el denominado «método de la diferencias divididas *progresivas*». El otro podemos obtenerlo partiendo de x_3, que resulta ser

$$\begin{aligned} P(x) &= f(x_3) + f(x_2; x_3)(x - x_3) + f(x_1; x_2; x_3)(x - x_3)(x - x_2) + \\ &\quad + f(x_0; x_1; x_2; x_3)(x - x_3)(x - x_2)(x - x_1), \end{aligned}$$

que se denomina «método de las diferencias divididas *regresivas*».

El método de Newton, en sus dos variantes, es muy usado cuando se trabaja con datos que pueden ser modificados (aumentando la cantidad de puntos disponibles para la interpolación) y, en consecuencia, aplicar el método de Lagrange se vuelve muy engorroso. Otra ventaja es que para evaluar los polinomios $P_n(x)$ requerimos n operaciones aritméticas, algo bastante menor al $O(n^2)$ que requiere el método de Lagrange [1]. Sin embargo, el método exige que los datos deban estar ordenados, según x_i, en forma ascendente (o descendente) para poder implementarlo. Si agregamos algún dato intermedio, la ventaja anterior se pierde porque la tabla 4.2 debe rehacerse, perdiendo practicidad.

Para mejorar esto existe una variante del método de Lagrange que nos permite interpolar de manera sencilla y al que resulta muy fácil agregarle puntos en cualquier orden.

[1] De todos modos, se requieren $O(n^2)$ operaciones para obtener los coeficientes $f(x_k; x_{k+1}; \ldots; x_n)$.

4.4. Interpolación baricéntrica de Lagrange

Supongamos que definimos un polinomio genérico $L(x)$ tal que

$$L(x) = (x - x_0)(x - x_1)\dots(x - x_n).$$

Definamos además los pesos baricéntricos como

$$w_i = \prod_{\substack{k=0 \\ k\neq i}}^{n} \frac{1}{x_i - x_k}, \text{ para todo } i = 0; 1; \dots; n.$$

Entonces podemos escribir cualquier polinomio de Lagrange como

$$L_{n;i} = L(x)\frac{w_i}{x - x_i},$$

y, en consecuencia, el polinomio interpolante será

$$P_n(x) = \sum_{i=0}^{n} f(x_i)\frac{L(x)w_i}{x - x_i} = L(x)\sum_{i=0}^{n} f(x_i)\frac{w_i}{x - x_i},$$

pues $L(x)$ es constante para todo los términos de la sumatoria.

Esto es una gran ventaja en dos sentidos. Primero, para evaluar $P_n(x)$ se necesitan sólo $O(n)$ operaciones, lo cual hace mucho más rápido el procedimiento. Y segundo, si agregamos el par de datos $x_{n+1}, f(x_{n+1})$, sólo debemos hacer lo siguiente:

- Dividir cada w_i por $x_i - x_{n+1}$.
- Calcular un nuevo w_{i+1}.

En ambos casos el costo computacional es de $n + 1$ operaciones. Es decir, ¡podemos actualizar el polinomio $P_n(x)$ con sólo $O(n)$ operaciones! A esta variante del método de Lagrange suele llamársela *método mejorado de Lagrange* y tiene una ventaja adicional respecto al método de Newton que rara vez se menciona: los coeficientes w_i no dependen de los datos $f(x_{n+1})$. Esto permite que podamos interpolar varias funciones con el mismo polinomio. Y mantiene, además, la ventaja de no necesitar ordenar los datos, como sí requiere el método de Newton.

Pero todavía no hemos terminado. Supongamos ahora que interpolamos la constante 1 con el polinomio hallado. En ese caso tenemos

$$1 = \sum_{i=0}^{n} 1 \cdot L_{n,i}(x) = L(x)\sum_{i=0}^{n} \frac{w_i}{x - x_i},$$

pues hemos visto que $L_{n,i}(x) = 1$ cuando $x = x_i$.

Si dividimos $P_n(x)$ por la expresión anterior, o sea, que la dividimos por 1, nos queda:

$$P_n(x) = \frac{L(x)\displaystyle\sum_{i=0}^{n} f(x_i)\frac{w_i}{x - x_i}}{L(x)\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}},$$

y simplificando $L(x)$, obtenemos que

$$P_n(x) = \frac{\displaystyle\sum_{i=0}^{n} f(x_i)\frac{w_i}{x - x_i}}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}},$$

que se denomina *interpolación baricéntrica de Lagrange*. Al igual que en el caso del método mejorado, sólo se necesitan $O(n)$ operaciones para actualizar el polinomio si agregamos un par de datos $x_{n+1}, f(x_{n+1})$ adicionales.

De todos modos, si la interpolación la realizamos con puntos uniformemente distanciados o distribuidos unos de otros, la mala condición del problema no se puede evitar (pues ningún algoritmo la mejora). La consecuencia directa de esto es el llamado *fenómeno de Runge*, que se da cuando aparecen oscilaciones no deseadas en ambos extremos del intervalo a interpolar (como puede verse en [1] o [4]). A pesar de esto, en general, la interpolación baricéntrica de Lagrange es más estable numéricamente que el método de Lagrange original y que el método de Newton, según el análisis hecho por N.J. Higham en [7]. Pero para evitar el fenómeno descripto, debemos cambiar la forma de resolver nuestro problema. Para ello tenemos otra forma de interpolar mediante polinomios.

4.5. Interpolación de Hermite

Muchas veces disponemos de más datos para interpolar. Por ejemplo, supongamos que para una partícula que se desplaza conocemos los siguientes datos: el instante t_i, la coordenada de la trayectoria, y_i y la velocidad v_i, para $i = 0; 1; \ldots; n$. En este caso además de los valores de $f(t_i)$ conocemos también los de $f'(t_i)$ pues $v_i = f'(t_i)$. Por lo tanto nuestra tabla original podría ser reescrita como (tabla 4.3):

Tabla 4.3: Datos incluyendo la primera derivada

t	**y**	**v**
t_0	y_0	v_0
t_1	y_1	v_1
t_2	y_2	v_2
t_3	y_3	v_3

Ahora contamos con más información para construir nuestro polinomio interpolante. En efecto, de disponer de sólo cuatro valores asociados a nuestros puntos (en este caso, el instante t_i), pasamos a tener ocho valores. Si queremos utilizar todos los datos disponibles, en lugar de interpolar con una curva de tercer grado, podemos usar ahora una curva de grado 7, pues este polinimio tiene ocho coeficientes, a saber:

$$y(t) = a_0 + a_1\ t + a_2\ t^2 + a_3\ t^3 + a_4\ t^4 + a_5\ t^5 + a_6\ t^6 + a_7\ t^7,$$

del cual podemos hallar la primera derivada, que resulta ser

$$v(t) = y'(t) = a_1 + 2a_2\ t + 3a_3\ t^2 + 4a_4\ t^3 + 5a_5\ t^4 + 6a_6\ t^5 + 7a_7\ t^6.$$

Al igual que al principio, podemos reemplazar cada uno de los valores en las dos funciones, con lo cual obtendremos un sistema de ocho ecuaciones con ocho incógnitas, sistema que puede resolverse sin problemas. Cuando conocemos el valor de la función en el punto como así también su derivada, la interpolación se denomina *Interpolación de Hermite*. El siguiente teorema define la interpolación de Hermite.

Teorema 4.3. Sea $f \in \mathrm{C}^1[a; b]$ y sean $x_0; x_1; \ldots; x_n \in [a; b]$ distintos, el polinomio único de menor grado que concuerda con f y f' en $x_0; x_1; \ldots; x_n$ es el polinomio de Hermite de grado a lo sumo $2n + 1$, que está dado por la siguiente expresión:

$$H_{2n+1}(x) = \sum_{i=0}^{n} f(x_i) H_{n;i}(x) + \sum_{i=0}^{n} f'(x_i) \hat{H}_{n;i}(x),$$

donde

$$H_{n;i}(x) = [1 - 2(x - x_i)L'_{n;i}(x_i)]L^2_{n;i}(x),$$

y

$$\hat{H}_{n;i}(x) = (x - x_i)L^2_{n;i}(x),$$

donde $L_{n;i}(x)$ es el $i-$ésimo polinomio de Lagrange de grado n. Si además $f \in \mathrm{C}^{2n+2}[a;b]$, entonces se cumple que

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \dots (x - x_n)^2}{(2n+2)!} f^{(2n+2)}(\xi),$$

con ξ tal que $a < \xi < b$.

Demostración Primero, recordemos que

$$L_{n;i}(x) = \begin{cases} 1 & \text{si } x = x_i \\ 0 & \text{si } x = x_j conj \neq i. \end{cases},$$

por lo tanto, tenemos que:

$$H_{n,i}(x_j) = 0 \wedge \hat{H}_{n,i}(x_j) = 0,$$

para $j \neq i$, en tanto que

$$H_{n,i}(x_i) = [1 - 2(x_i - x_i)L'_{n;i}(x_i)]L^2_{n;i}(x_i) = [1 - 2(0)L'_{n;i}(x_i)] \cdot 1^2 = 1,$$

y

$$\hat{H}_{n,i}(x_i) = (x_i - x_i)L^2_{n;i}(x_i) = (x_i - x_i) \cdot 1^2 = 0.$$

Entonces, nos queda que:

$$H_{2n+1}(x_i) = \sum_{i=0}^{n} f(x_i)H_{n;i}(x_i) + \sum_{i=0}^{n} f'(x_i)\hat{H}_{n;i}(x_i) = f(x_i) + \sum_{i=0}^{n} f'(x_i) \cdot 0 = f(x_i),$$

para $i = 0; 1; 2; \dots; x_n$, es decir $H_{2n+1}(x) = f(x)$ en los puntos dados.

Demostremos ahora que $H'_{2n+1}(x) = f'(x)$. Como $L_{n;i}(x)$ es un factor de $H'_{n;i}(x)$, entonces se cumple que $H'_{n;i}(x_j) = 0$ cuando $j \neq i$. Si $j = i$, tenemos que

$$\begin{aligned} H'_{n;j}(x_j) &= -2 \cdot L^2_{n;j}(x_j) + [1 + 2(x_j - x_j)L'_{n;j}(x_j)]2L_{n;j}(x_j)L'_{n;j}(x_j) \\ &= -2 \cdot L^2_{n;j}(x_j) + 2 \cdot L^2_{n;j}(x_j) = 0, \end{aligned}$$

o sea, $H'_{n;i}(x_j) = 0$ para todas la j e i.

Por otro lado, observemos que

$$\begin{aligned} \hat{H}'_{n;i}(x_j) &= L^2_{n;i}(x_j) + (x_j - x_i)2L_{n;i}(x_j)L'_{n;i}(x_j) \\ &= L_{n;i}(x_j)[L_{n;i}(x_j) + 2(x_j - x_i)L'_{n;i}(x_j)], \end{aligned}$$

y en consecuencia, cuando $j \neq i$ tendremos que:

$$\hat{H}'_{n;i}(x_j) = L^2_{n;i}(x_j) + (x_j - x_i)2L_{n;i}(x_j)L'_{n;i}(x_j) = 0 + 0 = 0,$$

pues $L_{n;i}(x_j) = 0$, y cuando $j = i$

$$\hat{H}'_{n;j}(x_j) = L^2_{n;j}(x_j) + (x_j - x_j)2L_{n;j}(x_j)L'_{n;j}(x_j) = 1^2 + 0 = 1.$$

Si combinamos ambos casos tenemos

$$\begin{aligned} H'_{2n+1}(x_j) &= \sum_{i=0}^{n} f(x_j)H'_{n;i}(x_j) + \sum_{i=0}^{n} f'(x_j)\hat{H}'_{n;i}(x_j) \\ &= \sum_{i=0}^{n} f(x_j)\cdot 0 + f'(x_j)\hat{H}'_{n;j}(x_j) = 0 + f'(x_j) = f'(x_j), \end{aligned}$$

entonces $H_{2n+1}(x) = f(x)$ y $H'_{2n+1}(x) = f'(x)$ para $x_0; x_1; \ldots; x_n$.

En realidad, la interpolación de Hermite es un caso particular de los denominados *polinomios osculantes*, cuando $m_i = 1$. Veamos la siguiente definición.

Definición 4.1. Dados $x_0; x_1; \ldots; x_n$, todos distintos y los enteros no negativos $m_0; m_1; \ldots; m_n$, se denomina *polinomio osculante* que aproxima una función $f \in \mathrm{C}^m[a,b]$ donde se cumple que $m = \max\{m_0; m_1; \ldots; m_n\}$ y $x_i[a;b]$ para cada $i = 0; 1; \ldots; n$, al polinomio de menor grado que concuerda con la función f y con todas sus derivadas de orden menor o igual m_i en x_i para cada $i = 0; 1; \ldots; n$. El máximo grado de este polinimio es

$$M = \sum_{i=0}^{n} m_i \ + \ n,$$

pues el número de condiciones que debe cumplir es

$$\sum_{i=0}^{n}(m_i + 1) = \sum_{i=0}^{n} m_i + (n+1),$$

y un polinomio de grado M tiene $M+1$ coeficientes.

Esto quiere decir que además de las derivadas primeras podemos tener las derivadas segundas, terceras, etc., para armar el polinomio interpolante. Con esos datos (inclusive puede ocurrir que contemos con datos parciales de las derivadas), el procedimiento visto para la interpolación de Hermite se puede ampliar para obtener curvas que tengan segundas o terceras derivadas. Sin embargo, como el método está basado en los polinomios de Lagrange, si se agregan datos, el método es bastante engorroso, porque deben repetirse todos los cálculos para obtener el nuevo polinomio interpolante. Tal como vimos en el caso anterior, existe una forma alternativa de armar el polinomio aplicando el método de Newton, que nos permite desarrollarlo con la siguiente fórmula:

$$P_n(x) = f(x_0) + \sum_{k}^{n} f(x_0; x_1; \ldots; x_k) \prod_{j=0}^{k-1}(x - x_j).$$

Dado que conocemos los valores de la derivada primera, debemos redefinir nuestra sucesión de datos. Por ejemplo, si tomamos los datos de la tabla 4.3, nuestra nueva sucesión de puntos es $t_0; t_0; t_1; t_1; t_2; t_2; t_3; t_3$, es decir, definimos una nueva sucesión $z_0; z_1; \ldots; z_{2n+1}$ tal que

$$z_{2i} = z_{2i+1} = t_i,$$

con $i = 0; 1; 2; \ldots; n$. Como con esta nueva sucesión no podemos definir $f(z_{2i}; z_{2i+1})$ de la forma vista anteriormente, resulta conveniente definirla aprovechando que conocemos $f'(z_{2i}) = f'(x_i)$, con lo que aprovechamos los datos conocidos. En consecuencia, podemos construir la tabla 4.4 con los coeficientes para armar el polinomio según el método de Newton.

Construida nuestra tabla, el polinomio de Hermite se arma de la siguiente manera:

$$H_{2n+1}(x) = f(z_0) + \sum_{k=1}^{2n+1} \left[f(z_0; z_1; \ldots; z_k) \prod_{j=0}^{k-1}(x - z_j) \right].$$

Tabla 4.4: Interpolación Hermite aplicando el Método de Newton

$\mathbf{z}$	$\mathbf{f(z)}$	$\mathbf{f(z_i; z_{i+1})}$	$\mathbf{f(z_i; z_{i+1}; z_{i+2})}$	$\mathbf{f(z_i; z_{i+1}; z_{i+2}; z_{i+3})}$
$z_0 = x_0$	$f(z_0) = f(x_0)$			
		$f(z_0; z_1) = f'(x_0)$		
$z_1 = x_0$	$f(z_1) = f(x_0)$		$f(z_0; z_1; z_2)$	
		$f(z_1; z_2)$		$f(z_0; z_1; z_2; z_3)$
$z_2 = x_1$	$f(z_2) = f(x_1)$		$f(z_1; z_2; z_3)$	
		$f(z_2; z_3) = f'(x_1)$		$f(z_1; z_2; z_3; z_4)$
$z_3 = x_1$	$f(z_3) = f(x_1)$		$f(z_2; z_3; z_4)$	
		$f(z_3; z_4)$		$f(z_2; z_3; z_4; z_5)$
$z_4 = x_2$	$f(z_4) = f(x_2)$		$f(z_3; z_4; z_5)$	
		$f(z_4; z_5) = f'(x_2)$		$f(z_3; z_4; z_5; z_6)$
$z_5 = x_2$	$f(z_5) = f(x_2)$		$f(z_4; z_5; z_6)$	
		$f(z_5; z_6)$		$f(z_4; z_5; z_6; z_7)$
$z_6 = x_3$	$f(z_6) = f(x_3)$		$f(z_5; z_6; z_7)$	
		$f(z_6; z_7) = f'(x_3)$		
$z_7 = x_3$	$f(z_7) = f(x_3)$			

Si aplicamos esto a nuestros datos originales de la tabla 4.3, obtendríamos un polinomio de grado 7. Pero este polinomio puede sufrir los mismos problemas que ya vimos para los polinomios de Lagrange, es decir, oscilaciones no deseadas en los extremos del intervalo de interpolación, si la distribución de los puntos es uniforme. De ahí que el método de Hermite no suele usarse de esta forma, sino como parte de una interpolación por segmentos. Así, para cada intervalo entre puntos tenemos cuatro datos que podemos utilizar para interpolar valores entre $x_i; x_{i+1}$. Veamos como aplicarlo a nuestra tabla 4.3.

Para armar la curva que interpola entre t_0 y t_1, contamos con los valores de y_0, y_1, v_0 y v_1, con lo cual podemos armar un polinomio de Hermite de tercer grado que cumpla con las condiciones $H_3(t_0) = f(t_0) = y_0$; $H_3(t_1) = f(t_1) = y_1$, $H_3'(t_0) = f'(t_0) = v_0$ y $H_3'(t_1) = f'(t_1) = v_1$. Lo mismo podemos hacer entre t_1 y t_2, y para el intervalo t_2 y t_3. Tendremos, entonces, cuatro polinomios de Hermite para todo el intervalo, a saber, $H_{1;0}(t)$, $H_{1;1}(t)$, $\hat{H}_{1;0}(t)$ y $\hat{H}_{1;1}(t)$. Los polinomios resultantes son:

$$
\begin{aligned}
H_{1;0}(t) &= \left[1 - 2(x - x_0)\frac{1}{x_0 - x_1}\right]\left(\frac{x - x_1}{x_0 - x_1}\right)^2 \\
H_{1;1}(t) &= \left[1 - 2(x - x_1)\frac{1}{x_1 - x_0}\right]\left(\frac{x - x_0}{x_1 - x_0}\right)^2 \\
\hat{H}_{1;0}(t) &= (x - x_0)\left(\frac{x - x_1}{x_0 - x_1}\right)^2 \\
\hat{H}_{1;1}(t) &= (x - x_1)\left(\frac{x - x_0}{x_1 - x_0}\right)^2
\end{aligned}
$$

Como además se cumple que $H_{3;i}(t_{i+1}) = H_{3;i+1}(t_{i+1})$ y $H'_{3;i}(t_{i+1}) = H'_{3;i+1}(t_{i+1})$, tenemos continuidad para la curva y su primera derivada. Podemos armar una curva con segmentos de curvas de tercer grado, que puede representar a la función y a la primera derivada, sin tener que preocuparnos por los efectos negativos de las oscilaciones no deseadas en los extremos. Este método se usa en el *método de los elementos finitos* para armar las funciones de forma en los elementos de viga.

De todos modos, como para poder armar este tipo de curvas debemos conocer los valores de las derivadas en cada punto, algo que no siempre es posible, usar estos segmentos de curvas con polinomios de Hermite no siempre resultan ser una solución aplicable. De ahí que existe otra manera de obtener curvas con estas características.

4.6. Interpolación por «splines»

Supongamos que en lugar de proponer interpolar los datos de la tabla 4.1 mediante un solo polinomio que pase por todos los puntos, lo hagamos mediante segmentos de curvas, en este caso con polinimios de tercer grado, denominados *trazadores cúbicos*, similares al caso de la interpolación por segmentos de polinomios de Hermite. Definamos las curvas como

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3,$$

con $i = 0; 1; \ldots; n-1$. Como en el caso anterior, observemos que tenemos cuatro constantes para cada polinomio, por lo tanto, debemos agregar condiciones para poder armar nuestra interpolación. Como no conocemos los valores de las derivadas en los puntos, vamos a imponer que las curvas cumplan con estas condiciones:

1. $S_i(x_i) = f(x_i)$ para cada $i = 0; 1; \ldots; n$;
2. $S_{i+1}(x_{i+1}) = S_i(x_{i+1})$ para cada $i = 0; 1; \ldots; n-2$;
3. $S'_{i+1}(x_{i+1}) = S'_i(x_{i+1})$ para cada $i = 0; 1; \ldots; n-2$;
4. $S''_{i+1}(x_{i+1}) = S''_i(x_{i+1})$ para cada $i = 0; 1; \ldots; n-2$;
5. Alguna de las siguiente condiciones de borde:
   a) $S''_0(x_0) = S''_{n-1}(x_n) = S''_n(x_n) = 0$ (frontera libre);
   b) $S'_0(x_0) = f'(x_0) = \alpha$ y $S'_{n-1}(x_n) = S'_n(x_n) = f'(x_n) = \beta$ (frontera sujeta).

La primera condición nos asegura que las curvas pasen por los datos, en tanto que las tres condiciones siguientes aseguran la continuidad del conjunto de curvas tanto para las funciones S_i como para sus derivadas primera y segunda.

Para obtener cada polinomio, empecemos por plantear las condiciones definidas arriba. En primer lugar, como $S_i(x_i) = f(x_i)$, tendremos que:

$$S_i(x_i) = a_i = f(x_i).$$

Al aplicar la segunda condición tenemos que:

$$a_{i+1} = S_{i+1}(x_{i+1}) = S_i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3,$$

para cada $i = 0; 1; \ldots; n-2$. Para simplificar la notación definamos que $h_i = (x_{i+1} - x_i)$, y que $a_n = f(x_n)$. Entonces nos queda que

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3,$$

es válida para cada $i = 0; 1; \ldots; n-1$.

En forma análoga tenemos que

$$S'_i(x_i) = b_i,$$

por lo tanto, también se cumple que

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2,$$

es válida para cada $i = 0; 1; \ldots; n-1$.

Finalmente, tenemos que

$$S_i''(x_i) = 2c_i.$$

Como se cumple que $c_n = S_n''(x_n)/2$, nos queda que:

$$c_{i+1} = c_i + 3d_i h_i,$$

una vez más, para cada $i = 0; 1; \ldots; n-1$. Si despejamos d_i y reemplazamos en las dos expresiones anteriores, nos queda:

$$\begin{aligned} a_{i+1} &= a_i + b_i h_i + \frac{h_i^2}{3}(2c_i + c_{i+1}), \\ b_{i+1} &= b_i + h_i(c_i + c_{i+1}), \end{aligned}$$

para cada $i = 0; 1; \ldots; n-1$.

En la primera ecuación podemos despejar b_i, que resulta ser

$$b_i = \frac{a_{i+1} - a_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}).$$

Si usamos la segunda para obtener b_i en vez de b_{i+1} y utilizamos la expresión que hallamos recién para obtener b_{i-1}, nos queda

$$\begin{aligned} \frac{a_{i+1} - a_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}) &= \frac{a_i - a_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{3}(2c_{i-1} + c_i) + h_{i-1}(c_{i-1} + c_i) \\ h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} &= \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}), \end{aligned}$$

para cada $i = 1; 2; \ldots; n-1$.

Ahora nos falta determinar si con este esquema podemos obtener un resultado único para los valores de c_i. Para ello tenemos el siguiente teorema:

Teorema 4.4. Sea f en $a = x_0 < x_1 < \ldots < x_n = b$, entonces f tendrá un interpolante único de frontera libre o natural en los nodos $x_0; x_1; \ldots; x_n$.

Demostración Si la curva es de frontera libre o natural, entonces se cumple que $S_0''(a) = 0$ y $S_{n-1}''(b) = S_n''(b) = 0$, por lo tanto tendremos que

$$c_n = \frac{S_n''(x_n)}{2} = 0;$$

y que

$$0 = S_0''(x_0) = 2c_0 + 6d_0(x_0 - x_0) \Rightarrow \ c_0 = 0.$$

En consecuencia, nos queda un sistema de ecuaciones de la forma $Ax = B$ con

$$A = \begin{bmatrix} 1 & 0 & 0 & \ldots & \ldots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \ddots & \ddots & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & \ldots & \ldots & 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \text{ y } x = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_n \end{bmatrix}.$$

Este sistema de ecuaciones lineales tiene solución única, lo que nos asegura que existe un sólo conjunto de valores c_i y, en consecuencia, un solo conjunto de curvas $S_i(x)$. Una vez obtenidos los valores de los c_i, podemos hallar los restantes coeficiente, b_i y d_i con las expresiones ya vistas:

$$\begin{aligned} d_i &= \frac{c_{i+1} - c_i}{3h_i}, \\ b_i &= \frac{a_{i+1} - a_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}), \end{aligned}$$

con lo que obtenemos las $S_i(x)$ curvas o polinomios que interpolan los datos.

Para el caso de las «splines» con frontera sujeta tenemos el siguiente teorema.

Teorema 4.5. Sea f en $a = x_0 < x_1 < \ldots < x_n = b$, y diferenciable en a y en b, entonces f tendrá un interpolante único de frontera sujeta en los nodos $x_0; x_1; \ldots; x_n$.

Demostración Puesto que conocemos $f'(a) = f'(x_0)$, tenemos que

$$b_0 = f'(a) = f'(x_0) = \frac{a_1 - a_0}{h_0} - \frac{h_0}{3}(2c_0 + c_1),$$

y nos queda que

$$2h_0c_0 + h_0c_1 = 3\left[\frac{a_1 - a_0}{h_0} - f'(a)\right].$$

Análogamente, tenemos que

$$f'(b) = f'(x_n) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n),$$

que podemos escribir como

$$\begin{aligned} f'(b) = &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n) \\ &= \frac{a_n - a_{n-1}}{h_{n-1}} + \frac{h_{n-1}}{3}(c_{n-1} + 2c_n), \end{aligned}$$

y que nos deja la siguiente ecuación:

$$h_{n-1}c_{n-1} - 2h_{n-1}c_n = 3\left[f'(b) - \frac{a_n - a_{n-1}}{h_{n-1}}\right].$$

En consecuencia, nos queda el siguiente sistema de ecuaciones

$$A = \begin{bmatrix} 2h_0 & h_0 & 0 & \ldots & \ldots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \ddots & \ddots & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & \ldots & \ldots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix},$$

$$B = \begin{bmatrix} 3\left[\frac{a_1-a_0}{h_0} - f'(a)\right] \\ \frac{3}{h_1}(a_2-a_1) - \frac{3}{h_0}(a_1-a_0) \\ \vdots \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 3\left[f'(b) - \frac{a_n-a_{n-1}}{h_{n-1}}\right] \end{bmatrix}, \text{ y } x = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_n \end{bmatrix}.$$

Como en el caso anterior, el sistema de ecuaciones lineales tiene solución única, es decir, existe un único vector c_0, c_1, ..., c_n, y consecuentemente, un sólo conjunto de curvas $S_i(x)$.

En cuanto al error que cometemos al interpolar una curva utilizando «splines», para el caso con frontera libre podemos expresarlo como

$$\max_{a\le x\le b} |f(x) - S(x)| \le \frac{5}{384} M \max_{0\le i\le n-1} |h_i|^4,$$

donde $S(x)$ es el conjunto de las $S_i(x)$ curvas y $h_i = x_{i+1} - x_i$. Sin embargo, cuando se utiliza este caso, el orden del error en los extremos es proporcional a $|h_i|^2$ y no a $|h_i|^4$, por lo que no siempre es bueno aplicar el caso de frontera libre o natural.

Finalmente, existe un tercer caso cuando no conocemos las derivadas extremas ($f'(a)$ y $f'(b)$), denominado *aproximación sin un nodo* [2], en el cual se considera que $d_0 = d_1$ y $d_{n-2} = d_{n-1}$, que es lo mismo que considerar que $S_0(x) = S_1(x)$ y $S_{n-2}(x) = S_{n-1}(x)$, lo cual también introduce un error en los extremos del orden de $|h_i|^2$.

Para este último caso tenemos lo siguiente:

$$\begin{aligned} c_1 = c_0 + 3d_0h_0 &\Rightarrow d_0 = \frac{c_1 - c_0}{3h_0} \\ c_2 = c_1 + 3d_1h_1 &\Rightarrow d_1 = \frac{c_2 - c_1}{3h_1}. \end{aligned}$$

Como $d_0 = d_1$, entonces

$$\begin{aligned} \frac{c_1 - c_0}{3h_0} &= \frac{c_2 - c_1}{3h_1} \\ h_1c_1 - h_1c_0 &= h_0c_2 - h_0c_1, \end{aligned}$$

lo que nos deja la siguiente expresión para la primera fila del sistema:

$$h_1c_0 - (h_0 + h_1)c_1 + h_0c_2 = 0.$$

Análogamente, para d_{n-2} y d_{n-1} tenemos algo similar:

$$\begin{aligned} c_{n-1} = c_{n-2} + 3d_{n-2}h_{n-2} &\Rightarrow d_{n-2} = \frac{c_{n-1} - c_{n-2}}{3h_{n-2}} \\ c_n = c_{n-1} + 3d_{n-1}h_{n-1} &\Rightarrow d_{n-1} = \frac{c_n - c_{n-1}}{3h_{n-1}}, \end{aligned}$$

con las cuales obtenemos la última fila del sistema:

$$h_{n-1}c_{n-2} - (h_{n-2} + h_{n-1})c_{n-1} + h_{n-2}c_n = 0.$$

[2] Algunos textos denominan a esta aproximación como condición *no un nodo*, por la expresión en inglés *not a knot approximation*.

Así, el sistema queda como:

$$A = \begin{bmatrix} h_1 & -(h_0+h_1) & h_0 & \dots & \dots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \ddots & \ddots & \vdots \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \dots & \dots & h_{n-1} & -(h_{n-2}+h_{n-1}) & h_{n-2} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2-a_1) - \frac{3}{h_0}(a_1-a_0) \\ \vdots \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 0 \end{bmatrix}, \text{ y } x = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_n \end{bmatrix}.$$

Esta variante de la interpolación por «spline» es poco usada porque tiene muchas más indefiniciones que la natural.

4.7. Notas finales

Hemos visto diferentes métodos para interpolar valores a partir de datos discretos usando funciones polinómicas completas, como son los métodos de Lagrange, de Newton y de Hermite, y también la interpolación mediante segmentos de curvas, como es el caso del método de Hermite fragmentado y el de «spline» cúbico. Dentro de este último conjunto está también el método de interpolación lineal por segmentos, cuyas funciones se obtienen utilizando el método de Lagrange tradicional entre dos puntos. Así, los dos polinomios de Lagrange necesarios son:

$$\begin{aligned} L_{1;0}(x) &= \frac{x-x_1}{x_0-x_1} \\ L_{1;1}(x) &= \frac{x-x_0}{x_1-x_0}, \end{aligned}$$

donde x_0 es el punto inicial y x_1 el punto final de la interpolación. Con estos dos polinomios, el polinomio completo de Lagrange resulta ser:

$$\begin{aligned} P_1(x) &= f(x_0)L_{1;0}(x) + f(x_1)L_{1;1}(x) \\ &= f(x_0)\frac{x-x_1}{x_0-x_1} + f(x_1)\frac{x-x_0}{x_1-x_0} = f(x_1)\frac{x-x_0}{x_1-x_0} - f(x_0)\frac{x-x_1}{x_1-x_0} \\ &= \frac{f(x_1)-f(x_0)}{x_1-x_0}x - \frac{f(x_1)x_0 - f(x_0)x_1}{x_1-x_0}. \end{aligned}$$

Además de los métodos vistos, existen otros más complejos que mejoran nuestra aproximación de valores intermedios, pero que suelen ser también más difíciles de implementar y con mayor costo computacional. En particular, para ciertas curvas que no pueden ser definidas mediante polinomios contamos con curvas paramétricas denominadas *curvas de Bezier*. (Para más datos, véase [1].)

Respecto a la interpolación baricéntrica de Lagrange, Berrut y Trefethen (vésae [14]) señalan en su artículo, que resulta curioso que el método no figure en ningún libro de texto de análisis numérico como alternativa al método tradicional, teniendo en cuenta la simplicidad del mismo para ser implementado en una computadora.

Por último, resulta interesante observar que el diseño asistido por computadora (CAD por sus siglas en inglés) hace un uso intensivo de la interpolación fragmentada (o segmentada), con las «spline», y las curvas paramétricas (curvas de Bezier). Las primeras son muy usadas en programas como el AutoCAD®, en tanto que las segundas, en programas como el Corel Draw®, OpenDraw o similares. Por tal motivo, resulta útil conocer los fundamentos matemáticos de cada una de ellas.

Capítulo 5

Mejor aproximación y ajuste de funciones

5.1. Mejor aproximación

5.1.1. Introducción

Uno de los problemas que suele tener que resolver un ingeniero es el de armar una función que ajuste datos obtenidos experimentalmente. Hemos visto en el capítulo anterior como interpolar valores mediante el armado de polinomios que pasan por los puntos dados. Además, de acuerdo con lo visto, si la cantidad de puntos era muy grande, interpolar mediante polinomios creaba curvas de alto grado, que se vuelven muy inestables en los extremos. Vimos que de todos modos esto se podía resolver en parte mediante interpolaciones fragmentadas o segmentos de curvas (polinomios de Hermite cuando conocemos la primera derivada o «spline» cuando no la conocemos). Pero en todos los casos, una de las condiciones fundamentales es que los puntos x_i sean distintos. ¿Qué hacemos cuando esto no es así, cuando la cantidad de puntos exceden la capacidad de armar polinomios interpolantes o cuando los puntos que usaremos son aproximaciones a los valores reales?

Supongamos que tenemos una serie de datos empíricos obtenidos en laboratorio, tales que el conjunto de datos no cumple estrictamente que los x_i sean distintos, con lo cual para un mismo x_i tenemos varios valores de $f(x_i)$. (En realidad suele suceder que aunque los x_i sean distintos, varios x_j sean suficientemente cercanos como para considerarlos iguales.) Además que la cantidad de datos disponibles hagan imposible que armemos un polinomio de grado menor a 4 o 5 que pase por todos los puntos y así evitar el mal condicionamiento del problema. Lo que necesitamos, entonces, es una curva que *ajuste* lo mejor posible los datos que disponemos, o sea, que el error entre los puntos y esa función de ajuste sea el menor posible, *sin que la curva pase por los puntos dados.*

Para ello tenemos una forma de estimar este error. Supongamos que efectivamente se cumpla que los x_i sean distintos, que $x_0 < x_1 < \ldots < x_n$ para los cuales conocemos $f(x_0)$,$f(x_1)$,... ,$f(x_n)$. Asumamos que la aproximación la haremos con una función que definiremos de la siguiente manera:

$$g(x) = c_0\phi_0(x) + c_1\phi_1(x) + \ldots + c_m\phi_m(x) = \sum_{i=0}^{m} c_i\phi_i(x),$$

donde $m < n$, es decir, tenemos menos funciones disponibles que puntos, y las $\phi_i(x)$ son linealmente independientes.

Dado que hemos impuesto que la función elegida no debe pasar por lo puntos tomados como dato, buscaremos que el error entre los datos $(f(x_i))$y los $g(x_i)$ de la función de ajuste sea el menor posible, plantearemos que

$$r_i = f(x_i) - g(x_i), \text{ para } 0 \leq i \leq n,$$

es decir, que el residuo, sea mínimo. Como se trata de un vector, una forma de analizar este caso es mediante.

5.1.2. Error y normas

Para obtener una función que minice este residuo, analizaremos que opciones disponemos respecto a la norma, a saber:

1. Que la norma uno del residuo sea mínima, es decir, $\|r\|_1$ sea mínima;
2. Que la norma infinita del residuo sea mínima, es decir, $\|r\|_\infty$ sea mínima;
3. Que la norma dos (euclídea) del residuo sea mínima, es decir, $\|r\|_2$ sea mínima.

La primera norma es buena si uno quiere eliminar aquellos valores considerados como desviaciones, por ejemplo, mediciones mal hechas o valores que fácilmente puede inferirse erróneos. Consiste en minimizar la siguiente expresión:

$$\|r\|_1 = \sum_{i=0}^{n} |r_i| = \sum_{i=0}^{n} |f(x_i) - y(x_i)|$$

La segunda, es un caso de mínimo-máximo en la cual se tiene que:

$$\min_{c_0;c_1;\ldots;c_m} \max_{0\leq j\leq n} |f(x_i) - y(x_i)| .$$

Esto es útil cuando los valores máximos del error deben ser considerados al momento de la verificación.

Ambos casos resultan muy útiles cuando se trabaja con datos discretos, en los que tiene suma importancia verificar la exactitud de esos datos, o eventualmente, encontrar errores de medición, de transcripción, etc.

Los dos casos recién analizados, $\|r\|_1$, y $\|r\|_\infty$ llevan a la *programación lineal*, materia que está fuera del alcance de nuestro curso, y que resultan mucho más complejos de analizar que la última opción indicada.

Ésta consiste en minimizar la expresión:

$$\|r\|_2 = \sqrt{\sum_{i=0}^{n} |r_i|^2} = \sqrt{\sum_{i=0}^{n} [f(x_i) - y(x_i)]^2},$$

o, lo que es lo mismo,

$$\|r\|_2^2 = \sum_{i=0}^{n} |r_i|^2 = \sum_{i=0}^{n} [f(x_i) - y(x_i)]^2 .$$

Como nuestra función la podemos expresar como:

$$y(x) = \sum_{j=0}^{m} c_j \phi_j x,$$

tendremos que la expresión a minimizar es:

$$E(c_0; c_1; \ldots; c_m) = \sum_{i=0}^{n} \left[f(x_i) - \sum_{j=0}^{m} c_j \phi_j(x_i) \right]^2 ,$$

de ahí el nombre de *método de los cuadrados mínimos*, pues lo que se minimiza es el cuadrado del residuo.

5.1.3. Método de los cuadrados mínimos

Para obtener que la función $E(c_0; c_1; \ldots; c_m)$ sea mínima, debemos aplicar un concepto conocido: hacer que $\dfrac{\partial\ E}{\partial c_j} = 0$, puesto que E es función de los coeficientes. En consecuencia, tendremos que:

$$\begin{aligned}\frac{\partial\ E}{\partial c_j} &= \frac{\partial}{\partial c_j}\left[\sum_{i=0}^{n}\left(f(x_i) - \sum_{k=0}^{m} c_k\ \phi_k(x_i)\right)^2\right] = 0 \\ &= \sum_{i=0}^{n}\frac{\partial}{\partial c_j}\left(f(x_i) - \sum_{k=0}^{m} c_k\ \phi_k(x_i)\right)^2 = 0.\end{aligned}$$

que si desarrollamos nos queda:

$$\frac{\partial\ E}{\partial c_j} = 2\sum_{i=0}^{n}\left(f(x_i) - \sum_{k=0}^{m} c_k\ \phi_k(x_i)\right)\ (-\phi_j(x_i)) = 0 \text{ para } j = 0; 1; \ldots; m.$$

Al distribuir el producto nos queda:

$$\begin{aligned}\sum_{i=0}^{n}\left(f(x_i)\phi_j(x_i) - \sum_{k=0}^{m} c_k\ \phi_k(x_i)\phi_j(x_i)\right) &= 0 \\ \sum_{i=0}^{n} f(x_i)\phi_j(x_i) - \sum_{i=0}^{n}\sum_{k=0}^{m} c_k\ \phi_k(x_i)\phi_j(x_i) &= 0 \\ \sum_{i=0}^{n}\sum_{k=0}^{m} c_k\ \phi_k(x_i)\phi_j(x_i) &= \sum_{i=0}^{n} f(x_i)\phi_j(x_i),\end{aligned}$$

para $j = 0; 1; \ldots; m$. Como podemos intercambiar las sumatorias, finalmente nos queda:

$$\sum_{k=0}^{m} c_k\ \sum_{i=0}^{n}\phi_k(x_i)\phi_j(x_i) = \sum_{i=0}^{n} f(x_i)\phi_j(x_i),$$

para $j = 0; 1; \ldots; m$.

Avancemos un poco más. Al desarrollar la sumatoria en i del término de la izquierda, nos queda:

$$\sum_{i=0}^{n}\phi_k(x_i)\phi_j(x_i) = \phi_k(x_0)\phi_j(x_0) + \phi_k(x_1)\phi_j(x_1) + \ldots + \phi_k(x_n)\phi_j(x_n).$$

Lo mismo podemos hacer con la sumatoria del término de la derecha, con lo que tenemos

$$\sum_{i=0}^{n} f(x_i)\phi_j(x_i) = f(x_0)\phi_j(x_0) + f(x_1)\phi_j(x_1) + \ldots + f(x_n)\phi_j(x_n).$$

Para facilitar la notación, definiremos lo siguiente:

$$\begin{aligned}\sum_{i=0}^{n}\phi_k(x_i)\phi_j(x_i) &= (\phi_k; \phi_j) \\ \sum_{i=0}^{n} f(x_i)\phi_j(x_i) &= (f; \phi_j).\end{aligned}$$

Entonces, la expresión que nos queda es

$$\sum_{k=0}^{m} c_k\ (\phi_k; \phi_j) = (f; \phi_j),$$

para $j = 0; 1; \ldots; m$. Ahora desarrollaremos la sumatoria en k, con lo cual obtenemos lo siguiente:

$$c_0\ (\phi_0; \phi_j) + c_1\ (\phi_1; \phi_j) + \ldots + c_m\ (\phi_m; \phi_j) = (f; \phi_j).$$

Como $j = 0; 1; \ldots; m$, entonces podemos armar $m + 1$ ecuaciones, lo que finalmente nos deja:

$$\begin{array}{rcl} c_0\ (\phi_0; \phi_0) + c_1\ (\phi_1; \phi_0) + \ldots + c_m\ (\phi_m; \phi_0) & = & (f; \phi_0) \\ c_0\ (\phi_0; \phi_1) + c_1\ (\phi_1; \phi_1) + \ldots + c_m\ (\phi_m; \phi_1) & = & (f; \phi_1) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots & & \vdots \\ c_0\ (\phi_0; \phi_m) + c_1\ (\phi_1; \phi_m) + \ldots + c_m\ (\phi_m; \phi_m) & = & (f; \phi_m), \end{array}$$

que podemos escribir también en forma matricial como

$$\begin{bmatrix} (\phi_0; \phi_0) & (\phi_1; \phi_0) & \ldots & (\phi_m; \phi_0) \\ (\phi_0; \phi_1) & (\phi_1; \phi_1) & \ldots & (\phi_m; \phi_1) \\ \vdots & \vdots & \ddots & \vdots \\ (\phi_0; \phi_m) & (\phi_1; \phi_m) & \ldots & (\phi_m; \phi_m) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} (f; \phi_0) \\ (f; \phi_1) \\ \vdots \\ (f; \phi_m) \end{bmatrix}.$$

Esta matriz resulta ser simétrica, pues $(\phi_i; \phi_j) = (\phi_j; \phi_i)$, y definida positiva. El problema se reduce a resolver un sistema ecuaciones lineales cuyas incógnitas son los coeficientes c_k. Obtenidos estos coeficientes, los reemplazamos en la función que hemos definido, que será la que aproxime nuestros puntos.

Existe otra forma de plantear el problema, esta vez en forma matricial desde el principio. Supongamos que representamos nuestros puntos con la función elegida. Entonces nos queda:

$$\begin{array}{rcl} f(x_0) & = & \displaystyle\sum_{k=0}^{m} c_k \phi_k(x_0) = c_0\phi_0(x_0) + c_1\phi_1(x_0) + \ldots + c_m\phi_m(x_0) \\ f(x_1) & = & \displaystyle\sum_{k=0}^{m} c_k \phi_k(x_1) = c_0\phi_0(x_1) + c_1\phi_1(x_1) + \ldots + c_m\phi_m(x_1) \\ & \vdots & \\ f(x_n) & = & \displaystyle\sum_{k=0}^{m} c_k \phi_k(x_n) = c_0\phi_0(x_n) + c_1\phi_1(x_n) + \ldots + c_m\phi_m(x_n) \end{array}$$

Si escribimos esto en foma matricial nos queda

$$\begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \ldots & \phi_m(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \ldots & \phi_m(x_1) \\ \vdots & & & \\ \phi_0(x_n) & \phi_1(x_n) & \ldots & \phi_m(x_n) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix},$$

que resulta ser un sistema de m incógnitas con n ecuaciones, donde $m < n$, en el cual no existe una única solución. Si hacemos

$$f = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}; \Phi = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \ldots & \phi_m(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \ldots & \phi_m(x_1) \\ \vdots & & & \\ \phi_0(x_n) & \phi_1(x_n) & \ldots & \phi_m(x_n) \end{bmatrix} \text{ y } c = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix},$$

podemos decir que nos queda una ecuación del tipo $f = \Phi c$. Como lo que buscamos es aproximar una función, definamos el residuo como $r = f - \Phi c$. Al igual que en el desarrollo anterior, vamos a obtener nuestra nueva función haciendo que $\|r\|_2^2$ sea mínimo. En conscuencia, tenemos

$$\|r\|_2^2 = \|f - \Phi c\|_2^2.$$

Recordemos que $\|r\|_2^2 = r^T\, r$, entonces tendremos que

$$r^T\, r = (f - \Phi c)^T\ (f - \Phi c)\,.$$

De nuevo, para obtener que el residuo sea mínimo, anulemos la primera derivada, es decir, hagamos

$$\frac{\partial\ r^T\, r}{\partial c_j} = \frac{\partial}{\partial c_j}\left[(f - \Phi c)^T\ (f - \Phi c)\right] = 0.$$

Al derivar nos queda

$$-\Phi^T(f - \Phi c) - (f - \Phi c)^T\,\Phi = 0,$$

que desarrollada se transforma en

$$\Phi^T f - \Phi^T\Phi\, c + f^T\Phi - c^T\Phi^T\Phi = 0.$$

Como $\Phi^T f = f^T\Phi$ y $c^T\Phi^T\Phi = \Phi^T\Phi\, c$, la ecuación anterior nos queda como

$$\begin{aligned} \Phi^T f - \Phi^T\Phi\, c + \Phi^T f - \Phi^T\Phi\, c &= 0 \\ 2(\Phi^T f - \Phi^T\Phi\, c) &= 0 \\ \Phi^T f - \Phi^T\Phi\, c &= 0 \Rightarrow \\ \Phi^T\Phi\, c &= \Phi^T f, \end{aligned}$$

donde $\Phi^T\Phi$ es una matriz simétrica definida positiva, y tiene la forma

$$\begin{bmatrix} (\phi_0;\phi_0) & (\phi_1;\phi_0) & \dots & (\phi_m;\phi_0) \\ (\phi_0;\phi_1) & (\phi_1;\phi_1) & \dots & (\phi_m;\phi_1) \\ \vdots & \vdots & \ddots & \vdots \\ (\phi_0;\phi_m) & (\phi_1;\phi_m) & \dots & (\phi_m;\phi_m) \end{bmatrix};$$

y $\Phi^T f$ tiene la forma

$$\begin{bmatrix} (f;\phi_0) \\ (f;\phi_1) \\ \vdots \\ (f;\phi_m) \end{bmatrix}.$$

Si hacemos $A = \Phi^T\Phi$, $x = c$ y $B = \Phi^T f$, volvemos a tener nuestro sistema de ecuaciones lineales en la forma $Ax = B$. De nuevo, el método de los cuadrados mínimos no es otra cosa que la resolución de un sistema de ecuaciones lineales para obtener los coeficientes c de nuestra función de ajuste, algo a lo que habíamos llegado mediante la deducción anterior.

Este método suele usarse para obtener la recta de regresión. Para obtenerlo, basta que observemos que

$$y(x) = \sum_{i=0}^{m} c_i\phi_i(x) = c_0 + c_1 x,$$

es la recta que ajusta nuestros datos, con lo cual $\phi_0 = 1$ y $\phi_1 = x$. El siguiente paso es armar la matriz A. Sabemos que

$$(\phi_k;\phi_j) = \sum_{i=0}^{n} \phi_k(x_i)\phi_j(x_i) \text{ y } (f;\phi_j) = \sum_{i=0}^{n} f(x_i)\phi_j(x_i),$$

entonces podemos escribir las componentes de A y B como

$$\begin{aligned}
(\phi_0;\phi_0) &= \sum_{i=0}^{n} 1\cdot 1 = n+1\\
(\phi_1;\phi_0) &= \sum_{i=0}^{n} x_i\cdot 1 = \sum_{i=0}^{n} x_i\\
(\phi_0;\phi_1) &= (\phi_1;\phi_0) = \sum_{i=0}^{n} x_i\\
(\phi_1;\phi_1) &= \sum_{i=0}^{n} (x_i\cdot x_i) = \sum_{i=0}^{n} (x_i)^2\\
(f;\phi_0) &= \sum_{i=0}^{n} (f(x_i)\cdot 1) = \sum_{i=0}^{n} f(x_i)\\
(f;\phi_1) &= \sum_{i=0}^{n} (f(x_i)\cdot x_i),
\end{aligned}$$

y nuestro sistema quedará de la siguiente manera:

$$\begin{bmatrix} n+1 & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} (x_i)^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} f(x_i) \\ \sum_{i=0}^{n} (f(x_i)\cdot x_i) \end{bmatrix}.$$

Despejando c_0 y c_1 obtenemos:

$$\begin{aligned}
c_0 &= \frac{\sum_{i=0}^{n}(x_i)^2 \sum_{i=0}^{n} f(x_i) - \sum_{i=0}^{n}(f(x_i)\cdot x_i)\sum_{i=0}^{n} x_i}{(n+1)\sum_{i=0}^{n}(x_i)^2 - \left(\sum_{i=0}^{n} x_i\right)^2}\\
c_1 &= \frac{(n+1)\sum_{i=0}^{n}(f(x_i)\cdot x_i) - \sum_{i=0}^{n} x_i \sum_{i=0}^{n} f(x_i)}{(n+1)\sum_{i=0}^{n}(x_i)^2 - \sum_{i=0}^{n} x_i}
\end{aligned}$$

Existen algunas variantes para este tipo de regresiones, que son:

$$\begin{aligned}
\ln(y) &= \ln(c_0) + c_1 \ln(x)\ (y = c_0 x^{c_1})\\
\ln(y) &= \ln(c_0) + c_1 x\ (y = c_0 e^{c_1 x})\\
y &= c_0 + c_1 \ln(x),
\end{aligned}$$

que permiten ajustar valores según distintas curvas. Sin embargo, las primeras expresiones no son ajustes por cuadrados mínimos en un sentido estricto. Lo correcto sería proponer una función del tipo $\sum_i c_i\phi_i(x)$ en lugar de transformar los datos. (Para más detalles, véase [1].)

Si ampliamos este esquema a una función polinómica de grado mayor o igual a 2, tendremos que

$$y(x) = \sum_{k=0}^{m} c_k\phi_k(x) = \sum_{k=0}^{m} c_k\, x^k = c_0 + c_1 x + c_2 x^2 + \ldots + c_m x^m.$$

Al armar el sistema de ecuaciones nos quedará

$$\begin{bmatrix} n+1 & \sum_{i=0}^{n} x_i & \dots & \sum_{i=0}^{n} x_i^{m-1} & \sum_{i=0}^{n} x_i^{m} \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^{2} & \dots & \sum_{i=0}^{n} x_i^{m} & \sum_{i=0}^{n} x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum_{i=0}^{n} x_i^{m-1} & \sum_{i=0}^{n} x_i^{m} & \dots & \sum_{i=0}^{n} x_i^{2(m-1)} & \sum_{i=0}^{n} x_i^{2m-1} \\ \sum_{i=0}^{n} x_i^{m} & \sum_{i=0}^{n} x_i^{m+1} & \dots & \sum_{i=0}^{n} x_i^{2m-1} & \sum_{i=0}^{n} x_i^{2m} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \\ c_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} f(x_i) \\ \sum_{i=0}^{n} (f(x_i) \cdot x_i) \\ \vdots \\ \sum_{i=0}^{n} (f(x_i) \cdot x_i^{m-1}) \\ \sum_{i=0}^{n} (f(x_i) \cdot x_i^{m}) \end{bmatrix}.$$

La matriz de coeficientes es similar a una *matriz de Vandemonde*, matriz que obtuvimos para interpolar una serie de puntos, de ahí que cualquier ajuste de curvas hecho con polinomios resulta ser un problema *mal condicionado*. Por supuesto, la mala condición de la matriz se hace cada vez más evidente a medida que m sea más grande. Es por eso que no se recomienda trabajar con polinomios de grado mayor a 4 o 5, para evitar que la mala condición de la matriz sea un problema. Aún así, trabajar con un polinomio de grado 5 conlleva trabajar con coeficientes que incluyen x^{10}, lo que resulta casi equivalente a interpoalr con polinomios de grado 10. Por esto, conviene que recordemos que el ajuste polinomial, al igual que la interpolación polinomial son problemas con tendencia a ser mal condicionados.

5.2. Ajuste de funciones

5.2.1. Introducción

En el punto anterior hemos visto un método para ajustar curvas a partir de datos numéricos (discretos), con el objetivo de obtener valores de la función $f(x)$ para valores de x distintos a los datos en el intervalo dado. E

Ahora bien, existen situaciones en las cuales aún conociendo la función $f(x)$, resulta conveniente efectuar algún tipo de aproximación. Un ejemplo típico de ello es el caso de las funciones trigonométricas (por ejemplo, $cos(x)$), para la cual es necesario realizar alguna aproximación para calcular sus valores. La más común es la hecha mediante las series de Taylor. Para estas funciones puede ser muy útil aplicar el desarrollo en series, pero no suele ser el caso general, puesto que las series de Taylor son válidas sólo en el entorno de un punto, lo que le quita generalidad.

¿Y en qué casos necesitaríamos nosotros contar con una aproximación de una función conocida? Supongamos que tenemos la siguiente función:

$$f(x) = \frac{e^x - cos(x)}{ln(x) \cdot arctan(x)},$$

en un intervalo $[a, b]$. Supongamos además, que nuestro problema exige que integremos esa función $f(x)$ en el intervalo dado. Podemos ver que la situación ya no es tan fácil como parece. Si bien disponemos de la función, hallar la primitiva puede ser todo un desafío, e incluso, imposible. Pero de alguna manera debemos salvar el escollo.

¿Que tal si en vez de hacer una integral «analítica» nos orientamos hacia una solución numérica? La idea no es tan descabellada pues lo que nosotros necesitamos es el resultado numérico y no la primitiva de la misma. Hagamos uso entonces de nuestras herramientas numéricas aprendidas anteriormente y, si es necesario, adecuemos nuestras expresiones al caso analizado.

5.2.2. Aproximación por mínimos cuadrados

Recordemos qué significa reducir al mínimo el error cuadrático entre la función y el polinomio de aproximación. Supongamos por un momento que conocemos tanto la función $f(x)$ como el polinomio de aproximación $P(x)$, en el intervalo $[a; b]$. Podemos graficar nuestra función y nuestro polinomio de manera que nos queden las curvas que se ven la figura 5.1.

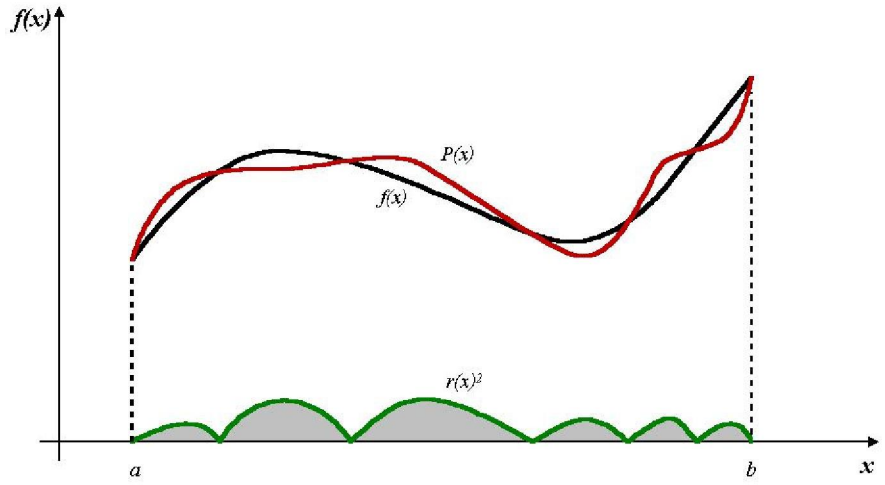


Figura 5.1: Error cuadrático

Si definimos que:

$$E(a_k) = \int_a^b \left[f(x) - \sum_{k=0}^{n} a_k x^k \right]^2 dx = \|r(a_k)\|_2^2 ,$$

entonces podemos ver que que el área bajo la curva $r(a_k)^2$ es el valor de nuestra integral. Por lo tanto, para que nuestro error cuadrático sea mínimo, deberemos buscar que la curva $r(a_k)^2$ sea los más parecida al eje de abscisas. (Esta definición es similar a la vista para ajuste de curvas.)

Para ello, vamos a derivar la función $E(a_k)$ respecto de los coeficientes a_k para obtener los valores de dichos coeficientesn que hagan mínimo el error cuadrático, tal como hicimos para el caso de un ajustse discreto. Entonces tendremos:

$$\frac{\partial E\left(a_0; a_1; \ldots, a_n\right)}{\partial a_j} = 0 \Rightarrow \frac{\partial}{\partial a_j} \left\{ \int_a^b \left[f(x) - \sum_{k=0}^{n} a_k x^k \right]^2 dx \right\} = 0.$$

Al derivar nos queda:

$$2 \cdot \int_a^b \left[f(x) - \sum_{k=0}^{n} a_k x^k \right] x^j dx = 0,$$

y como el 2 no incide, nos queda:

$$\int_a^b \left[f(x) - \sum_{k=0}^{n} a_k x^k \right] x^j dx = 0.$$

Si distribuimos el producto dentro de la integral, nos queda:

$$\sum_{k=0}^{n} a_k \int_a^b x^{k+j} dx = \int_a^b x^j f(x) dx \quad \text{para } j = 0; 1; \ldots; n.$$

¿Qué es lo hemos obtenido? Nuevamente, como en la aproximación de puntos discretos, un sistema de ecuaciones lineales de dimensión $n+1 \times n+1$. Sin embargo, no todo es tan sencillo. Analicemos un poco más en detalle la integral que afecta a los coeficientes a_k. Tenemos que:

$$\int_a^b x^{k+j} dx = \frac{x^{k+j+1}}{k+j+1}\Big|_a^b \Rightarrow \int_a^b x^{k+j} dx = \frac{b^{k+j+1} - a^{k+j+1}}{k+j+1}$$

Si definimos que $a = 0$ y $b = 1$, entonces la integral definida resulta en el coeficiente:

$$\frac{1}{k+j+1}$$

La matriz que se genera a partir del coeficiente anterior es conocida como *matriz de Hilbert*, que es una matriz *mal condicionada*. Como en el caso anterior de ajuste discreto, al tener una matriz mal condicionada, el sistema es muy sensible a los cambios en los datos, o modificaciones de la matriz de coeficientes, es decir, es muy sensible a los errores inherentes.

Un segundo problema, en este caso operativo, es que si por algún motivo se desea agregar un término más al polinomio, hay que recalcular el sistema (agregar una columna y una fila), lo que significa mucho trabajo adicional. Y nada asegura que los nuevos resultados estén exentos de errores. De todos modos, contamos con método muy potente para ajustar funciones pero con inconvientes operativos en el planteo numérico. Podemos buscar la forma de mejorarlo. Veamos como.

¿Cuál sería la mejor matriz de coeficientes para resolver un sistema de ecuaciones lineales? Evidentemente, aquella que independice cada incógnita de las otras. O sea, que la matriz de coeficientes sea una matriz diagonal. Supongamos modificar levemente la expresión del polinomio de aproximación por la siguiente:

$$P(x) = \sum_{k=0}^{n} a_k \phi_k(x),$$

En principio, no hemos hecho sino un cambio de notación, llamando a x^k como $\phi_k(x)$. Veamos qué ventajas nos trae esto. Por lo pronto, ahora disponemos de más posibilidades porque el método no cambia si proponemos una suma de funciones en vez de un polinomio como función de aproximación, tal como vimos para ajuste de curvas. Entonces nos queda:

$$\sum_{k=0}^{n} a_k \int_a^b \phi_k(x)\phi_j(x)dx = \int_a^b \phi_j f(x)dx \quad \text{para } j = 0; 1; \ldots; n.$$

que conceptualmente es muy parecido a lo anterior. Pero con una diferencia: ahora podemos tomar cualquier función para definir nuestras funciones $\phi_k(x)$ y por lo tanto, también nuestros $\phi_j(x)$. Busquemos entonces que nuestra matriz de coeficientes se convierta en una matriz diagonal. ¿Y cómo lo logramos? Sencillamente estableciendo que se cumpla lo siguiente:

$$\int_a^b \phi_k(x)\phi_j(x)dx = \begin{cases} 0 & \text{si} \quad k \neq j \\ M > 0 & \text{si} \quad k = j \end{cases},$$

donde M es un valor cualquiera. Por supuesto, lo ideal sería que $M = 1$. Esta condición que deben cumplir las $\phi_k(x)$ asegura que las funciones sean ortogonales. En consecuencia, nuestra matriz de coeficientes será diagonal.

No hemos dicho nada aún acerca de las funciones $\phi_k(x)$. Como estamos tratando de aproximar una función cualquiera, una buena idea es proponer que esas funciones sean también polinomios. Para hallar estos polinomios ortogonales entre sí, debemos agregar una segunda condición que es agregar una función *de peso*. Esta función de peso tiene por objeto asignar

diferentes grados de importancia a las aproximaciones de ciertas partes del intervalo. En esta situación tenemos:

$$\frac{\partial E\left(a_0; a_1; \ldots, a_n\right)}{\partial a_j} = 0 \Rightarrow \frac{\partial}{\partial a_j}\left\{\int_a^b w(x)\left[f(x) - \sum_{k=0}^{n} a_k\phi_k(x)\right]^2 dx\right\} = 0,$$

con lo cual finalmente nos queda:

$$\int_a^b w(x)\left[f(x) - \sum_{k=0}^{n} a_k\phi_k\right]\phi_j\, dx = 0.$$

En este caso se debe cumplir que:

$$\int_a^b w(x)\phi_k(x)\phi_j(x)dx = \begin{cases} 0 & \text{si} \quad k \neq j \\ M \neq 0 & \text{si} \quad k = j \end{cases}.$$

Si definimos que $w(x) = 1$, volvemos a tener nuestra expresión original para los $\phi_k(x)$ y los $\phi_j(x)$. Y si, además, el intervalo de interpolación lo fijamos en $[-1, 1]$, el resultado es que mediante este procedimiento obtenemos los ***polinomios de Legendre***. Estos polinomios los usaremos más adelante para integrar numéricamente.

5.2.3. Polinomios de Legendre

Veremos cómo se calculan los polinomios de Legendre. Antes, debemos recordar cómo se obtenía un conjunto de vectores ortogonales a partir de un conjunto no ortogonal. Esto se conseguía mediante el *proceso de Gram-Schmidt*. Adaptémoslo para el caso de funciones. Primero, debemos proponer las dos primeras funciones $\phi(x)$. Estas funciones son:

$$\phi_0(x) = 1;\ \phi_1(x) = x - B_1 \ \Rightarrow\ \phi_1(x) = (x - B_1)\,\phi_0(x).$$

donde B_1 es nuestra incógnita. Para obtenerla debemos plantear que:

$$\int_a^b w(x)\phi_0(x)\phi_1(x)\, dx = 0 \ \Rightarrow\ \int_a^b w(x)\phi_0(x)\phi_0(x)\,(x - B_1)\ dx = 0.$$

Distribuyendo en el paréntesis, obtenemos:

$$\int_a^b w(x)[\phi_0(x)]^2 x\, dx - B_1\int_a^b w(x)[\phi_0(x)]^2\, dx = 0$$

y entonces B_1 se puede hallar con:

$$B_1 = \frac{\displaystyle\int_a^b w(x)[\phi_0(x)]^2\, x\, dx}{\displaystyle\int_a^b w(x)[\phi_0(x)]^2 dx}.$$

Para los siguientes polinomios, es decir, cuando $k \geq 2$, debemos proponer que:

$$\phi_k(x) = (x - B_k)\ \phi_{k-1}(x) - C_k\phi_{k-2}(x)\ \text{ en }\ [a; b].$$

Operando algebraicamente en forma similar a la anterior obtenemos los coeficientes B_k y C_k:

$$B_k = \frac{\displaystyle\int_a^b x\, w(x)[\phi_{k-1}(x)]^2\, dx}{\displaystyle\int_a^b w(x)[\phi_{k-1}(x)]^2\, dx}$$

$$C_k = \frac{\displaystyle\int_a^b x\, w(x)\phi_{k-1}(x)\phi_{k-2}(x)\, dx}{\displaystyle\int_a^b w(x)[\phi_{k-2}(x)]^2\, dx}$$

Como hemos dicho, la función de peso en el caso de los polinomios de Legendre es $w(x) = 1$ y el intervalo $[-1; 1]$, por lo que las expresiones quedan como sigue:

1. El coeficiente B_1 se obtiene con:
$$B_1 = \frac{\int_{-1}^{1} x\, dx}{\int_{-1}^{1} dx};$$

2. Los coeficientes B_k se obtienen con la expresión
$$B_k = \frac{\int_{-1}^{1} x\, [\phi_{k-1}(x)]^2\, dx}{\int_{-1}^{1} [\phi_{k-1}(x)]^2\, dx};$$

3. Y, finalmente, los coeficientes C_k se obtienen con:
$$C_k = \frac{\int_{-1}^{1} x\, \phi_{k-1}(x)\phi_{k-2}(x)\, dx}{\int_{-1}^{1} [\phi_{k-2}(x)]^2\, dx}.$$

Existe un segundo conjunto de polinomios ortogonales muy utilizados que son los *polinomios de Chebishev*. También se generan aplicando las expresiones generales ya vistas, pero con una función de peso ($w(x)$) diferente: $w(x) = \frac{1}{\sqrt{1-x^2}}$.

5.3. Notas finales

Tanto la aproximación discreta de curvas como el ajueste de funciones tienen un amplio uso en la ingeniería. En el primer caso, existen muchas expresiones matemáticas resultantes de aproximar valores obtenidos experimentalmente en laboratorios o mediante mediciones realizadas sobre prototipos. En la ingeniería hidráulica se tienen muchas expresiones empíricas que surgen de experiencias en laboratorio que luego resultan en fórmulas matemáticas obtenidas mediante aproximaciones discretas.

Con el ajuste de funciones ocurre algo similar. El ejemplo más interesante es el uso de polinomios de Legendre en la cuadratura de Gauss para integrar numéricamente. Estos polinomios ajustan cualquier tipo de funciones y en particular, a cualquier polinomio, lo que facilita obtener soluciones numéricas «exactas» de cualquier integral numérica que incluya funciones polinómicas, como se verá en el capítulo siguiente.

Capítulo 6

Diferenciación e integración numérica

6.1. Diferenciación numérica

Como dijimos en la introducción del capítulo 2, trabajar en forma simbólica resulta bastante complicado cuando se requiere el uso de computadoras, aún cuando existen programas que lo hagan (en parte). No siempre la soluciones analíticas son aplicables al problema que se está tratando de resolver, y peor aún, en muchos casos no hay tal solución análitica, como veremos más adelante.

Por otro lado, muchas veces tampoco disponemos de las herramientas para trabajar en forma simbólica (analítica). Cuando sólo contamos con datos obtenidos de mediciones o de cálculos previos, y no de funciones, no suele ser práctico trabajar en forma simbólica. Obtener «la derivada de una función» con datos discretos no tiene mucho sentido.

Al mismo tiempo, muchos programas de aplicación ingenieril no pueden almacenar o guardar en sus líneas de código una base de datos que incluya las derivadas de cualquier función (lo mismo se aplica al caso inverso, la integración). La cantidad de información y la aleatoriedad que puede presentar una exigencia de este tipo vuelve impractible realizar esto en cada programa, además de llevar a construir interfaces amigables, que contribuyen a aumentar los requerimientos de memoria, tanto de operación como de almacenamiento.

Veremos a continuación como encarar la diferenciación mediante métodos numéricos con ayuda de varios ejemplos, analizando las ventajas y las desventajas de cada método empleado en la discretización para luego analizar la *extrapolación de Richardson*, método que puede usarse también para otros casos.

6.1.1. Diferencias progresivas, regresivas y centradas

La diferenciación es un tema muy conocido por los estudiantes de ingeniería. Los primeros años de la carrera consisten en estudiar en detalle cómo caracterizar y conocer a fondo una función dada, de manera que para analizar si tiene máximos o mínimos, si es convexa o cóncava, si puede aproximarse mediante un desarrollo en serie, lo primero que se aprende es el concepto de ***derivada***, tanto total como parcial. Tomemos, por ejemplo, la función

$$f(x) = seno\left(\frac{2\pi}{b}x\right).$$

Hallar la derivada primera de $f(x)$ respecto de x es un procedimiento sencillo pues resulta ser

$$f'(x) = \frac{\mathrm{d}\, f(x)}{\mathrm{d}x} = \left(\frac{2\pi}{b}\right) cos\left(\frac{2\pi}{b}x\right).$$

Si queremos conocer la derivada en el punto $x = \frac{b}{6}$ entonces basta con reemplazar ese valor en la expresión anterior y tendremos que

$$f'\left(\frac{b}{6}\right) = \left(\frac{2\pi}{b}\right) cos\left(\frac{2\pi}{b}\frac{b}{6}\right) = \left(\frac{2\pi}{b}\right) cos\left(\frac{\pi}{3}\right)$$

Si finalmente le damos un valor a b, (por ejemplo, $b = 6$), el valor de nuestra derivada en $x = \frac{b}{6} = 1$ será

$$f'(1) = \left(\frac{\pi}{3}\right) cos\left(\frac{\pi}{3}\right) = \frac{\pi}{6} \approx 0,5236$$

Supongamos ahora que queremos obtener ese mismo valor pero no recordamos cómo hallar la derivada en forma analítica. Aplicando el concepto del cual se deduce, podemos decir que

$$f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

que también suele escribirse como:

$$f'(x) \approx \frac{f(x + h) - f(x)}{h}$$

Para hallar la derivada en nuestro punto $x = \frac{b}{6}$ con $b = 6$ adoptemos el valor $h = 0,1$. Así tendremos que

$$f'(1) \approx \frac{f(1,1) - f(1)}{0,1} = \frac{seno\left(\frac{\pi}{3}1,1\right) - seno\left(\frac{\pi}{3}\right)}{0,1}$$

$$f'(1) \approx \frac{0,9135 - 0,8660}{0,1} = 0,4750$$

Podemos ver que nuestra aproximación es razonable pero no muy buena, y que el error cometido es del orden del 10 %. Como no estamos conformes con el resultado obtenido, proponemos otro algoritmo para hallar el valor buscado. Este algoritmo es

$$f'(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x}$$

o, como también suele escribirse

$$f'(x) \approx \frac{f(x) - f(x - h)}{h}$$

Hallemos ahora el valor de la derivada utilizando este nuevo algoritmo. El resultados es

$$f'(1) \approx \frac{f(1) - f(0,9)}{0,1} = \frac{seno\left(\frac{\pi}{3}\right) - seno\left(\frac{\pi}{3}0,9\right)}{0,1}$$

$$f'(1) \approx \frac{0,8660 - 0,8090}{0,1} = 0,5700$$

De nuevo, el valor obtenido tampoco es una aproximación muy buena, pues el error cometido del orden del 8 %. Una vez más, no estamos conformes con el resultado que nos arrojó este algoritmo y proponemos este otro

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

o también:

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h}$$

Reemplazando los valores, tendremos:

$$f'(1) \approx \frac{f(1,1) - f(0,9)}{0,2} = \frac{seno\left(\frac{\pi}{3}1,1\right) - seno\left(\frac{\pi}{3}0,9\right)}{0,2}$$

$$f'(1) \approx \frac{0,9135 - 0,8090}{0,2} = 0,5225$$

Evidentemente, el valor de la derivada en el punto pedido es bastante aproximado al considerado «real» o «exacto». Podemos notar que el error cometido es del orden del 0,2 %. Cada una de estas aproximaciones son equivalentes a efectuar una interpolación aplicando el método de Lagrange y luego derivar el polinomio hallado. Como se tienen dos puntos, el polinomio resultante es una recta. En la figura 6.1 se pueden ver las aproximaciones de la pendiente.

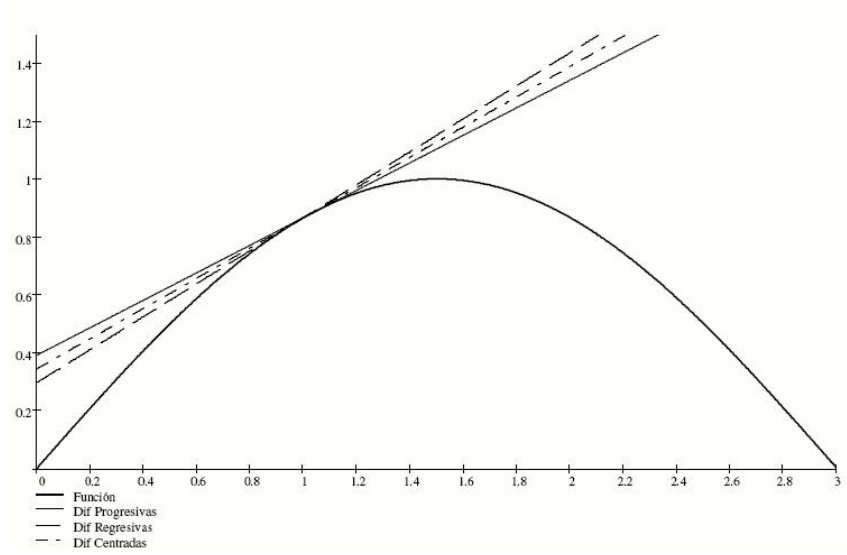


Figura 6.1: Pendiente según cada aproximación.

Hagamos una mejora escribiéndolo como:

$$f'(x) \approx \frac{f(x + \frac{h}{2}) - f(x - \frac{h}{2})}{h}$$

buscando mejorar la aproximación del resultado buscado. Reemplazando tendremos:

$$f'(1) \approx \frac{f(1,05) - f(0,95)}{0,1} = \frac{seno\left(\frac{2\pi}{6}1,05\right) - seno\left[\frac{2\pi}{6}(0,95)\right]}{0,1}$$

$$f'(1) \approx \frac{0,8910 - 0,8387}{0,1} = 0,5230$$

El resultado es una mejor aproximación pero no se nota una gran diferencia con respecto al anterior, puesto que el error cometido es del orden de 0,1 %. Pero sin lugar a dudas, este último algoritmo es mucho mejor.

Esta forma de aproximar la derivada en un punto se conoce como *aproximación por diferencias*, y se pueden clasificar según tres tipos:

1. **Diferencias progresivas**: cuando la derivada en un punto se aproxima según la expresión vista en primer término, o sea:

$$f'(x) = \frac{f(x + h) - f(x)}{h};$$

2. **Diferencias regresivas**: cuando la derivada en punto se aproxima según la expresión vista en segundo término, o sea:
$$f'(x) = \frac{f(x) - f(x-h)}{h}, \text{ y}$$

3. **Diferencias centradas**: cuando la derivada en un punto se aproxima según la expresión vista en último término, o sea:
$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}.$$

Como vimos, este último esquema es el que mejor aproxima.

Analizaremos ahora el por qué de esta mejor aproximación. Empecemos por el esquema de diferencias progresivas. Si desarrollamos por Taylor la función $f(x+h)$, tendremos que

$$f(x+h) = f(x) + f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} + f'''(x)\frac{h^3}{3!} + \ldots;$$

de la cual podemos despejar $f'(x)$, que es

$$f'(x)h = f(x+h) - f(x) - f''(x)\frac{h^2}{2!} - f'''(x)\frac{h^3}{3!} - \ldots;$$

$$f'(x) = \frac{f(x+h) - f(x)}{h} - f''(x)\frac{h}{2!} - f'''(x)\frac{h^2}{3!} - \ldots.$$

Si nuestro h es suficientemente pequeño, entonces los h^n para $n \geq 2$ se pueden despreciar. Finalmente tendremos que

$$f'(x) = \frac{f(x+h) - f(x)}{h} - f''(\xi)\frac{h}{2!} = \frac{f(x+h) - f(x)}{h} + O(h);$$

con $\xi \in [x; x+h]$. En este caso, nuestra aproximación tiene un orden de convergencia $O(h)$.

Si repetimos el proceso para el esquema de diferencias regresivas, tendremos que

$$f(x-h) = f(x) - f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} - f'''(x)\frac{h^3}{3!} + \ldots.$$

Como en el caso anterior, la expresión final será

$$f'(x) = \frac{f(x) - f(x-h)}{h} + f''(\xi)\frac{h}{2!} = \frac{f(x) - f(x-h)}{h} + O(h).$$

Al igual que en lo visto anteriormente, el orden de convergencia es $O(h)$.

Finalmente, hagamos lo mismo para el esquema de diferencias centradas. En este caso tendremos que

$$\begin{aligned} f(x+h) &= f(x) + f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} + f'''(x)\frac{h^3}{3!} + \ldots; \\ f(x-h) &= f(x) - f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} - f'''(x)\frac{h^3}{3!} + \ldots. \end{aligned}$$

Si hacemos $f(x+h) - f(x-h)$ nos queda

$$f(x+h) - f(x-h) = 2f'(x)\frac{h}{1!} + 2f'''(x)\frac{h^3}{3!} + \ldots.$$

Si despejamos $f'(x)$ de esta expresión, nos queda

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - f'''(\xi)\frac{h^2}{3!} = \frac{f(x+h) - f(x-h)}{2h} + O(h^2),$$

esta vez con $\xi \in [x-h; x+h]$.

Notemos que en este caso la convergencia es $O(h^2)$, razón por la cual la aproximación es mucho mejor respecto de los esquemas anteriores. Entonces es conveniente armar un esquema de *diferencias centradas* para aproximar una derivada en un punto dado. Además tiene otra ventaja. Como el error es proporcional a la tercera derivada, podemos obtener resultados muy precisos («exactos») para un polinomio de grado menor o igual a 2.

Al mismo tiempo, el hecho de que el orden de convergencia sea $O(h^2)$ nos permite inferir que si hacemos el paso (h) cada vez más chico, deberíamos tener un resultado con una mejor aproximación. Hagamos esto, y con la misma precisión del ejemplo anterior, calculemos de nuevo la derivada en el punto $x = 1$ con un nuevo paso, $h = 0,01$, para cada esquema.

1. Diferencias progresivas: $f'(1) = \frac{seno\left(\frac{\pi}{3}1{,}01\right) - seno\left(\frac{\pi}{3}\right)}{0{,}01} = \frac{0{,}8712 - 0{,}8660}{0{,}01} = 0,5200$

2. Diferencias regresivas: $f'(1) = \frac{seno\left(\frac{\pi}{3}\right) - seno\left(\frac{\pi}{3}0{,}99\right)}{0{,}01} = \frac{0{,}8660 - 0{,}8607}{0{,}01} = 0,5300$

3. Diferencias centradas: $f'(1) = \frac{seno\left(\frac{\pi}{3}1{,}01\right) - seno\left(\frac{\pi}{3}0{,}99\right)}{0{,}02} = \frac{0{,}8712 - 0{,}8607}{0{,}02} = 0,5250$

Al achicar el paso utilizado para reducir el error cometido podemos notar dos cosas. La primera es que para los esquemas progresivos y regresivos el resultado obtenido es más aproximado que en el caso anterior con un paso diez veces más grande, mientras que para el esquema centrado, el resultado no fue mejor. La segunda es que hemos perdido precisión, principalmente en el esquema con diferencias centradas. La pregunta es: ¿por qué? En todo caso, ¿habremos hecho algo mal?

En realidad no hemos hecho nada incorrecto. Sucede que no hemos tomado en cuenta la incidencia del error de redondeo en nuestro algoritmo, es decir, el hecho de trabajar solamente con cuatro dígitos al representar los resultados intermedios. Supusimos que achicar el paso inmediatamente nos mejoraba nuestra aproximación. Pero hemos visto que la aproximación depende también de la precisión usada en los cálculos, es decir, de la representación numérica, que como vimos, está asociada al error de redondeo [1].

El problema es que a medida que el paso h es cada vez más chico, lo mismo pasa con la operación $f(x+h) - f(x)$ o sus equivalentes. Esa diferencia se vuelve muy chica y es posible que nuestra unidad de máquina no pueda representarla correctamente. En consecuencia, debemos encontrar o desarrollar otro método para mejorar la aproximación del resultado buscado.

6.1.2. Aproximación por polinomios de Taylor

Propongamos el siguientes esquema, que se basa en tomar los intervalos $x \pm 2h$ y $x \pm h$, y el desarrollo por Taylor para cada caso:

$$\begin{aligned}
f(x+2h) &= f(x) + f'(x)\frac{2h}{1!} + f''(x)\frac{4h^2}{2!} + f'''(x)\frac{8h^3}{3!} + f^{iv}(x)\frac{16h^4}{4!} + f^{v}(x)\frac{32h^5}{5!} + \ldots; \\
f(x+h) &= f(x) + f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} + f'''(x)\frac{h^3}{3!} + f^{iv}(x)\frac{h^4}{4!} + f^{v}(x)\frac{h^5}{5!} + \ldots; \\
f(x-h) &= f(x) - f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} - f'''(x)\frac{h^3}{3!} + f^{iv}(x)\frac{h^4}{4!} - f^{v}(x)\frac{h^5}{5!} + \ldots; \\
f(x-2h) &= f(x) - f'(x)\frac{2h}{1!} + f''(x)\frac{4h^2}{2!} - f'''(x)\frac{8h^3}{3!} + f^{iv}(x)\frac{16h^4}{4!} - f^{v}(x)\frac{32h^5}{5!} + \ldots.
\end{aligned}$$

[1] En el capítulo 1 vimos como ejemplo de la incidencia del error de redondeo en un algoritmo, el cálculo de una derivada numérica, y como a medida que el paso h se hacía más chico, el error aumentaba.

Primero hagamos $f(x+2h) - f/x-2h)$ y $f(x+h) - f(x-h)$, con las cual obtendremos las siguientes igualdades:

$$\begin{aligned} f(x+2h) - f(x-2h) &= 4f'(x)\frac{h}{1!} + 16f'''(x)\frac{h^3}{3!} + 64f^v(x)\frac{h^5}{5!} + \ldots; \\ f(x+h) - f(x-h) &= 2f'(x)\frac{h}{1!} + 2f'''(x)\frac{h^3}{3!} + 2f^v(x)\frac{h^5}{5!} + \ldots. \end{aligned}$$

Si queremos mejorar la precisión de nuestros esquemas anteriores para calcular $f'(x)$, anulemos el término con h^3. Para ello, hagamos $[f(x+2h) - f(x-2h)] - 8 \times [f(x+h) - f(x-h)]$. Así, nos queda la siguiente igualdad:

$$f(x+2h) - f(x-2h) - 8f(x+h) + 8f(x-h) = -12f'(x)h + 48f^v(x)\frac{h^5}{5!} + \ldots.$$

De esta última expresión podemos despejar $f'(x)$, que resulta ser:

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + 4f^v(x)\frac{h^4}{5!} + \ldots;$$

y si truncamos en h^4, nos queda:

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + 4f^v(\xi)\frac{h^4}{5!}.$$

con $\xi \in [x-2h; x+2h]$ y un orden de convergencia $O(h^4)$. Con esta última expresión podemos decir que una aproximación de la primera derivada en un punto esta dada por:

$$f'(x) \approx \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}.$$

Ahora, apliquemos este nuevo esquema centrado para calculas la derivada buscada, con la misma representación numérica utilizada en los casos anteriores. Tomemos el paso $h = 0,1$ con el que obtendremos:

$$f'(x=1) = \frac{seno\left(\frac{\pi}{3}0,8\right) - 8 \cdot seno\left(\frac{\pi}{3}0,9\right) + 8 \cdot seno\left(\frac{\pi}{3}1,1\right) - seno\left(\frac{\pi}{3}1,2\right)}{12 \cdot h}$$

$$f'(x=1) = \frac{0,7431 - 6,4721 + 7,3084 - 0,9511}{12 \cdot 0,1} = \frac{0,6283}{1,2} = 0,5236$$

El resultado obtenido es sorprendente, pues para esa representación numérica, ¡se lo puede considerar exacto! Bastó que ampliáramos el intervalo de cálculo, es decir, los puntos que usamos para armar lo que se denomina una *malla* (en inglés *mesh*), para que la aproximación sea excelente. Este algoritmo se conoce como el *método de los cinco puntos* y tiene un orden de convergencia proporcional a la derivada quinta, lo que lo vuelve muy preciso. La única desventaja es que requiere operar con cinco puntos y esa malla deberá densificarse cada vez que la representación numérica sea más *precisa*, cuidando siempre de evitar que el paso sea muy chico, por el riesgo de que no pueda representarse correctamente el numerador. Veremos más adelante que este tipo de mallas son muy útiles para resolver ecuaciones diferenciales y/o sistemas de ecuaciones diferenciales.

En la figura 6.2 se puede la aproximación obtenida utilizando la aproximación por polinomios de Taylor.

Pero nuestro interés, por ahora, es calcular en forma numérica el valor de la derivada en un punto dado con la mejor aproximación posible. ¿Existirá otra forma de obtener ese valor con un grado de aproximación similar al obtenido con el esquema anterior usando un solo punto?

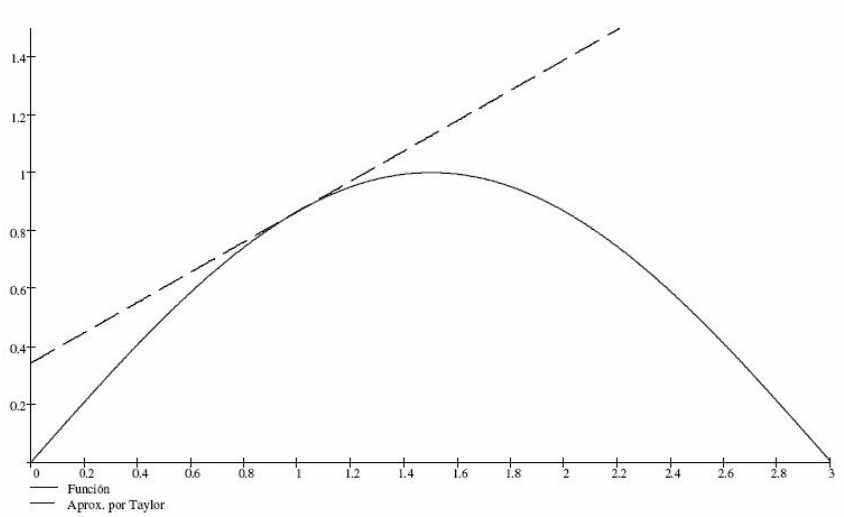


Figura 6.2: Aproximación por polinomios de Taylor.

6.1.3. Extrapolación de Richardson

Vimos en el punto anterior que para calcular una derivada en un punto y obtener la mejor aproximación, debemos trabajar con el esquema centrado y con un paso h pequeño, aún cuando esto trae aparejado una inestabilidad de los resultados. Tal como vimos al analizar la aproximación por polinomios de Taylor, nuestra aproximación de la derivada se puede expresar como:

$$M = N(h) + E(h),$$

donde M es el valor buscado, $N(h)$, la aproximación de M, $E(h)$, el error cometido y h, el paso. Supongamos que podemos expresar nuestra $E(h)$ de la siguiente forma:

$$E(h) = K_1 h + K_2 h^2 + K_3 h^3 + \dots.$$

Análogamente al caso anterior, para un h_1 el valor buscado se podrá expresar como

$$\text{(I)} \quad M = N_1(h_1) + K_1 h_1 + K_2 h_1^2 + K_3 h_1^3 + \dots.$$

Hagamos lo mismo pero para un h_2 tal que $q = \frac{h_1}{h_2}$. Entonces tendremos:

$$\text{(II)} \quad M = N_1(h_2) + K_1 h_2 + K_2 h_2^2 + K_3 h_2^3 + \dots.$$

Como $h_1 = q h_2$ podemos escribir (I) como:

$$\text{(III)} \quad M = N_1(h_1) + K_1 q h_2 + K_2 (q h_2)^2 + K_3 (q h_2)^3 + \dots.$$

Para mejorar el orden de aproximación de nuestro resultado, anulemos el término lineal de h, es decir, multipliquemos por q a (II) y luego restémosle (III):

$$\begin{aligned} qM - M &= qN_1(h_2) - N_1(h_1) + qK_1(h_2 - h_2) + qK_2\left(h_2^2 - q h_2^2\right) + qK_3\left(h_2^3 - q^2 h_2^3\right) + \dots \\ qM - M &= qN_1(h_2) - N_1(h_1) + qK_2\left(h_2^2 - q h_2^2\right) + qK_3\left(h_2^3 - q^2 h_2^3\right) + \dots. \end{aligned}$$

Si despejamos M, tendremos la siguiente expresión:

$$M = \frac{(q-1)N_1(h_2)}{q-1} + \frac{N_1(h_2) - N_1(h_1)}{q-1} - \frac{qK_2 h_2^2 (q-1)}{q-1} - \frac{qK_3 h_2^3 \left(q^2 - 1\right)}{q-1} + \dots$$

en la que podemos expresar M como:

$$M = \underbrace{N_1(h_2) + \frac{N_1(h_2) - N_1(h_1)}{q-1}}_{N_2(h_1)} - qK_2h_2^2 - q\,(q+1)\,K_3h_2^3 + \dots\ .$$

Al definir

$$N_2(h_1) = N_1(h_2) + \frac{N_1(h_2) - N_1(h_1)}{q-1},$$

nos queda que:

$$M = N_2(h) + K_2'h^2 + K_3'h^3 + \dots,$$

con

$$\begin{aligned} K_2' &= -qK_2; \\ K_3' &= -q(q+1)K_3; \\ \dots &\quad . \end{aligned}$$

Repitamos el proceso tomando nuevamente h_1 y h_2, entonces tenemos:

$$\text{(IV)}\quad M = N_2(h_1) + K_2'h_1^2 + K_3'h_1^3 + \dots.$$

$$\text{(V)}\quad M = N_1(h_2) + K_2'h_2^2 + K_3'h_2^3 + \dots.$$

Al igual que en el paso anterior, impondremos que $q = \frac{h_1}{h_2}$, y reescribamos (IV) de la siguiente forma:

$$\text{(VI)}\quad M = N_2(h_1) + K_2'\,(qh_2)^2 + K_3'\,(qh_2)^3 + \dots.$$

Análogamente al caso anterior, mejoremos nuestra aproximación anulando en este caso el término cuadrático de h, multiplicando por q^2 a (V) para luego restarle (VI):

$$\begin{aligned} q^2M - M &= q^2N_2(h_2) - N_2(h_1) - q^2K_2'\left(h_2^2 - h_2^2\right) + q^2K_3'h_2^3\,(q-1) + \dots \\ q^2M - M &= q^2N_2(h_2) - N_2(h_1) + q^2K_3'h_2^3\,(q-1) + \dots \end{aligned}$$

De la misma forma que para el caso anterior, obtenemos una nueva aproximación para M

$$M = \frac{(q^2-1)N_2(h_2)}{q^2-1} + \frac{N_2(h_2) - N_2(h_1)}{q^2-1} + \frac{q^2K_3'h_2^3\,(q-1)}{q^2-1} + \dots$$

$$M = \underbrace{N_2(h_2) + \frac{N_2(h_2) - N_2(h_1)}{q^2-1}}_{N_3(h_1)} + \frac{q^2K_3'h_2^3}{q+1} + \dots.$$

Una segunda forma de escribir esto último en función de h_1 es

$$M = \underbrace{N_2\left(\frac{h_1}{q}\right) + \frac{N_2\left(\frac{h_1}{q}\right) - N_2(h_1)}{q^2-1}}_{N_3(h_1)} + \frac{q^2K_3'\left(\frac{h_1}{q}\right)^3}{q+1} + \dots,$$

es decir, nos queda

$$M = N_3(h_1) + K_3''h_1^3 + \dots;$$

con

$$K_3'' = \frac{K_3'}{q(q+1)}; \dots,$$

aproximación que resulta mejor que la anterior.

Finalmente, podemos generalizar el método de aproximación de la siguiente forma:

$$N_j(h) = N_{j-1}\left(\frac{h}{q}\right) + \frac{N_{j-1}\left(\frac{h}{q}\right) - N_{j-1}(h)}{q^{j-1} - 1}.$$

Este método o algoritmo para mejorar una aproximación se conoce como *Extrapolación de Richardson*. Veremos más adelante una aplicación de este mismo método asociado a la integración numérica.

Un caso particular muy usado es cuando $q = 2$, cuya expresión general se define como:

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}\left(\frac{h}{2}\right) - N_{j-1}(h)}{2^{j-1} - 1}.$$

Este algoritmo permite aproximar una derivada numérica con poco esfuerzo y teniendo en cuenta la inestabilidad del algoritmo porque no requiere dividir por números excesivamente pequeños.

Apliquemos este método al ejemplo inicial y calculemos la derivada de $f(x) = seno\left(\frac{\pi}{3}x\right)$ en $x = 1$ con el algoritmo de diferencias progresivas.

Armemos una tabla para aplicar el algoritmo anterior de modo de visualizar fácilmente cada uno de los pasos. En primer lugar, vamos definir que la primera aproximación, es decir, $N_1(h)$ sea la derivada calculada numéricamente con h, que ocupará la primera columna. Usaremos la expresión:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} = \frac{seno\left[\frac{\pi}{3}(x+h)\right] - seno\left[\frac{\pi}{3}x\right]}{h}.$$

Las demás columnas serán $N_2(h)$, $N_3(h)$ y $N_4(h)$. En segundo lugar, tomaremos varios valores de h, por lo tanto tendremos varias filas con diferentes aproximaciones de la derivada buscada. Para cada caso calcularemos las aproximaciones con la fórmula de la Extrapolación de Richardson:

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}\left(\frac{h}{2}\right) - N_{j-1}(h)}{2^{j-1} - 1}.$$

En la tabla 6.1 tenemos los resultados obtenidos al aplicar la extrapolación de Richardson a nuestro ejemplo.

Tabla 6.1: Extrapolación de Richardson

$\mathbf{h_i}$	$\mathbf{y'_i = N_1}$	$\mathbf{N_2}$	$\mathbf{N_3}$	$\mathbf{N_4}$
0,2	0,4250			
0,1	0,4750	0,5250		
0,05	0,5000	0,5250	0,5250	
0,025	0,5120	0,5240	0,5237	0,5235

Analicemos rápidamente los resultados obtenidos. La primera columna contiene los resultados de aproximar la derivada con varios h diferentes. Vemos que a pesar de utilizar un h relativamente pequeño ($h = 0,025$) nuestra aproximación inicial no es muy buena.

La segunda columna es nuestra primera aplicación de la extrapolación de Richardson, usando los valores de la primera columna. A primera vista se puede observar que la aproximación es muy superior a la anterior. Algo similar ocurre en la tercera. Finalmente, en la cuarta, la aproximación final resulta ser casi el valor «exacto» para una representación de cuatro (4) decimales. Y si comparamos con la aproximación para $h = 0,025$, última fila de la primera columna, vemos que es muy superior. Si quisiéramos obtener una aproximación similar, deberíamos trabajar con más decimales, puesto que para $h = 0,01$ el valor de $f'(1)$ es 0,5200, que si bien tiene dos decimales correcto, es menos preciso que el hallado con Richardson.

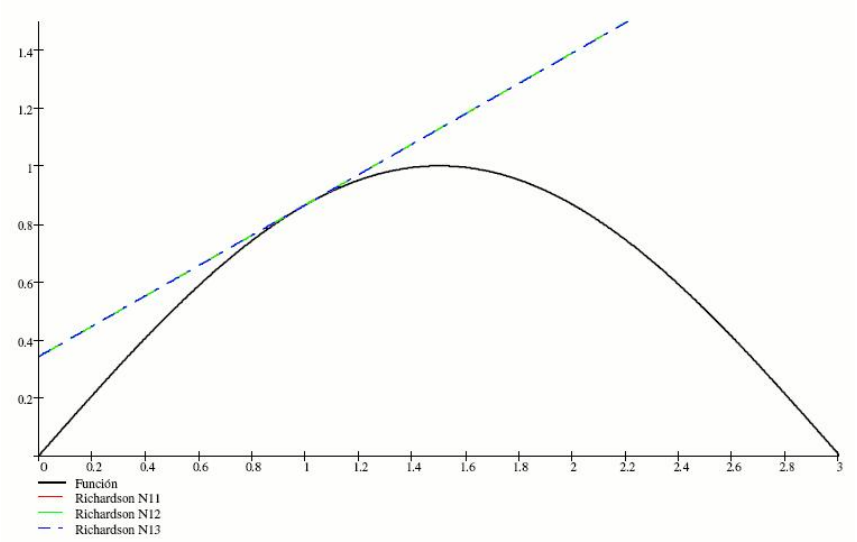


Figura 6.3: Aproximación con N_1.

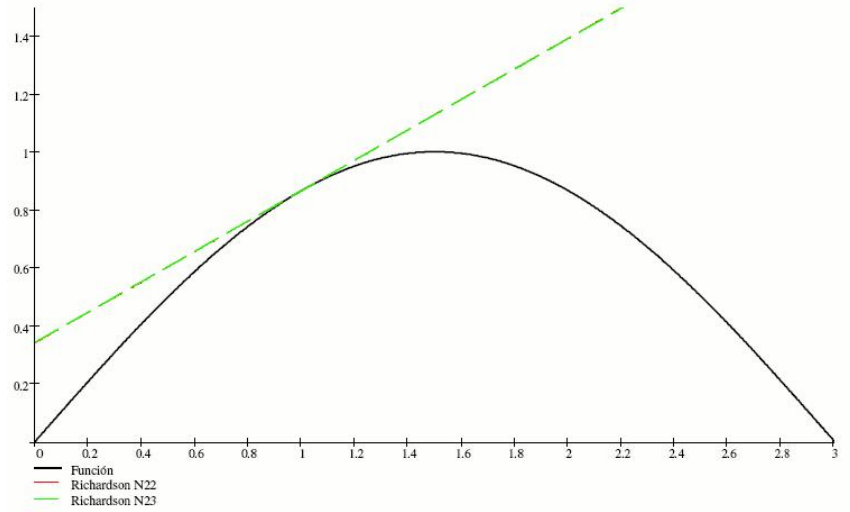


Figura 6.4: Aproximación con N_2.

En las figuras 6.3, 6.4 y 6.5 se pueden ver algunas de las aproximaciones de la pendiente en el punto dado en cada paso.

Si bien aplicamos este método para obtener una derivada numérica, puede aplicarse para cualquier caso que cumpla la condición:

$$M = N(h) + K_1 h + K_2 h^2 + K_3 h^3 + \dots.$$

como es el caso de la interpolación polinomial.

6.1.4. Notas finales

Es evidente que la diferenciación numérica es inestable o, dicho de otro modo, es muy dependiente de la precisión utilizada. Afinar el paso h en un algoritmo dado puede conducir a resultados de menor precisión o, en términos numéricos, inservibles; en consecuencia, no es conveniente reducir el paso h suponiendo que eso mejora la aproximación buscada.

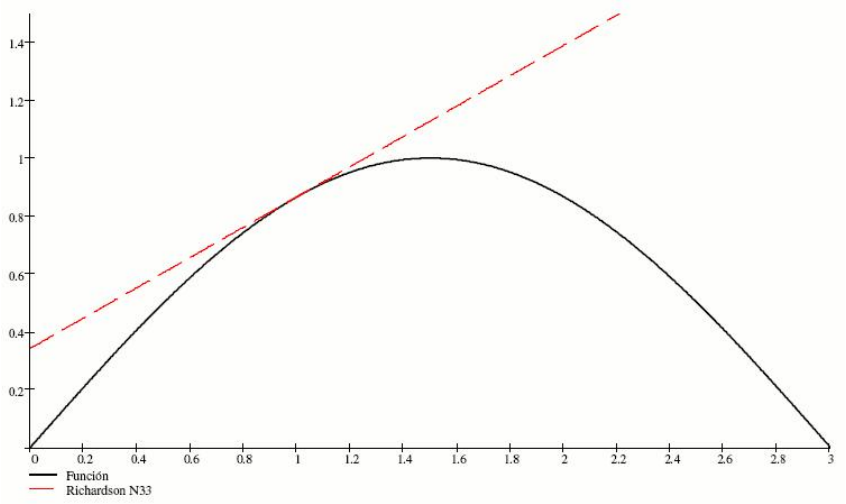


Figura 6.5: Aproximación con N_3.

Los distintos métodos vistos en los puntos anteriores indican que es preferible mejorar el algoritmo o desarrollar uno nuevo, antes que afinar el paso de cálculo. Más aún, es mucho más efectivo aplicar el método de extrapolación de Richardson a un algoritmo conocido y sencillo que desarrollar uno nuevo. En todo caso, la segunda opción sería utilizar los polinomios de Taylor o alguna aproximación polinomial que utilice la información disponible (puntos adyacentes o aledaños). Si bien esta aproximación puede ser laboriosa, queda ampliamente justificada al disminuir la incidencia del error de redondeo en los cálculos, sobre todo al no tener que dividir por un número «muy pequeño».

Los desarrollos vistos para el caso de aproximar una primera derivada pueden extrapolarse para derivadas de orden superior. Un ejemplo de ello es la aproximación centrada de la segunda derivada en un punto dado, cuya expresión es:

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}.$$

que se obtiene de considerar los polinomios de Taylor para $x-h$ y $x+h$. Efectivamente, al desarrollar ambos polinomios tendremos

$$\begin{aligned} f(x+h) &= f(x) + f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} + f'''(x)\frac{h^3}{3!} + \ldots; \\ f(x-h) &= f(x) - f'(x)\frac{h}{1!} + f''(x)\frac{h^2}{2!} - f'''(x)\frac{h^3}{3!} + \ldots. \end{aligned}$$

Si sumamos ambos polinomios obtenemos:

$$\begin{aligned} f(x+h) + f(x-h) &= 2f(x) + f''(x)h^2 + f^{iv}(x)\frac{h^4}{12} + \ldots; \\ f''(x) &= \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} - f^{iv}[\xi]\frac{h^2}{12}; \\ f''(x) &= \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + O(h^2), \end{aligned}$$

con $\xi \in [x-h; x+h)$.

Observemos que el error cometido al calcular la derivada segunda con la expresión dada es proporcional a h^2 y a $f^{iv}(\xi)$, es decir, similar al caso de la expresión centrada para la primera derivada. Podemos asegurar que en el caso de polinomios de grado 3 o inferior, o que no exista la derivada cuarta, la derivada segunda obtenida en forma numérica, es «exacta».

Mediante razonamientos análogos o similares pueden obtenerse algoritmos para calcular derivadas numéricas de orden superior.

6.2. Integración numérica

Como en el caso de la diferenciación numérica, la integración numérica tiene la misma dificultad de trabajar con métodos simbólicos. Existen muchos programas de aplicación en la ingeniería que dependen de obtener integrales definidas. Como es prácticamente imposible agregar una base de datos que incluya las primitivas de cualquier función, la única manera de calcular estas integrales es mediante métodos numéricos. Un ejemplo en este sentido es la utilización del método de los elementos finitos en el análisis estructural, que calcula la matriz de rigidez mediante la integración numérica.

Veremos a continuación varios métodos numéricos para calcular integrales definidas, analizando ventajas y desventajas de cada uno de ellos.

6.2.1. Fórmulas de Newton-Cotes

Antes de desarrollar las distintas fórmulas o métodos para obtener una integral definida en forma numérica, daremos algunas definiciones.

Definición 6.1. Dada una función $f(x)$ definida en $[a;b]$, se denomina *cuadratura numérica* de la integral $I(f) = \int_a^b f(x)\mathrm{d}x$ a una fórmula tal que:

$$Q_n(f) = \sum_{i=1}^{n} c_i f(x_i);$$

con $c_i \in \Re$ y $x_i \in [a;b]$. Los puntos x_i se denominan *puntos de cuadratura* (o raíces) y los valores c_i, coeficientes de *cuadratura o de peso*. Asimismo, se define el error de la cuadratura como $E_n(f) = I(f) - Q_n(f)$.

Definición 6.2. Una cuadratura numérica tiene grado de precisión m si $E_n(x^k) = 0$ para $k = 0;1;\ldots;m$ y $E_n\left(x^{m+1}\right) \neq 0$.

Observación 6.2.1. Si una cuadratura numérica tiene grado de precisión m, entonces $E_n(p_k) = 0$ para todo polinimio $p_k(x)$ de grado menor o igual a m $(k \leq m)$.

Definición 6.3. Se denomina *fórmula cerrada* de Newton-Cotes a toda cuadratura numérica cuyos nodos incluya a los extremos del intervalo.

Definición 6.4. Se denomina *fórmula abierta* de Newton-Cotes a toda cuadratura numérica cuyos nodos no incluya a los extremos del intervalo.

6.2.2. Fórmulas cerradas de Newton-Cotes

Fórmulas simples

Supongamos que tenemos la siguiente función (o curva) y queremos hallar el área bajo la misma en el intervalo $[a;b]$, como se ve en la figura 6.6.

Para empezar, podemos hacer dos aproximaciones muy groseras como se puede apreciar en las figuras 6.7(a) y 6.7(b):

En la aproximación de la figura 6.7(a), vemos que el área obtenida es mucho menor que el área buscada. En cambio, en la 6.7(b), podríamos suponer que la aproximación obtenida del área es similar o mayor. Podemos ver que si el área en color claro se compensa con el área en color oscuro excedente, entonces estaríamos obteniendo una buena aproximación. Si esto no fuera así,

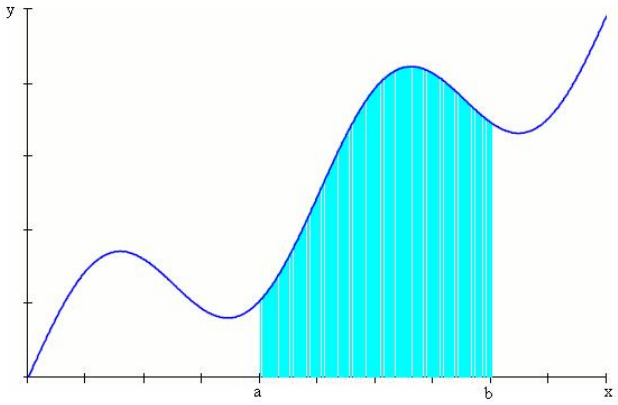
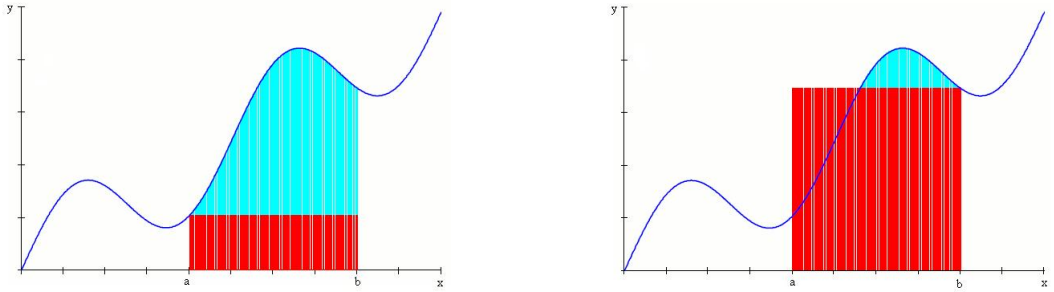


Figura 6.6: Área bajo la curva.



(a) Por defecto (b) Por exceso

Figura 6.7: Aproximación por rectángulos.

entonces obtendríamos una área por defecto (la parte oscura es menor que la parte clara) o por exceso (la parte oscura es mayor a la parte clara).

Estas dos aroximaciones se pueden expresar matemáticamente como:

$$Q_n(f) = f(a)(b-a);$$

para el caso (a) y,

$$Q_n(f) = f(b)(b-a);$$

para el caso (b).

Otra forma de aproximar el área consiste en la que vemos en la figura 6.8:

En este caso particular, no parece que esta aproximación sea mucho mejor, puesto que hay un área excedente en color claro. La expresión matemática para este caso es:

$$Q_n(f) = \frac{f(b)+f(a)}{2}(b-a).$$

Vamos a generalizar estas tres expresiones. Definamos $h = b - a$, y escribamos cada una de las expresiones de la siguiente forma:

- Aproximación por rectángulos (defecto): $Q_n(f) = h \cdot f(a)$;
- Aproximación por rectángulos (exceso): $Q_n(f) = h \cdot f(b)$;
- Aproximación por trapecio: $Q_n(f) = \frac{h}{2} \cdot [f(a) + f(b)]$.

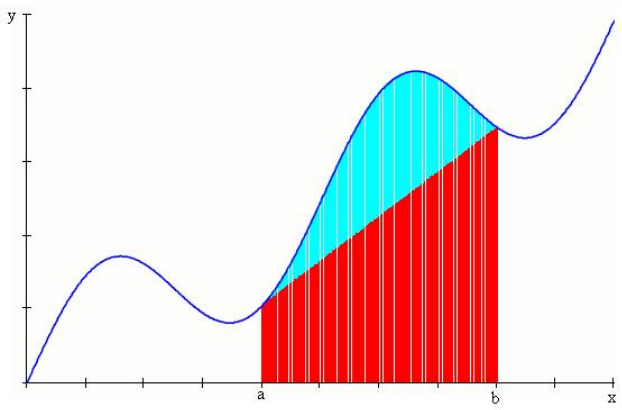


Figura 6.8: Aproximación por trapecios.

Para saber si nuestras aproximaciones son buenas, estimemos el error que cometemos en cada una. Primeramente, analicemos cualquiera de los dos métodos que aproximan por un rectángulo. Al desarrollar $f(x)$ respecto del punto a mediante una serie de Taylor, tenemos que

$$f(x) = f(a) + f'(a)(x-a) + f''(a)\frac{(x-a)^2}{2} + \dots.$$

Para obtener la integral basta con integrar la serie también. Entonces tenemos que

$$\int_a^b f(x)\mathrm{d}x = \int_a^b f(a)\mathrm{d}x + \int_a^b f'(a)(x-a)\mathrm{d}x + \int_a^b f''(a)\frac{(x-a)^2}{2}\mathrm{d}x + \dots.$$

Si integramos y truncamos en el término de la derivada primera, nos queda

$$\int_a^b f(x)\mathrm{d}x = f(a)(b-a) + f'(\xi)\frac{(b-a)^2}{2},$$

con $\xi \in [a;b]$. Como $h = b-a$ nos queda

$$\int_a^b f(x)\mathrm{d}x = f(a)h + f'(\xi)\frac{h^2}{2} = h\cdot f(a) + \frac{b-a}{2}\underbrace{f'(\xi)\cdot h}_{O(h)},$$

es decir, nuestra expresión tiene un error proporcional a la derivada primera y su orden de convergencia es $O(h)$. Para el caso de usar $f(b)$ el error es análogo.

Para analizar el método del trapecio, usemos una interpolación entre el punto a y b usando el polinomio de Lagrange y su error [2]. En este caso tenemos que

$$f(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a} + f''(\xi)\frac{(x-a)(x-b)}{2}.$$

Integremos el polinomio obtenido; así nos queda

$$\begin{aligned}
\int_a^b f(x)\mathrm{d}x &= \frac{f(a)}{a-b}\int_a^b (x-b)\mathrm{d}x + \frac{f(b)}{b-a}\int_a^b (x-a)\mathrm{d}x + f''(\xi)\int_a^b \frac{(x-a)(x-b)}{2}\mathrm{d}x \\
&= \frac{f(a)+f(b)}{2}(b-a) - f''(\xi)\frac{(b-a)^3}{12} \\
&= \frac{f(a)+f(b)}{2}h - f''(\xi)\frac{h^3}{12} \\
\int_a^b f(x)\mathrm{d}x &= \frac{f(a)+f(b)}{2}h - \frac{b-a}{2}f''(\xi)\frac{h^2}{6},
\end{aligned}$$

[2] Otro camino es truncar el desarrollo por Taylor del rectángulo en el término que contiene a $f''(a)$.

nuevamente con $\xi \in [a;b]$. Lo que hemos obtenido es un método cuyo error es proporcional a la derivada segunda, o sea, mejoramos nuestra aproximación.

Analicemos ahora una segunda mejora. Supongamos que podemos calcular la función en $x = \frac{a+b}{2}$, es decir, podemos obtener $f\left(\frac{a+b}{2}\right)$. En consecuencia, tenemos ahora tres puntos que nos pueden servir para obtener el área buscada. Hagamos pasar una curva por esos tres puntos utilizando el polinomio de Taylor y asumiendo en este caso que $h = \frac{b-a}{2}$, como se ve en la figura 6.9. Podemos ver en la figura que el área aproximada es mayor que el área buscada, lo que

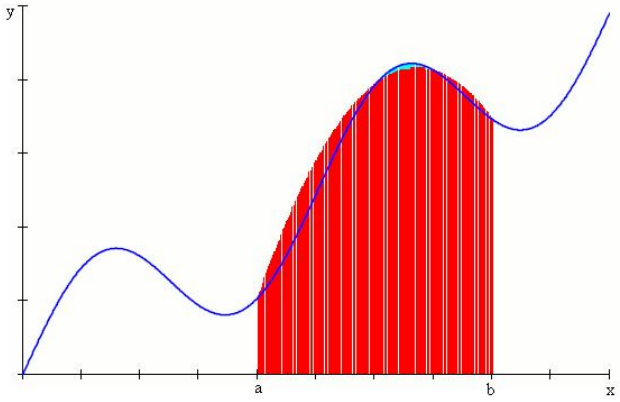


Figura 6.9: Aproximación por arcos de parábola cuadrática.

significa que obtendremos un valor por exceso.

La aproximación usando parábolas de segundo grado es la conocida fórmula de Simpson, cuya expresión matemática es:

$$Q_n(f) = \frac{h}{3}\left[f(a) + f(b) + 4 \cdot f\left(\frac{a+b}{2}\right)\right].$$

Analicemos el error que se comete con esta nueva expresión. Tomemos nuevamente nuestro desarrollo de Taylor pero a partir del punto $x_1 = \dfrac{a+b}{2}$ y cortemos la expresión en la derivada cuarta. Entonces nos queda

$$f(x) = f(x_1) + f'(x_1)(x-x_1) + f''(x_1)\frac{(x-x_1)^2}{2} + f'''(x_1)\frac{(x-x_1)^3}{6} + f^{iv}(\xi_1)\frac{(x-x_1)^4}{24},$$

con $\xi_1 \in [a,b]$.

Si integramos nuevamente nos queda

$$\begin{aligned}
\int_a^b f(x)\mathrm{d}x &= \int_a^b f(x_1)\mathrm{d}x + \int_a^b f'(x_1)(x-x_1)\mathrm{d}x + \int_a^b f''(x_1)\frac{(x-x_1)^2}{2}\mathrm{d}x + \\
&\quad + \int_a^b f'''(x_1)\frac{(x-x_1)^3}{6}\mathrm{d}x + \int_a^b f^{iv}(x_1)\frac{(x-x_1)^4}{24}\mathrm{d}x \\
&= f(x_1)(b-a) + f'(x_1)\left.\frac{(x-x_1)^2}{2}\right|_a^b + f''(x_1)\left.\frac{(x-x_1)^3}{6}\right|_a^b + \\
&\quad + f'''(x_1)\left.\frac{(x-x_1)^4}{24}\right|_a^b + f^{iv}(\xi)\left.\frac{(x-x_1)^5}{120}\right|_a^b
\end{aligned}$$

Ahora tomemos que $h = b - x_1 = x_1 - a$. Entonces nos queda

$$\int_a^b f(x)\mathrm{d}x = f(x_1)2h + f''(x_1)\frac{h^3}{3} + f^{iv}(\xi)\frac{h^5}{60}.$$

Aproximemos la derivada segunda en x_1 mediante una derivada discreta, como la vista en diferenciación numérica, cual es:

$$f''(x_1) = \frac{f(a) - 2f(x_1) + f(b)}{h^2} - \frac{h^2}{12} f^{iv}(\xi_2),$$

con $\xi_2 \in [a; b]$.

Al reemplazarla en la fórmula de integración, nos queda

$$\begin{aligned}
\int_a^b f(x)\mathrm{d}x &= f(x_1)2h + \frac{f(a) - 2f(x_1) + f(b)}{h^2}\frac{h^3}{3} - f^{iv}(\xi_2)\frac{h^5}{36} + f^{iv}(\xi_1)\frac{h^5}{60} \\
&= \frac{h}{3}[f(a) + 4f(x_1) + f(b)] - f^{iv}(\xi)\frac{h^5}{90} \\
&= \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] - \frac{b-a}{2}\frac{f^{iv}(\xi)}{90}h^4,
\end{aligned}$$

con $\xi \in [a; b]$. Usualmente el término de error de define como:

$$E(h) = M\, h^4, \text{ con } M = \frac{b-a}{180} f^{iv}(\xi),$$

de ahí que el orden de convergencia sea $O(h^4)$.

Vemos que el error del método de Simpson es proporcional a la derivada cuarta, por lo tanto, esta expresión nos da una integral «exacta» para polinomios de grado menor o igual a tres.

Unifiquemos los cuatro casos en un intervalo de integración. Si tomamos como intervalo $[a; b]$ el intervalo $[-1; 1]$, nos queda para cada método lo siguiente:

- Aproximación por rectángulo (defecto): $Q_n(x) = 2 \cdot f(-1)$.
- Aproximación por rectángulo (exceso): $Q_n(x) = 2 \cdot f(1)$.
- Aproximación por trapecios: $Q_n(x) = 1 \cdot f(-1) + 1 \cdot f(1)$.
- Aproximación por Simpson: $Q_n(x) = \frac{1}{3} \cdot f(-1) + \frac{4}{3} \cdot f(0) + \frac{1}{3} \cdot f(1)$.

Si nos fijamos en la definición de cuadratura podemos ver que hemos definido para cada caso un valor de c_i y un valor de x_i, que son los siguientes:

- Aproximación por rectángulo (defecto): $c_1 = 2$, $x_1 = -1$.
- Aproximación por rectángulo (exceso): $c_1 = 2$, $x_1 = 1$.
- Aproximación por trapecios: $c_1 = c_2 = 1$, $x_1 = -1$, $x_2 = 1$.
- Aproximación por Simpson: $c_1 = c_3 = \frac{1}{3}$, $c_2 = \frac{4}{3}$, $x_1 = -1$; $x_2 = 0$; $x_3 = 1$;

con lo cual podemos escribirlos según la forma general definida como *cuadratura numérica*:

$$Q_n(f) = \sum_{i=1}^{n} c_i f(x_i);$$

siendo $n = 1$ para la fórmula del rectángulo, $n = 2$ para la del trapecio y $n = 3$ para la de Simpson.

Aún cuando estas aproximaciones tienen una precisión interesante (sobre todo la última), no son lo suficientemente precisas para resolver cualquier problema. Para mejorar nuestra aproximación, veremos a continuación algunas formas de mejorar la precisión de las cuadraturas.

Fórmulas compuestas

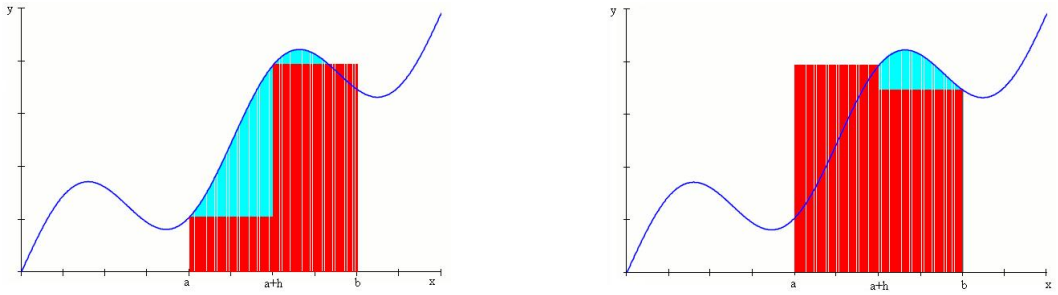
Supongamos que en lugar de utilizar la fórmula del rectángulo con el paso $h = b - a$, dividimos ese intervalo en intervalos más chicos. Empecemos por definir un nuevo paso más chico, tomando $h = \frac{b-a}{2}$. Ahora podemos aproximar la integral con dos subintervalos, tanto por defecto como por exceso, que resultan ser $[a; a+h]$ y $[a+h; b]$, con los cuales se obtienen las siguientes aproximaciones:

$$Q_n(f) = h \cdot f(a) + h \cdot f(a+h);$$

o

$$Q_n(f) = h \cdot f(a+h) + h \cdot f(b).$$

Ambas aproximaciones se pueden ver en las figuras 6.10(a) y 6.10(b). La primera es una aproximación francamente por defecto, en cambio, en la segunda tenemos una primer intervalo con una aproximación por exceso y otro intervalo por defecto; en conjunto podemos inferir que la aproximación resulta ser por exceso.



(a) Por defecto (b) Por exceso

Figura 6.10: Aproximación compuesta por rectángulos.

Si hacemos un desarrollo similar con la fórmula del trapecio, tomando el mismo paso ($h = \frac{b-a}{2}$), y por ende, los mismos subintervalos, tendremos:

$$Q_n(f) = \frac{h}{2}\left[f(a) + f(a+h)\right] + \frac{h}{2}\left[f(a+h) + f(b)\right] = \frac{h}{2}\left[f(a) + 2f\left(\frac{a+b}{2}\right) + f(b)\right].$$

La aproximación obtenida se puede ver en el figura 6.11, que resulta ser una aproximación por defecto.

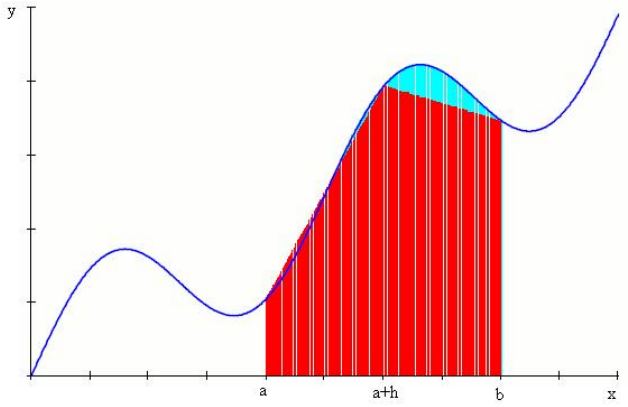


Figura 6.11: Aproximación compuesta por trapecios.

Al igual que en los casos anteriores, podemos mejorar la aproximación de la fórmula de Simpson usando la misma técnica. Si dividimos nuestro intervalo inicial en dos, de manera de trabajar con dos subintervalos y definimos $h = \frac{b-a}{4}$, tendremos la nueva aproximación:

$$Q_n(f) = \frac{h}{3}\left[f(a) + f(a+2h) + 4 \cdot f(a+h)\right] + \frac{h}{3}\left[f(a+2h) + f(b) + 4 \cdot f(a+3h)\right].$$

Podemos simplificar la expresión para que nos quede una más general:

$$Q_n(f) = \frac{h}{3}\left[f(a) + f(b) + 2 \cdot f(a+2h) + 4 \cdot f(a+h)\right].$$

El resultado de aplicar esta fórmula, como se puede ver en la figura 6.12, muestra que la aproximación obtenida es muy precisa, y que el resultado es muy cercano al «exacto».

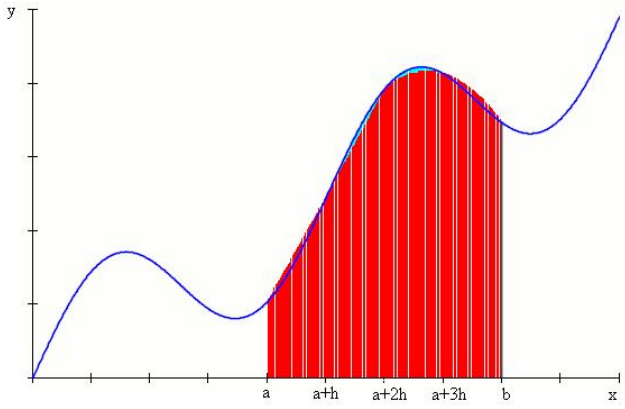


Figura 6.12: Aproximación compuesta por Simpson.

Podemos generalizar las expresiones de los métodos para n subintervalos:

- Rectángulos:

$$Q_n(f) = h \cdot \sum_{i=0}^{n-1} f(a + i \cdot h),$$

$$Q_n(f) = h \cdot \left[\sum_{i=1}^{n-1} f(a + i \cdot h) + f(b)\right],$$

  con $h = \frac{b-a}{n}$;

- Trapecios:

$$Q_n(f) = \frac{h}{2}\left[f(a) + f(b) + 2 \cdot \sum_{i=1}^{n-1} f(a + i \cdot h)\right],$$

  también con $h = \frac{b-a}{n}$; y

- Simpson:

$$Q_n(f) = \frac{h}{3}\left\{f(a) + f(b) + 2 \cdot \sum_{i=1}^{n-1} f(a + 2i \cdot h) + 4 \cdot \sum_{i=1}^{n} f[a + (2i-1) \cdot h]\right\}$$

  con $h = \frac{b-a}{2n}$ y $n = 1; 2; 3; \ldots; k$.

Estas fórmulas permiten mejorar la precisión reduciendo el paso h. En particular, en el caso del método compuesto de Simpson, el error se define como

$$E(h) = \frac{b-a}{180} f^{iv}(\mu) h^4 = M\, h^4, \text{ con } M = \frac{b-a}{180} f^{iv}(\mu),$$

con $\mu \in [a; b]$. Si bien se trata de una mejora en la precisión, la misma no es demasiado significativa, pues el orden de convergencia sigue siendo $O(h^4)$.

Sin embargo, esta metodología tiene una desventaja. A medida que achicamos el paso aumentamos notablemnte la cantidad de operaciones que debemos realizar, lo que significa más tiempo de procesamiento. Esto no siempre es práctico; por ejemplo, dividir el intervalo para Simpson en 100 subintervalos representa un esfuerzo de cálculo que no siempre mejora la precisión del resultado en el mismo sentido. Puede ocurrir que nuestra representación numérica nos limite el tamaño del paso h, lo que nos impide «afinar» el paso todo lo necesario. Algo similar puede ocurrir con las otras fórmulas.

Por otro lado, toda vez que querramos afinar nuestro cálculo reduciendo el paso h, debemos calcular prácticamente todo otra vez, pues salvo los valores de la función en los extremos del intervalo, el resto de los valores no suelen ser útiles (salvo excepciones). Cambiar el paso no suele tener «costo cero». Busquemos, en consecuencia, otra forma para obtener resultados más precisos sin tener achicar el paso, incrementar demasiado las cantidad de operaciones a realizar o repetir todos los cálculos.

Método de Romberg

Como primer paso para desarrollar un método más eficiente que mejore nuestros resultados, analicemos el error que se comete al aplicar cualquiera de las fórmulas de cuadratura vistas en los puntos anteriores. En forma general, la aproximación se puede expresar de la siguiente forma:

$$\begin{aligned} I(f) = \int_a^b f(x)\mathrm{d}x &= \int_a^b \sum_{i=1}^{n} f(x_i) L_i(x)\mathrm{d}x + \int_a^b \frac{f^{(n)}[\xi(x)]}{n!} \prod_{i=1}^{n}(x - x_i)\mathrm{d}x \\ &= \underbrace{\sum_{i=1}^{n} c_i f(x_i)}_{Q_n(f)} + \frac{1}{n!}\int_a^b f^{(n)}[\xi(x)] \prod_{i=1}^{n}(x - x_i)\mathrm{d}x; \end{aligned}$$

como vimos al principio, el error está dado por:

$$E_n(f) = I(f) - Q_n(f) = \frac{1}{n!}\int_a^b f^{(n)}[\xi(x)] \prod_{i=1}^{n}(x - x_i)\mathrm{d}x.$$

Para cada uno de los métodos tenemos:

Rectángulos: $E_1(f) = \frac{b-a}{2} \cdot f'(\xi)\, h.$

Trapecios: $E_2(f) = -\frac{b-a}{12} \cdot f''(\xi) h^2.$

Simpson: $E_3(f) = -\frac{b-a}{90} f^{iv}(\xi) h^4.$

Podemos notar que las aproximaciones mediante cualquiera de las fórmulas vistas se pueden expresar como:

$$\begin{aligned} M &= N(h) + K_1 \cdot h + K_2 \cdot h^2 + K_3 \cdot h^3 + \ldots \\ M - N(h) = E(h) &= K_1 \cdot h + K_2 \cdot h^2 + K_3 \cdot h^3 + \ldots \end{aligned}$$

lo que nos permite aplicar el método de *extrapolación de Richardson*, visto para diferenciación numérica. En el caso particular del método compuesto del trapecio, el error puede expresarse mediante potencias pares de h:

$$E(h) = K_1 \cdot h^2 + K_2 \cdot h^4 + \ldots + K_s \cdot h^{2s} + O(h^{2s+1}).$$

Esto nos induce a generar una adaptación de este método a la integración, que se conoce como **método de Romberg**. El desarrollo para obtenerlo es el siguiente. Partamos de la fórmula compuesta del trapecio:

$$Q_n(f) = \frac{h}{2}\left[f(a) + f(b) + 2 \cdot \sum_{i=1}^{n-1} f(a + i \cdot h)\right];$$

y de acuerdo con lo visto, definimos que:

$$I(f) = \frac{h}{2}\left[f(a) + f(b) + 2 \cdot \sum_{i=1}^{n-1} f(a + i \cdot h)\right] - \frac{b-a}{12}h^2 f''(\xi);$$

con $a < \xi < b$ y $h = \frac{b-a}{n}$.

En primer lugar, vamos obtengamos todas las aproximaciones para $m_1 = 1, m_2 = 2, m_3 = 4, \ldots, m_n = 2^{n-1}$, con n positivo. En consecuencia, tendremos un h_k para cada valor de m_k que estará definido por $h_k = \frac{b-a}{m_k} = \frac{b-a}{2^{k-1}}$. De esta forma podemos expresar la regla del trapecio como:

$$I(f) = \frac{h_k}{2}\left[f(a) + f(b) + 2 \cdot \sum_{i=1}^{2^{k-1}-1} f(a + i \cdot h_k)\right] - \frac{b-a}{12}h_k^2 f''(\xi_k).$$

Vamos a definir ahora que:

$$R_{k,1}(h_k) = \frac{h_k}{2}\left[f(a) + f(b) + 2 \cdot \sum_{i=1}^{2^{k-1}-1} f(a + i \cdot h_k)\right];$$

y con esta nueva fórmula vamos a obtener los distintos $R_{k,1}$. En efecto, para $k = 1$ tenemos que

$$R_{1,1} = \frac{h_1}{2}\left[f(a) + f(b)\right] = \frac{b-a}{2}\left[f(a) + f(b)\right],$$

con $h_1 = b - a$. Para el caso de $k = 2$ tenemos que

$$\begin{aligned}
R_{2,1} &= \frac{h_2}{2}\left[f(a) + f(b) + 2f(a + h_2)\right] \\
&= \frac{b-a}{4}\left[f(a) + f(b) + 2f\left(a + \frac{b-a}{2}\right)\right] \\
&= \frac{1}{2}\left[\underbrace{\frac{b-a}{2}\left(f(a) + f(b)\right)}_{R_{1,1}} + \overbrace{\frac{b-a}{2}}^{h_1} 2f(a + h_2)\right] \\
&= \frac{1}{2}\left[R_{1,1} + h_1 f(a + h_2)\right],
\end{aligned}$$

con $h_2 = \dfrac{b-a}{2}$. Análogamente, para $k = 3$, $h_3 = \dfrac{b-a}{4}$ y, entonces

$$\begin{aligned}
R_{3,1} &= \frac{h_3}{2}\Big\{f(a) + f(b) + 2\big[f(a + h_3) + f(a + \underbrace{2h_3}_{h_2}) + f(a + 3h_3)\big]\Big\} \\
&= \frac{b-a}{8}\left\{f(a) + f(b) + 2f(a + h_2) + 2\left[f(a + h_3) + f(a + 3h_3)\right]\right\} \\
&= \frac{1}{2}\left\{R_{2,1} + h_2\left[f(a + h_3) + f(a + 3h_3)\right]\right\}.
\end{aligned}$$

Si generalizamos para todos los k, tenemos que

$$R_{k,1} = \frac{1}{2}\left\{ R_{k-1;1} + h_{k-1} \sum_{i=1}^{2^{k-2}} f[a + (2i-1)h_k] \right\}.$$

Cada uno de estos $R_{k,1}$ son aproximaciones de nuestro valor buscado. Para refinar estos resultados podemos aplicar, ahora sí, la extrapolación de Richardson con $q = 4$. Por lo tanto tendremos que:

$$R_{k,2} = R_{k,1} + \frac{R_{k,1} - R_{k-1;1}}{4^1 - 1};$$

con $k = 2; 3; \ldots; n$. Si generalizamos, obtenemos la siguiente expresión:

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1;j-1}}{4^{j-1} - 1};$$

con $k = 2; 3; \ldots; n$ y $j = 2; 3; \ldots; k$. Al aplicar este método, generamos una tabla como la 6.2, donde cada $R_{k,j}$ es una mejor aproximación del resultado, siendo la mejor el $R_{n,n}$.

Tabla 6.2: Método de Romberg

$\mathbf{R_{1,i}}$	$\mathbf{R_{2,i}}$	$\mathbf{R_{3,i}}$	$\ldots$	$\mathbf{R_{n,i}}$
$R_{1,1}$				
$R_{2,1}$	$R_{2,2}$			
$R_{3,1}$	$R_{3,2}$	$R_{3,3}$		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$R_{n,1}$	$R_{n,2}$	$R_{n,3}$	$\ldots$	$R_{n,n}$

La ventaja de este método es que permite calcular una nueva fila con sólo hacer una aplicación de la fórmula compuesta del trapecio y luego usar los valores ya calculados para obtener el resto de los valores de las demás columnas de esa nueva fila; no requiere recalcular todo.

Una cuestión a tener en cuenta al aplicar este método, es que supone que la fórmula compuesta del trapecio permite la aplicación de la extrapolación de Richardson, esto es, se debe cumplir que $f(x) \in C^{2(k+1)}[a, b]$; si esto no se cumple, no tiene sentido seguir afinando el resultado hasta la iteración k. Si generalizamos, es evidente que una función $f(x)$ que cumpla con tener infinitas derivadas continuas en el intervalo $[a; b]$, es una función a la cual resulta muy conveniente aplicarle el método de Romberg.

6.2.3. Fórmulas abiertas de Newton-Cotes

En los puntos anteriores hemos visto las fórmulas cerradas para integrar numéricamente. Existen también fórmulas abiertas de Newton-Cotes. La más conocida es la del punto medio. Supongamos que tomamos la fórmula del rectángulo pero en lugar de aproximar el área con los extremos, tomamos el punto medio del intervalo, es decir, $c = \frac{a+b}{2}$. En ese caso nuestra aproximación del área buscada estraá dada por:

$$Q_n(f) = (\underbrace{b-a}_{h}) \cdot f(c) = h \cdot f(c).$$

La aproximación efectuada con esta fórmula se puede ver en la figura 6.13.

Al igual que en los casos anteriores, se puede desarrollar una fórmula compuesta, similar a la fórmula compuesta del rectángulo pero tomando los puntos medios de los subintervalos.

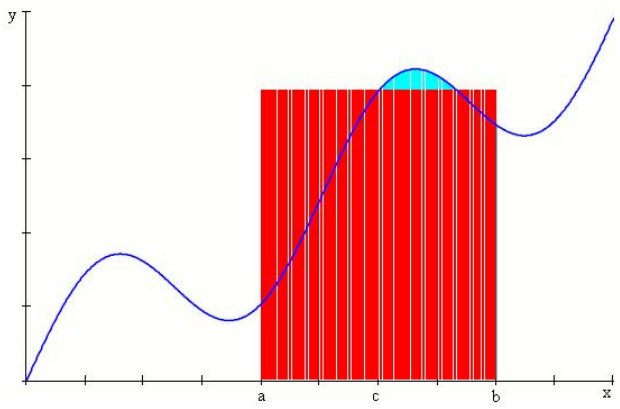


Figura 6.13: Fórmula del punto medio.

Sin embargo, la idea principal de las fórmulas abiertas no está relacionada con tomar puntos de un intervalo según un paso uniforme sino en determinar los puntos para efectuar la integración eligiéndolos de una manera «inteligente». ¿Qué significa inteligente? Analicemos brevemente la fórmula del punto medio. Al elegir dicho punto y no los extremos del intervalo, suponemos que el rectángulo que queda formado aproxima mejor la integral buscada. Si dividimos este intervalo en varios subintervalos más pequeños, tendremos la fórmula compuesta. Así y todo, estamos algo limitados.

Podríamos avanzar en la idea y desarrollar una fórmula similar para el método de Simpson, es decir, crear una curva que no pase por los extremos y nos permita obtener una buena aproximación. Pero de todas maneras tenemos la misma limitante: debemos trabajar con puntos equidistantes [3]. Esto puede llevar a que debamos utilizar las fórmulas compuestas con muchos términos para alcanzar aproximaciones razonables. Veamos en el punto siguiente un método de integración que explota la idea de las fórmulas abiertas de Newton-Cotes eligiendo puntos en la curva de manera de optimizar la aproximación de la integral buscada.

6.2.4. Cuadratura de Gauss

Recordemos la fórmula para una cuadratura:

$$Q_n(f) = \sum_{i=1}^{n} c_i f(x_i).$$

Supongamos ahora que elegimos una curva que pase por ciertos puntos y que aproxime la integral de la función dada, usando la fórmula de cuadratura. Una curva de ese tipo se ve en la figura 6.14.

Elegiremos estos puntos que optimizan la integral buscada q aquellos en los cuales la función se intersecta con la curva de aproximación. Entonces, nuestro problema es elegir la curva más conveniente. Por ejemplo, en la figura 6.14 se eligió una parábola, por lo tanto, se tienen dos puntos que en los cuales la parábola se intersecta con la función. Podríamos haber utilizado una recta, una parábola cúbica, etc.

En los métodos anteriores, para obtener la integral buscada, hemos utilizado puntos conocidos o que podíamos conocer a partir de definir el paso h. Por ejemplo, en el método del trapecio utilizamos dos puntos para aproximar nuestra integral, x_1 y x_2, de manera que nuestra

[3] Recordemos que la base de la integración numérica es la interpolación polinómica, que se vuelve inestable cuando los puntos usados están separados uno de otro en forma equidistante.

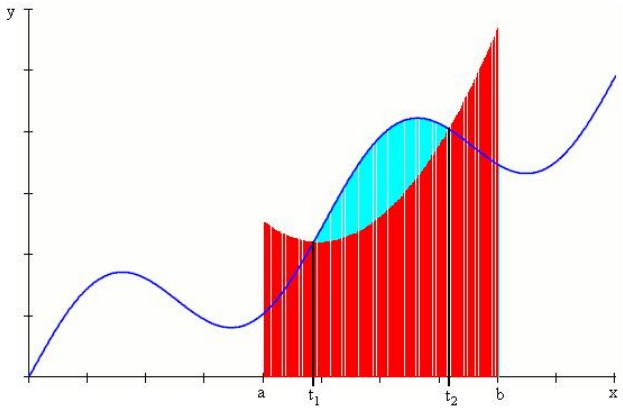


Figura 6.14: Cuadratura usando una curva de aproximación.

aproximación queda de la siguiente manera:

$$I(f) = \frac{h}{2}\left[f(x_1) + f(x_2)\right].$$

De la fórmula de cuadratura podemos extraer que, si definimos que $h = b - a$, donde $[a; b]$ es nuestro intervalo de integración, para el caso del método del trapecio tendremos que:

$$x_1 = a,\ x_2 = b,\ c_1 = c_2 = \frac{b-a}{2}.$$

Supongamos ahora que definimos un intervalo fijo de integración, por ejemplo, el $[-1, 1]$. En vez de fijar los valores x_i, armemos una aproximación de nuestra integral de manera tal que dipsongamos de más «variables» para aproximar nuestra integral. En esta nueva situación debemos obtener para ese intervalo los puntos x_i y los coeficientes c_i para nuestra fórmula de cuadratura, esto es, debemos buscar los puntos $x_1; x_2; \ldots; x_n$ y los coefcientes $c_1; c_2; \ldots; c_n$ que optimicen nuestra aproximación. En consecuencia, tenemos $2n$ incógnitas que debemos obtener. Si recordamos que un polinomio de grado $2n-1$ tiene $2n$ coeficientes (por ejemplo, un polinomio de tercer grado tiene la forma $a_0+a_1x+a_2x^2+a_3x^3$), podríamos decir que hallar esos parámetros para nuestra fórmula de cuadratura es equivalente a obtener los coeficientes de ese polinomio de grado $2n - 1$.

Gauss definió estos polinomios que permiten aproximar la integral en el intervalo $[-1; 1]$ dependiendo de la cantidad de puntos que se deseen utilizar. Estos polinomios son ortogonales y conocidos como *polinomios de Legendre*, y son los siguientes:

$$P_0(x) = 1 \qquad P_1(x) = x$$
$$P_2(x) = \frac{1}{2}(3x^2 - 1) \qquad P_3(x) = \frac{1}{2}(5x^3 - 3x)$$
$$P_k(x) = \frac{1}{2^k\, k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}(x^2 - 1)^k.$$

Calculando la raíz de cada polinomio se obtienen los puntos x_i. Con éstos y la ayuda de un polinomio interpolante de Lagrange integrado en el intervalo $[-1; 1]$, los coeficientes c_i. (En [1] se pueden ver más detalles de cómo obtener los coeficientes de peso.)

En la tabla 6.3 se dan algunos los valores de las raíces y los coeficientes, de acuerdo con la cantidad de puntos que se utilicen para aproximar la integral.

Este método es muy útil cuando lo que queremos aproximar son integrales de funciones polinómicas, puesto que los resultados son exactos cuando $g \leq 2n - 1$, donde g es el grado del

Tabla 6.3: Raíces y coeficientes de la cuadratura de Gauss-Legendre

n	$\mathbf{x_i}$	$\mathbf{c_i}$
1	$x_1 = 0.0000000000$	$c_1 = 2.0000000000$
2	$x_1 = -\frac{1}{\sqrt{3}} = -0.5773502692$	$c_1 = 1.0000000000$
	$x_2 = \frac{1}{\sqrt{3}} = 0.5773502692$	$c_2 = 1.0000000000$
3	$x_1 = -0.7745966692$	$c_1 = 0.5555555556$
	$x_2 = 0.0000000000$	$c_2 = 0.8888888889$
	$x_3 = 0.7745966692$	$c_3 = 0.5555555556$
4	$x_1 = -0.8611363116$	$c_1 = 0.3478548451$
	$x_2 = -0.3399810436$	$c_2 = 0.6521451549$
	$x_3 = 0.3399810436$	$c_3 = 0.6521451549$
	$x_4 = 0.8611363116$	$c_4 = 0.3478548451$
5	$x_1 = -0.9061798459$	$c_1 = 0.2369268850$
	$x_2 = -0.5384693101$	$c_2 = 0.4786286705$
	$x_3 = 0.0000000000$	$c_3 = 0.5688888889$
	$x_4 = 0.5384693101$	$c_4 = 0.4786286705$
	$x_5 = 0.9061798459$	$c_5 = 0.2369268850$

polinomio a integrar y n la cantidad de puntos de Gauss. Por ejemplo, con $n = 2$, es decir, con dos puntos de Gauss, podemos aproximar cualquier integral de polionomios cuyo grado sea menor o igual a tres, pues se cumple que $g = 3 \leq 2 \cdot 2 - 1$.

Si el intervalo de integración no es $[-1; 1]$, basta con hacer un cambio de coordenadas. Si se tiene la siguiente integral:

$$I(f) = \int_a^b f(x)\,\mathrm{d}x,$$

debemos hacer la siguiente transformación lineal para poder aproximar con cuadratura de Gauss:

$$x = \frac{b-a}{2}t + \frac{b+a}{2};\ I(f) = \frac{b-a}{2}\int_{-1}^{1} f(t)\,\mathrm{d}t.$$

Finalmente, una cuestión a tener en cuenta es qué error se comete por aproximar una integral mediante cuadratura de Gauss. La expresión del error el intervalo $[-1; 1]$ está dado por

$$E = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^2} f^{2n}(\xi),$$

donde n es el número de puntos utilizados y $\xi \in [-1, 1]$. Si ampliamos el método al intervalo $[a; b]$, tenemos que el error está dado por

$$E = \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[(2n)!]^2} f^{2n}(\xi),$$

con $\xi \in [a, b]$. Vemos que en ambos casos el error cometido es proporcional a la derivada de orden $2n$. Por ejemplo, si $n = 2$, entonces el error cometido es proporcional a la derivada cuarta ($f^{iv}(\xi)$), pues tenemos

$$E = \frac{(b-a)^{2\cdot 2+1}(2!)^4}{(2\cdot 2+1)[(2\cdot 2)!]^2} f^{2\cdot 2}(\xi) = \frac{(b-a)^5(2!)^4}{5(4!)^2} f^{iv}(\xi).$$

Esto confirma que con dos puntos de Gauss ($n = 2$) obtenemos una integral «exacta» para polinomios de grado 3 o menor, pues en esos casos se cumple que $f^{iv}(x) = 0$ para cualquier x, por lo tanto, también para $f^{iv}(\xi)$ con $\xi \in [a, b]$.

Al igual que para los métodos anteriores, podemos pensar en un método compuesto para Gauss. Efectivamente, si dividimos el intervalo $[a;b]$ en subintervalos más pequeños, podemos utilizar la cuadratura de Gauss en esos subintervalos, con la correspondiente transformación lineal, e inclusive usar un aproximación con n no mayor a 3, con excelentes resultados.

6.2.5. Integrales múltiples

Al igual que para el caso de integrales simples, podemos calcular en forma numérica integrales múltiples, en dos o tres dimensiones. Tomemos la siguiente integral:

$$\iint_A f(x;y)\mathrm{d}A,$$

donde A es una regióm rectangular en el plano tal que

$$A = \{(x;y)|a \leq x \leq b;\ c \leq y \leq d\}.$$

Entonces, podemos escribir la integral de arriba como

$$\int_c^d \left[\int_a^b f(x;y)\mathrm{d}x\right]\mathrm{d}y.$$

Integremos respecto a x usando el método del trapecio. De esta manera obtendremos

$$\int_a^b f(x;y)\mathrm{d}x \approx \frac{b-a}{2}\left[f(a;y)+f(b;y)\right].$$

Reemplacemos esta expresión en la integral doble y hagamos lo mismo pero respecto a y. Entonces nos queda que

$$\begin{aligned}\int_c^d \left[\int_a^b f(x;y)\mathrm{d}x\right]\mathrm{d}y &\approx \int_c^d \frac{b-a}{2}\left[f(a;y)+f(b;y)\right]\mathrm{d}y\\ &\approx \frac{b-a}{2}\int_c^d \left[f(a;y)+f(b;y)\right]\mathrm{d}y\\ &\approx \frac{b-a}{2}\left[\int_c^d f(a;y)\mathrm{d}y+\int_c^d f(b;y)\mathrm{d}y\right]\end{aligned}$$

Si aplicamos a cada integral la regla del trapecio, nos queda

$$\begin{aligned}\int_c^d f(a;y)\mathrm{d}y &\approx \frac{d-c}{2}\left[f(a;c)+f(a;d)\right]\\ \int_c^d f(b;y)\mathrm{d}y &\approx \frac{d-c}{2}\left[f(b;c)+f(b;d)\right].\end{aligned}$$

Al reemplazar estas dos expresiones en la general nos queda que

$$\int_c^d\int_a^b f(x;y)\ \mathrm{d}x\,\mathrm{d}y \approx \frac{(b-a)(d-c)}{4}\left[f(a;c)+f(a;d)+f(b;c)+f(b;d)\right].$$

En definitiva, podemos obtener una aproximación de una integral múltiple, en este caso doble, mediante la aplicación del método del trapecio en dos dimensiones. También aplicando el método de Simpson podemos obtener una aproximación de dicha integral. En este caso, la expresión es

$$\begin{aligned}\int_c^d\int_a^b f(x;y)\ \mathrm{d}x\,\mathrm{d}y \approx\ & \frac{h_x\ h_y}{9}\Bigg\{f(a;c)+f(a;d)+f(b;c)+f(b;d)+\\ &+4\left[f\left(a;\frac{c+d}{2}\right)+f\left(b;\frac{c+d}{2}\right)+f\left(\frac{a+b}{2};c\right)+f\left(\frac{a+b}{2};d\right)\right]+\\ &+16\left[f\left(\frac{a+b}{2};\frac{c+d}{2}\right)\right]\Bigg\},\end{aligned}$$

donde $h_x = \frac{b-a}{2}$ y $h_y = \frac{d-c}{2}$. Si reemplazamos esto último en la expresión general y además definimos $x_0 = a$, $x_1 = \frac{a+b}{2}$,$x_2 = b$, $y_0 = c$, $y_1 = \frac{c+d}{2}$ e $y_2 = d$, tenemos que

$$\begin{aligned}\int_c^d \int_a^b f(x;y)\ \mathrm{d}x\,\mathrm{d}y &\approx \frac{(b-a)(d-c)}{36}\{f(x_0;y_0)+f(x_0;y_2)+f(x_2;y_0)+f(x_2;y_2)+\\ &\quad +4\left[f(x_0;y_1)+f(x_1;y_0)+f(x_1;y_2)+f(x_2;y_1)+4f(x_1;y_1)\right]\}.\end{aligned}$$

El error cometido por aproximar la integral mediante esta fórmula está dado por:

$$\begin{aligned}E_T &= \frac{(b-a)(d-c)}{12}\left[h_x^2\frac{\partial^2 f(\hat{\xi};\hat{\mu})}{\partial x^2}+h_y^2\frac{\partial^2 f(\bar{\xi};\bar{\mu})}{\partial y^2}\right] \quad \text{(Método del trapecio)},\\ E_S &= \frac{(b-a)(d-c)}{90}\left[h_x^4\frac{\partial^4 f(\hat{\xi};\hat{\mu})}{\partial x^4}+h_y^4\frac{\partial^4 f(\bar{\xi};\bar{\mu})}{\partial y^4}\right] \quad \text{(Método de Simpson)},\end{aligned}$$

que, como podemos observar, son muy parecidos a los vistos para el caso de integrales simples.

Estos métodos también se pueden modificar para obtener las fórmulas compuestas, similares a las vistas anteriormente. (Para más detalles, véase [1].)

Así como hemos aplicado los métodos de trapecio y de Simpson, lo mismo podemos hacer con la cuadratura de Gauss. Si aplicamos el mismo razonamiento para integrar según x tendremos que

$$\int_a^b f(x;y)\ \mathrm{d}x \approx \frac{b-a}{2}\sum_{i=1}^{n} c_i f(x_i;y).$$

Si hacemos lo mismo respecto de y, obtendremos

$$\begin{aligned}\int_c^d \int_a^b f(x;y)\ \mathrm{d}x\,\mathrm{d}y &\approx \int_c^d \frac{b-a}{2}\sum_{i=1}^{n} c_i f(x_i;y)\,\mathrm{d}y\\ &\approx \frac{b-a}{2}\sum_{i=1}^{n}\int_c^d c_i f(x_i;y)\,\mathrm{d}y\\ &\approx \frac{b-a}{2}\sum_{i=1}^{n}\frac{d-c}{2}\sum_{j=1}^{m} c_i c_j f(x_i;y_j)\\ &\approx \frac{b-a}{2}\frac{d-c}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} c_i c_j f(x_i;y_j)\\ &\approx \frac{(b-a)(d-c)}{4}\sum_{i=1}^{n}\sum_{j=1}^{m} c_i c_j f(x_i;y_j),\end{aligned}$$

con

$$\begin{aligned}x_i &= \frac{b-a}{2}t_i+\frac{b+a}{2}\\ y_j &= \frac{d-c}{2}t_j+\frac{d+c}{2},\end{aligned}$$

donde t_i y t_j son las raíces de los polinomios de Legendre, y c_i y c_j, los coeficientes de peso. Por ejemplo, si tomamos $n = m = 2$ tenemos que $t_1 = -\frac{1}{\sqrt{3}}$, $t_2 = \frac{1}{\sqrt{3}}$ y $c_1 = c_2 = 1$, y la aproximación nos queda como

$$\int_c^d \int_a^b f(x;y)\ \mathrm{d}x\,\mathrm{d}y \approx \frac{(b-a)(d-c)}{4}\left[f(x_1;y_1)+f(x_1;y_2)+f(x_2;y_1)+f(x_2;y_2)\right].$$

con

$$x_1 = -\frac{b-a}{2}\frac{1}{\sqrt{3}}+\frac{b+a}{2} \qquad x_2 = \frac{b-a}{2}\frac{1}{\sqrt{3}}+\frac{b+a}{2},$$

y

$$y_1 = -\frac{d-c}{2}\frac{1}{\sqrt{3}} + \frac{d+c}{2} \qquad y_2 = \frac{d-c}{2}\frac{1}{\sqrt{3}} + \frac{d+c}{2}.$$

Podemos ver que con este método solamente tenemos que evaluar la función a integrar en cuatro puntos, en cambio, con el método de Simpson debemos evaluar la misma función en nueve puntos. Este método es muy utilizado por el *Método de los Elementos Finitos* para obtener integrales dobles.

6.3. Notas finales

La integración numérica es uno de los métodos numéricos más utilizados en la ingeniería y en la ciencia en general. Inclusive, muchos programas para computadoras hacen usos de los algoritmos vistos en este capítulo. Por ejemplo, el MatLab® aplica el método de Simpson en su función `quad` que calcula integrales definidas, en tanto que el Mathcad®, aplica por omisión el método de Romberg.

Por otro lado, uno de los métodos numéricos más utilizados en el análisis estructural, el *Método de los Elementos Finitos*, aplica la integración numérica en forma sistemática para obtener la matriz de rigidez de un sistema estático. Más aún, para ciertos casos especiales hace uso exclusivo de la cuadratura de Gauss, como es el caso de la integración en una y dos dimensiones para elementos lineales o de superficie (elementos de barra, de viga, de estado plano y de placa) e inclusive para determinados tipos de elementos se ayuda con una «integración reducida» para evitar ciertos problemas del modelo numérico.

Capítulo 7

Ecuaciones diferenciales ordinarias

7.1. Ecuaciones diferenciales ordinarias con valores iniciales

7.1.1. Introducción

Muchos de los problemas que debemos resolver como ingenieros se pueden representar mediante ecuaciones diferenciales ordinarias, que son aquellas que están expresadas en derivadas totales [1]. Como ejemplos de este tipo de ecuaciones tenemos las siguientes:

- El equilibrio de una viga sometida a flexión ($\frac{\mathrm{d}M}{\mathrm{d}x} + p = 0$);
- Un circuito del tipo LR ($L\,\frac{\mathrm{d}i}{\mathrm{d}t} + R\,i = V$);
- La transmisión del calor unidimensional ($q = -kA\frac{\mathrm{d}T}{\mathrm{d}x}$) [2].

Así, buena parte de los métodos que empleamos para «atacar» un determinado problema resultan ser soluciones analíticas de ecuaciones diferenciales que se aplican en forma metódica y que se han obtenido a partir de ciertas condiciones, que pueden ser iniciales o de borde. Un caso bien conocido es la resolución de sistemas hiperestáticos en Estática (también los isoestáticos), en los que se aplican métodos prácticos y numéricos (como el método de Cross) derivados de las soluciones análiticas.

Del conjunto de ecuaciones diferenciales empezaremos por las más sencilla, que son aquellas que involucran a la primera derivada, de las que basta conocer las condiciones iniciales. Si bien en cualquier curso de Análisis matemático se aprenden métodos analíticos para obtener las soluciones de dichas ecuaciones, sabemos que no siempre son aplicables o no siempre obtendremos soluciones analíticas. Por ejemplo, y volviendo al caso de estructuras hiperestáticas, no resulta sencillo resolver la ecuación diferencial para el caso de una carga concentrada. Es en estos casos cuando los métodos numéricos se convierten en la única herramienta para obtener algún tipo de solución aproximada que nos permita resolver el problema.

Existen muchos ejemplos de ecuaciones diferenciales con condiciones iniciales, entre los cuales podemos mencionar los siguientes:

- **Dinámica de poblaciones**. El economista inglés Thomas Malthus propuso el siguiente modelo matemático para definir el crecimiento demográfico:

$$\frac{\mathrm{d}P}{\mathrm{d}t} = kP;$$

[1] Para una mejor compresión del tema, ver [15].

[2] En realidad, se trata de un sistema de ecuaciones diferenciales, pues $q = \frac{\mathrm{d}Q}{\mathrm{d}t}$)

con $k > 0$, es decir, que la tasa de crecimiento de la población es proporcional a la población total. (Este modelo en realidad no es muy preciso, pues deja de lado otros factores como la inmigración, por ejemplo, pero en su momento daba una buena aproximación al problema demográfico.)

- **Desintegración radiactiva**. El siguiente modelo matemático es el que se aplica para el estudio de la desintegración radiactiva:

$$\frac{\mathrm{d}A}{\mathrm{d}t} = kA;$$

en este caso, con $k < 0$. Este modelo es la base del método de datación por Carbono 14, usado en muchas disciplinas científicas.

- **Ley de Newton del enfriamiento o calentamiento**. Isaac Newton propuso la siguiente ley matemática para el cambio de temperatura:

$$\frac{\mathrm{d}T}{\mathrm{d}t} = k(T - T_m);$$

con $k < 0$, donde T_m es la temperatura del medio, y T la del objeto analizado.

- **Ley de Torricceli**. El drenado de un tanque cumple con el siguiente modelo:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = -A_h\sqrt{2gh}.$$

Si definimos $V = A_w\, h$, entonces la expresión anterior se puede escribir como

$$\frac{\mathrm{d}h}{\mathrm{d}t} = -\frac{A_h}{A_w}\sqrt{2gh}.$$

La mayoría de los libros toma el caso del péndulo como el ejemplo tradicional de las ecuaciones diferenciales ordinarias con valores iniciales. El modelo matemático que representa este fenómeno está dado por:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -\frac{g}{L}\,\mathrm{sen}(\theta),$$

donde g es la aceleración de la gravedad, L, la longitud del péndulo, y θ, el ángulo del péndulo respecto de la vertical. Este ejemplo suele linealizarse para el caso de ángulos muy pequeños, pues se cumple que $\mathrm{sen}(\theta) = \tan(\theta) = \theta$, y la ecuación diferencial queda

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -\frac{g}{L}\theta,$$

modelo que en realidad está representado con una ecuación diferencial de segundo orden.

Otro ejemplo de la ingeniería civil en el ámbito del análisis estructural es la ecuación del esfuerzo normal en una barra, que se define como

$$\frac{\mathrm{d}N}{\mathrm{d}x} = -t(x);$$

donde $t(x)$ es una carga uniformemente distribuida en el eje de la barra.

En lo que sigue veremos, primero, las condiciones para que la solución de una ecuación diferencial ordinaria tenga solución única, y en segundo término, varios métodos para resolver numéricamente este tipo de ecuaciones.

7.1.2. Condición de Lipschitz

Una ecuación diferencial ordinaria con valor inicial está definida de la siguiente manera:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y) \;\text{ con } a \leq t \leq b \;\text{ e } y(a) = y_0.$$

Una función $f(t,y) \in D \subset \Re^2$, con D convexo, cumple con la condición de Lipschitz si satisface que

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

o

$$\left|\frac{\partial f(t,y)}{\partial y}\right| \leq L,$$

para todo $(t,y) \in D$.

Para que una ecuación diferencial tenga solución única se debe satisfacer el siguiente teorema.

Teorema 7.1. Sea $f(t,y)$ continua en D, tal que $D = \{(t,y)|a \leq t \leq b; -\infty \leq y \leq +\infty\}$. Si $f(t,y)$ satisface la condición de Lipschitz en D en la variable y, entonces el problema de valor inicial

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y) \;\text{ con } a \leq t \leq b \;\text{ e } y(a) = y_0$$

tiene solución única $y(t)$ para $a \leq t \leq b$.

Por lo tanto, toda ecuación diferencial con valor inicial que cumpla con la condición de Lipschitz tiene solución única.

7.1.3. Problema bien planteado

Un problema de valor inicial del tipo

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y) \;\text{ con } a \leq t \leq b \;\text{ e } y(a) = y_0$$

se dice bien planteado si:

- El problema tiene solución única (cumple con la condición de Lipschitz);
- Para cualquier $\epsilon > 0$, existe una constante positiva $k(\epsilon)$ con la propiedad de que siempre que $|\epsilon_0| < \epsilon$, y un $\delta(t)$ que sea continuo, con $\delta(t) < \epsilon$ en $[a;b]$, el problema tiene solución única $z(t)$; es decir,

  $$\frac{\mathrm{d}z}{\mathrm{d}t} = f(t,z) + \delta(t) \;\text{ con } a \leq t \leq b \;\text{ e } z(a) = y_0 + \epsilon_0,$$

  con

  $$|z(t) - y(t)| < k(\epsilon)\,\epsilon,$$

  para toda $a \leq t \leq b$.

En definitiva, un problema está bien planteado si una perturbación (un $\delta(t)$) del problema original no cambia la esencia del mismo. El siguiente teorema define la condición de *problema bien planteado.*

Teorema 7.2. Sea $D = \{(t,y)|a \leq t \leq b; -\infty \leq y \leq +\infty\}$. Si $f(t,y)$ es continua y satisface la condición de Lipschitz en la variable y en el conjunto D, entonces el problema de valor inicial

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y) \;\text{ con } a \leq t \leq b \;\text{ e } y(a) = y_0$$

se dice *bien planteado.*

7.1.4. Métodos de Euler

Un vez definidas las condiciones que debe cumplir el problema de valor inicial para tener solución única, nos ocuparemos de los métodos para resolverlo.

Para empezar, tomemos la formulación del problema

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y).$$

Desarrollamos por Taylor la función $y(t)$, desconocida, en un entorno $[t; t+h]$ para obtener $y(t+h)$:

$$y(t+h) = y(t) + y'(t)\, h + y''(t)\frac{h^2}{2} + \dots \ .$$

Como $y'(t) = f(t, y)$, entonces escribamos la expresión como sigue:

$$y(t+h) = y(t) + f(t, y)\, h + y''(t)\frac{h^2}{2} + \dots \ .$$

Dado que nuestro entorno de la solución está dado por $[a; b]$, definamos el paso h como $h = \dfrac{b-a}{N}$, donde N es el número de intervalos. Ahora definamos que $t_{i+1} = t_i + h$. Así, nuestra expresión anterior nos queda:

$$y(t_{i+1}) = y(t_i) + h\, f[t_i; y(t_i)] + y''(t_i)\frac{h^2}{2} + \dots \ .$$

Si truncamos en la segunda derivada, nos queda

$$y(t_{i+1}) = y(t_i) + h\, f[t_i; y(t_i)] + y''(\xi_i)\frac{h^2}{2},$$

con $\xi_i \in [t_i; t_{i+1}]$.

Puesto que lo que buscamos es una aproximación de $y(t_i)$, defininamosla como w_i. Entonces nuestra expresión nos queda de la siguiente forma:

$$w_{i+1} = w_i + h\, f(t_i; w_i),$$

para $i = 0; 1; \dots; N-1$. Este método se conoce como *método de Euler explícito.*

Supongamos ahora que desarrollamos $y(t)$ en t_i+h para obtener $y(t_i)$. Entonces tendremos que

$$y(t_i) = y(t_i + h) - y'(t_i + h)\, h + y''(t_i + h)\frac{h^2}{2} + \dots \ ;$$

y, como $y'(t_i + h) = f[t_i + h; y(t_i + h)]$, nos queda que

$$y(t_i) = y(t_i + h) - f[t_i + h; y(t_i + h)]\, h + y''(t_i + h)\frac{h^2}{2} + \dots \ .$$

Nuevamente, como $t_{i+1} = t_i + h$, y despejando $y(t_{i+1})$ limitando otra vez la expresión a la segunda derivada, tenemos que

$$y(t_{i+1}) = y(t_i) + h\, f[t_{i+1}; y(t_{i+1})] - y''(\xi_i)\frac{h^2}{2},$$

con $\xi_i \in [t_i; t_{i+1}]$.

En forma análoga, lo que en realidad buscamos es una aproximación de $y(t_{i+1})$, por lo tanto tendremos la siguiente expresión:

$$w_{i+1} = w_i + h\, f(t_{i+1}; w_{i+1}),$$

para $i = 0; 1; \dots; N-1$. Este método se conoce como *método de Euler implícito.*

Para ambos métodos no hay prácticamente diferencias con relación a la estabilidad numérica. En efecto, si los perturbamos, obtenemos las siguientes expresiones para los errores locales:

- Método explícito: $\varepsilon_{i+1} = \varepsilon_i \left(1 + h\frac{\partial f}{\partial w}\right)$;
- Método implícito: $\varepsilon_{i+1} = \varepsilon_i \dfrac{1}{1 - h\frac{\partial f}{\partial w}}$;

Vemos que cuando $\dfrac{\partial f}{\partial w} < 0$ ambos métodos son estables, pues $\varepsilon_{i+1} < \varepsilon_i$, en tanto que cuando $\dfrac{\partial f}{\partial w} > 0$, entonces $\varepsilon_{i+1} > \varepsilon_i$, y el error es creciente para los dos métodos que estamos analizando.

Entonces, ¿cuál es la diferencia al decidir aplicar uno u otro método? La principal razón está en que el método explícito de Euler utiliza la pendiente en t_i para obtener el nuevo punto w_{i+1}, en tanto que el método implícito utiliza la pendiente en el punto t_{i+1} para obtener el nuevo punto w_{i+1}. Por este motivo es que el método de Euler implícito suele dar mejores resultados que el explícito. (Sin embargo, no siempre es así, pues en algunos casos el método implícito se vuelve muy inestable.)

La desventaja del método implícito está en que para que sea «fácil» implementarlo, requiere trabajar algebraicamente la expresión para transformarla en una explícita, es decir, que no aparezca w_{i+1} a ambos lados de la igualdad. Como esto no siempre es posible, la implementación del método implícito no siempre resulta ser sencilla. Cuando esto ocurre, suele complementarse con algún método para obtener raíces de ecuaciones, generalmente el método de las aproximaciones sucesivas, pues ya se tiene la función $g(t)$ que haga convergente la sucesión.

Pero existe otra forma de resolver esta situación. Supongamos que planteamos el siguiente sistema:

$$\begin{aligned} w^*_{i+1} &= w_i + h\, f(t_i; w_i) \\ w_{i+1} &= w_i + h\, f(t_i; w^*_{i+1}), \end{aligned}$$

es decir, obtenemos una primera aproximación de w_{i+1} con el método explícito, que llamaremos w^*_{i+1}, que luego usaremos para obtener una nueva aproximación de w_{i+1}, con el método implícito, que «corrige» la anterior. Este método se conoce como *método predictor-corrector de Euler* [3].

Si bien los métodos de Euler son bastante sencillos de implementar, los resultados que se obtienen no son buenas aproximaciones de nuestro problema. Se los usa solamente como introducción a los métodos numéricos y para el análisis del error.

En efecto, para analizar el error, consideremos estos dos lemas:

1. Para toda $x \geq -1$ y para cualquier m positiva, tenemos que $0 \leq (1+x)^m \leq e^{mx}$.
2. Si s y t son números reales positivos, $\{a_i\}_{i=0}^k$ es una sucesión que satisface $a_0 \geq -t/s$, y se cumple que
$$a_{i+1} \leq (1+s)a_i + t, \quad \text{para cada } i = 0; 1; 2; \ldots; k,$$
entonces
$$a_{i+1} \leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}.$$

A partir de estos dos lemas se tiene el siguiente teorema.

Teorema 7.3. Sea $f(t, y)$ continua, que satisface la condición de Lipschitz con la constante L en
$$D = \{(t, y) | a \leq t \leq b; -\infty \leq y \leq +\infty\},$$

[3] Este método no suele estar incluir en los libros de texto, posiblemente porque no mejora la aproximación de una manera significativa. Una excepción es [8].

y existe una constante M, tal que $|y''(t)| \leq M$ para toda $t \in [a;b]$. Si $y(t)$ es la solución única del problema de valor inicial dado por

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y) \text{ con } a \leq t \leq b \text{ e } y(a) = y_0,$$

y los $w_0, w_1, \ldots, w_N$ son las aproximaciones a nuestra función, obtenidas por el método de Euler, entonces se cumple que

$$|y(t_i) - w_i| \leq \frac{M}{2L}\left[e^{L(t_i-a)} - 1\right].$$

La demostración de este teorema se puede ver [1].

Orden de convergencia

El error que acabamos de analizar es el error global, pues hemos estimado una cota del error entre el valor real (o exacto) y la aproximación por un método numérico. Sin embargo, los métodos numéricos suelen definirse según el *error local*, es decir, el error entre dos iteraciones sucesivas. Este error, en el método de Euler, está dado por:

$$e_L = \frac{y(t_{i+1} - y(t_i)}{h} - f[t_i; y(t_i)].$$

Como vimos, el método explícito de Euler se puede obtener a partir de una desarrollo de Taylor, del cual resulta que

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \ldots = y(t_i) + hf[t_i; y(t_i)] + f'[t_i; y(t_i)]\frac{h^2}{2} + \ldots;$$

por lo tanto

$$\begin{aligned} \frac{y(t_{i+1}) - y(t_i)}{h} - f[t_i; y(t_i)] &= \frac{h}{2}f'[\xi; y(\xi)] \\ e_L &= \frac{h}{2}f'[\xi; y(\xi)], \end{aligned}$$

con $\xi \in [t_i; t_{i+1}]$, lo que muestra que el error local del método de Euler es $O(h)$, es decir, tiene un orden de convergencia lineal. Con un análisis similar podemos demostrar que el método implícito es del mismo orden de convergencia. Y dado que ambos métodos son de convergencia lineal, lo mismo podemos decir del predictor-corrector.

7.1.5. Métodos de Taylor de orden superior

Vimos que el método de Euler es muy fácil de aplicar pero poco preciso. Una forma de mejorarlo es partiendo otra vez del desarrollo por Taylor pero ampliando la cantidad de términos de la serie:

$$y(t_{i+1}) = y(t_i) + h\,y'(t_i) + \frac{h^2}{2!}y''(t_i) + \frac{h^3}{3!}y'''(t_i) + \ldots + \frac{h^n}{n!}y^{(n)}(t_i).$$

Como además tenemos que

$$\frac{\mathrm{d}\,y(t)}{\mathrm{d}t} = y'(t) = f(t;y), y(t_i) = y_i \text{ y } y(t_{i+1}) = y_{i+1},$$

el desarrollo por Taylor lo podemos escribir de la siguiente manera:

$$y_{i+1} = y_i + h\,f(t_i; y_i) + \frac{h^2}{2!}f'(t_i; y_i) + \frac{h^3}{3!}f''(t_i; y_i) + \ldots + \frac{h^n}{n!}f^{(n-1)}(t_i; y_i).$$

Es decir, podemos armar un esquema para obtener los y_{i+1} a partir de un polinomio de Taylor, calculando las derivadas totales de la función $f(t;y)$. El error que se introduce en este esquema es el primer término que dejamos de considerar, que en nuestro caso es

$$E = \frac{h^{n+1}}{(n+1)!} f^{(n)}[\xi; y(\xi)] \quad \text{con} \quad \xi \in [t_i; t_{i+1}],$$

y como el error local está dado por

$$e_L = \frac{y(t_{i+1}) - y(t_i)}{h} - f[t_i; y(t_i)];$$

para este caso queda definido como

$$e_L = \frac{h^n}{(n+1)!} f^{(n)}[\xi_i; y(\xi_i)],$$

con $\xi \in [t_i; t_{i+1}]$. Estos métodos se conocen como *métodos de Taylor de orden superior*, pues podemos definir el orden de convergencia igual a n, siempre que al menos $f(t;y) \in \mathrm{C}^{n-1}[a;b]$. Podemos ver que el método de Euler es un caso particular del método de Taylor para $n = 1$. (Podríamos armar métodos de Taylor de orden superior implícitos, aunque de escasa utilidad, dado que deberíamos transformar algebraicamente el algoritmo para obtener una formulación explícita.)

7.1.6. Métodos de Runge-Kutta

Los métodos de Taylor resultan muy instructivos para entender cómo mejorar nuestras aproximaciones, pero muy poco prácticos al momento de implementar un algoritmo de cálculo. El principal escollo para esto es la necesidad de calcular las derivadas de $y(t)$ (o de $f(t,y)$), algo que no siempre es fácil de hacer. Eso obligaría en muchos casos a programar algoritmos particulares según el problema que enfrentemos, lo que le quita generalidad.

Un segundo problema está relacionado directamente con la facilidad para obtener las derivadas de la función $f(t;y)$. Aún cuando se pueda probar que $f(t,y) \in \mathrm{C}^{n-1}[a;b]$, puede ser muy complicado obtener las derivadas de mayor orden, perdiéndose la capacidad de obtener rápidamente una aproximación de la solución buscada.

Es por eso que existen otros métodos para aproximar la solución de una ecuación diferencial que consiguen órdenes de convergencia similares a los de Taylor pero que no requieren la obtención de las derivadas de la función $f(t;y)$. Son los denominados *métodos de Runge-Kutta*.

Para poder construir los métodos de Runge-Kutta, nos basaremos en el siguiente teorema.

Teorema 7.4. Sea $f(t;y) \in \mathrm{C}^{n+1}\ D$ con $D = \{(t;y) | a \le t \le b, c \le y \le d\}$, y sea $(t_0;y_0) \in D$. Entonces, para toda $(t;y) \in D$, existe $\xi \in [t_0;t]$ y $\mu \in [y_0;y]$ con

$$f(t;y) = P_n(t;y) + R_n(t;y),$$

tal que

$$\begin{aligned} P_n(t;y) \;=\; & f(t_0;y_0) + \left[(t-t_0) \frac{\partial\, f(t_0;y_0)}{\partial t} + (y-y_0) \frac{\partial\, f(t_0;y_0)}{\partial y} \right] + \\ & + \left[\frac{(t-t_0)^2}{2!} \frac{\partial^2\, f(t_0;y_0)}{\partial t^2} + (t-t_0)(y-y_0) \frac{\partial^2\, f(t_0;y_0)}{\partial t\, \partial y} + \right. \\ & \left. + \frac{(y-y_0)^2}{2!} \frac{\partial^2\, f(t_0;y_0)}{\partial y^2} \right] + \ldots + \\ & + \left[\frac{1}{n!} \sum_{j=0}^{n} \binom{n}{j} (t-t_0)^{n-j} (y-y_0)^j \frac{\partial^n\, f(t_0;y_0)}{\partial t^{n-j}\, \partial y^j} \right], \end{aligned}$$

y

$$R_n(t,y) = \frac{1}{(n+1)!}\sum_{j=0}^{n+1}\binom{n+1}{j}(t-t_0)^{n+1-j}(y-y_0)^j\frac{\partial^{n+1} f(\xi;\mu)}{\partial t^{n+1-j}\,\partial y^j}.$$

A la función $P_n(t;y)$ se la denomina *polinomio de Taylor de grado* n *en dos variables* para la función $f(t;y)$ alrededor de $(t_0;y_0)$, en tanto que $R_n(t;y)$ es el residuo o error asociado a $P_n(t;y)$.

Esto es necesario pues los métodos de Runge-Kutta se basan en aproximar el polinomio de Taylor para una variable mediante polinomios de Taylor de dos variables. (Para más detalles de cómo se obtiene esta aproximación, ver [1].)

Existen varios métodos de Runge-Kutta que se clasifican según del orden de convergencia. Los más sencillos son los de orden 2, entre los cuales tenemos:

1. **Método del punto medio**. Está dado por

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + h\, f\left[t_i + \frac{h}{2}; w_i + \frac{h}{2} f\big(t_i; w_i\big)\right], \end{aligned}$$

   para $i = 0; 1; 2; \ldots; n-1$.

2. **Método de Euler modificado**. Está dado por

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + \frac{h}{2}\Big\{f(t_i; w_i) + f[t_i + h; w_i + hf(t_i; w_i)]\Big\}, \end{aligned}$$

   para $i = 0; 1; 2; \ldots; n-1$.

3. **Método ímplicito ponderado o de Crank-Nicolson**. Está dado por

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + \frac{h}{2}\Big[f\big(t_i; w_i\big) + f\big(t_{i+1}; w_{i+1}\big)\Big], \end{aligned}$$

   para $i = 0; 1; 2; \ldots; n-1$.

4. **Método de Heun**. Está dado por

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + \frac{h}{4}\left\{f\big(t_i; w_i\big) + 3f\left[t_i + \frac{2}{3}h; w_i + \frac{2}{3}h\, f(t_i; w_i)\right]\right\}, \end{aligned}$$

   para $i = 0; 1; 2; \ldots; n-1$.

Paralelamente, los métodos del punto medio y de Crank-Nicolson pueden obtenerse también integrando la función $f(t;y)$ en el intervalo $[t_i; t_{i+1}]$, aplicando las reglas del rectángulo y del trapecio respectivamente. Así, si partimos de la siguiente expresión

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t,y)\mathrm{d}t,$$

y aplicamos la regla del rectángulo para la integral, obtenemos que

$$y(t_{i+1}) = y(t_i) + h\, f\left[t_i + \frac{h}{2}; y(t_i) + \frac{h}{2} f\Big(t_i; y(t_i)\Big)\right],$$

por lo que la aproximación podemos escribirla como

$$w_{i+1} = w_i + h\, f\left[t_i + \frac{h}{2}; w_i + \frac{h}{2} f(t_i; w_i)\right].$$

De forma análoga, si aplicamos la regla del trapecio tenemos

$$y(t_{i+1}) = y(t_i) + \frac{h}{2}\left[f\Big(t_i; y(t_i)\Big) + f\Big(t_{i+1}; y(t_{i+1})\Big)\right],$$

y, nuestra aproximación podemos escribirla como

$$w_{i+1} = w_i + \frac{h}{2}\left[f(t_i; w_i) + f(t_{i+1}; w_{i+1})\right].$$

Para obtener métodos de mayor orden de convergencia, debemos aplicar el teorema 7.4. Con él se obtiene uno de los métodos más usados para resolver ecuaciones diferenciales ordinarias, el de *Runge-Kutta de orden 4*, cuya formulación es la siguiente:

$$\begin{aligned} w_0 &= y_0 \\ k_1 &= hf(t_i; w_i) \\ k_2 &= hf\Big(t_i + \frac{h}{2}; w_i + \frac{1}{2}k_1\Big) \\ k_3 &= hf\Big(t_i + \frac{h}{2}; w_i + \frac{1}{2}k_2\Big) \\ k_4 &= hf\big(t_i + h; w_i + k_3\big) \\ w_{i+1} &= w_i + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \end{aligned}$$

para $i = 0; 1; 2; \ldots; n-1$.

El método de Runge-Kutta de orden 4 tiene un error local de truncamiento $O(h^4)$, siempre que la función $y(t)$ tenga al menos cinco derivadas continuas.

Este método es tan preciso, que programas como el MatLab® y el Mathcad® tienen desarrollados distintas funciones que aplican este método. Por ejemplo, Mathcad® cuenta con la función `rkfixed(`y`;`x_1`;`x_2`;`$npoints$`;`D`)` que resuelve ecuaciones diferenciales de orden uno utilizando dicho método, en la cual y es el valor inicial, x_1 y x_2 son los extremos del intervalo, $npoints$ es la cantidad de intervalos, y entonces $h = \dfrac{x_2 - x_1}{npoints}$, y D es la función $f(x, y)$ que debemos resolver.

Este método puede asociarse a la siguiente formulación:

$$w_{i+1} = w_i + \frac{h}{6}\left[f(t_i; w_i) + 4f\Big(t_{i+\frac{1}{2}}; w_{i+\frac{1}{2}}\Big) + f(t_{i+1}; w_{i+1})\right],$$

equivalente al método de Simpson de integración numérica, cuyo orden de convergencia es $\mathrm{O}(h^4)$.

7.1.7. Métodos de paso múltiple

Los métodos anteriores se basan en obtener los valores siguientes utilizando solamente el valor anterior, sin tener en cuenta los demás valores ya calculados. Es por eso que se denominan de *paso simple*. Pero la pregunta que nos podemos hacer es: si estamos tratando de aproximar una función, tal que se cumpla que $\dfrac{\mathrm{d}\, y}{\mathrm{d}t} = f(t; y)$, por qué no utilizar el conjunto de los valores obtenidos, o al menos un grupo de ellos, para obtener los puntos siguientes.

Esa idea es la que domina a los denominados *métodos de paso múltiple*. El método más sencillo es el denominado *método del salto de rana*, cuya expresión es

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_{i-1} + 2h\, f(t_i; w_i), \end{aligned}$$

para $i = 1; 2; \ldots; n-1$. El valor de w_1 debemos calcularlo con otro método. Como las aproximaciones que obtenemos por el método del salto de rana son del mismo orden que las que se obtienen por cualquier método de Runge-Kutta de orden 2, es conveniente aproximar w_1 con alguno de esos métodos.

Pero existen otros métodos, muy utilizados, que mejoran la notoriamente la aproximación que podemos obtener.

Métodos de Adams

Los métodos de Adams son métodos de paso múltiple muy utilizados. Se dividen en dos grupos: los métodos explícitos, o de *Adams-Bashforth*, y los métodos implícitos, o de *Adams-Moulton*.

En ambos casos, la idea es usar los puntos w_i; w_{i-1}; $\ldots$;w_{i+1-p} para obtener el w_{i+1}, en el caso de los métodos de Adams-Bashforth, en tanto que en los de Adams-Moulton se usan los w_{i+1}; w_i; w_{i-1}; $\ldots$;w_{i+2-p}, donde p es el orden de convergencia. Así, un método de Adams-Bashforth de orden 2 usa los puntos w_i y w_{i-1}, en tanto que un método de Adams-Moulton usa los puntos w_{i+1} y w_i. Veamos como obtener algunos de estos métodos.

Para obtener el método de Adams-Bashforth de orden 2 partimos de la expresión

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y)\mathrm{d}t.$$

Al igual que en el caso de los métodos de Runge-Kutta de orden dos, armemos un polinomio interpolante, pero en este caso, utilizando el método de Newton de diferencias divididas regresivas, para aproximar $f(t; y)$. Así, nos queda que

$$\begin{aligned} f(t;y) &\approx f\Big(t_i; y(t_i)\Big) + \frac{f\Big(t_i; y(t_i)\Big) - f\Big(t_{i-1}; y(t_{i-1})\Big)}{t_i - t_{i-1}}(t - t_i) \\ &\approx f\Big(t_i; y(t_i)\Big) + \frac{f\Big(t_i; y(t_i)\Big) - f\Big(t_{i-1}; y(t_{i-1})\Big)}{h}(t - t_i). \end{aligned}$$

Al integrar el polinomio interpolante obtenemos

$$\begin{aligned} \int_{t_i}^{t_{i+1}} f(t, y)\mathrm{d}t &\approx h\, f\Big(t_i; y(t_i)\Big) + h\, \frac{f\Big(t_i; y(t_i)\Big) - f\Big(t_{i-1}; y(t_{i-1})\Big)}{2}. \\ &\approx \frac{h}{2}\left[3f\Big(t_i; y(t_i)\Big) - f\Big(t_{i-1}; y(t_{i-1})\Big)\right]. \end{aligned}$$

y si reemplazamos en la expresión inicial, tenemos

$$y(t_{i+1}) = y(t_i) + \frac{h}{2}\left[3f\Big(t_i; y(t_i)\Big) - f\Big(t_{i-1}; y(t_{i-1})\Big)\right].$$

Como siempre, lo que buscamos es una aproximación de $y(t_{i+1})$, entonces el método de Adams-Bashforth de orden 2 queda formulado de la siguiente manera:

$$w_{i+1} = w_i + \frac{h}{2}\left[3f(t_i; w_i) - f(t_{i-1}; w_{i-1})\right],$$

para $i = 1; 2; \ldots; n-1$. Por lo tanto, debemos calcular w_1 con algún otro método, por ejemplo, el de Runge-Kutta de orden 2.

De todos los métodos que se pueden desarrollar, uno de los métodos de Adams-Bashforth más usados es el de orden 4, cuya expresión es

$$w_{i+1} = w_i + \frac{h}{24}\left[55f(t_i; w_i) - 59f(t_{i-1}; w_{i-1}) + 37f(t_{i-2}; w_{i-2}) - 9f(t_{i-3}; w_{i-3})\right],$$

para $i = 3; 4; \ldots; n-1$. Nuevamente, debemos hallar w_1; w_2 y w_3 con ayuda de otro método. Al igual que en el método de orden 2, en este caso podemos usar el RK O4.

Para obtener los métodos de Adams-Moulton, procedemos de forma análoga. Para obtener el de orden 2, planteemos el siguiente polinomio interpolante para aproximar $f(t; y)$:

$$\begin{aligned} f(t;y) &\approx f\Big(t_{i+1}; y(t_{i+1})\Big) + \frac{f\Big(t_{i+1}; y(t_{i+1})\Big) - f\Big(t_i; y(t_i)\Big)}{t_{i+1} - t_i}(t - t_{i+1}) \\ &\approx f\Big(t_{i+1}; y(t_{i+1})\Big) + \frac{f\Big(t_{i+1}; y(t_{i+1})\Big) - f\Big(t_i; y(t_i)\Big)}{h}(t - t_{i+1}). \end{aligned}$$

Al integrarlo, obtenemos que

$$\begin{aligned} \int_{t_i}^{t_{i+1}} f(t,y)\mathrm{d}t &\approx h\, f\Big(t_{i+1}; y(t_{i+1})\Big) + h\, \frac{f\Big(t_{i+1}; y(t_{i+1})\Big) - f\Big(t_i; y(t_i)\Big)}{2}. \\ &\approx \frac{h}{2}\left[f\Big(t_{i+1}; y(t_{i+1})\Big) + f\Big(t_i; y(t_i)\Big)\right]. \end{aligned}$$

Nuevamente, al reemplazar en la expresión inicial, tenemos

$$y(t_{i+1}) = y(t_i) + \frac{h}{2}\left[f\Big(t_{i+1}; y(t_{i+1})\Big) + f\Big(t_i; y(t_i)\Big)\right],$$

y como lo que buscamos es una aproximación de $y(t_{i+1})$, tenemos que

$$w_{i+1} = w_i + \frac{h}{2}\left[f(t_{i+1}; w_{i+1}) + f(t_i; w_i)\right].$$

Resulta interesante ver que el método de Adams-Moulton de orden 2 es el método de Crank-Nicolson.

Al igual que con Adams-Bashforth, uno de los métodos más usados es el de Adams-Moulton de orden 4, cuya expresión es

$$w_{i+1} = w_i + \frac{h}{24}\left[9f(t_{i+1}; w_{i+1}) + 19f(t_i; w_i) - 5f(t_{i-1}; w_{i-1}) + f(t_{i-2}; w_{i-2})\right],$$

para $i = 2; 3; \ldots; n-1$. Nuevamente, debemos obtener w_1 y w_2 con ayuda del RK O4.

El uso de los métodos de Adams-Moulton conlleva la necesidad de reformular la expresión para convertirla en un método explícito. Como esto no siempre es posible, una forma de aplicarlo es mediante la combinación de un método de Adams-Bashforth y uno de Adams-Moulton, ambos del mismo orden de convergencia. Esta combinación se conoce como *método predictor-corrector de Adams*. Por ejemplo, el método predictor-corrector de orden 2 es el siguiente:

$$\begin{aligned} w^*_{i+1} &= w_i + \frac{h}{2}\left[3f(t_i; w_i) - f(t_{i-1}; w_{i-1})\right] \\ w_{i+1} &= w_i + \frac{h}{2}\left[f(t_{i+1}; w^*_{i+1}) + f(t_i; w_i)\right], \end{aligned}$$

para $i = 2; 3; \ldots; n-1$ y donde el valor de $w_1 = w^*_1$ debemos obtenerlo usando el método de Runge-Kutta de orden 2 o resolviendo en forma explícita la segunda ecuación del método. Uno de los métodos más usados es el predictor-corrector de Adams de orden 4, cuya expresión es:

$$\begin{aligned} w^*_{i+1} &= w_i + \frac{h}{24}\left[55f(t_i; w_i) - 59f(t_{i-1}; w_{i-1}) + 37f(t_{i-2}; w_{i-2}) - 9f(t_{i-3}; w_{i-3})\right] \\ w_{i+1} &= w_i + \frac{h}{24}\left[9f(t_{i+1}; w^*_{i+1}) + 19f(t_i; w_i) - 5f(t_{i-1}; w_{i-1}) + f(t_{i-2}; w_{i-2})\right], \end{aligned}$$

para $i = 4; 5; \ldots; n-1$ y donde $w_1 = w_1^*$, $w_2 = w_3^*$ y $w_3 = w_3^*$ los obtenemos usando RK O4.

En general, suelen ser más precisos los métodos de Adams-Moulton que los de Adams-Bashforth. El de Adams-Moulton de orden 4 entrega resultados muy parecidos, en precisión, al método de Runge-Kutta de orden 4. Sin embargo, por una cuestión de sencillez al momento de programar, los paquetes de software prefieren incluir este último y no el método de Adams-Moulton de orden 4.

7.2. Ecuaciones diferenciales ordinarias con condiciones de contorno

7.2.1. Introducción

En el punto anterior hemos visto los diferentes métodos numéricos para la resolución de ecuaciones diferenciales ordinarias con valores inciales. Estos métodos son principalmente para resolver ecuaciones diferenciales de primer orden, tanto lineales como no lineales, y pueden ser adaptados para ecuaciones de orden superior, siempre que éstas sean de valores iniciales.

Cuando en vez de ecuaciones diferenciales de primer orden debemos resolver numéricamente ecuaciones diferenciales de orden dos o superior, los métodos aplicados para el caso anterior pueden utilizarse casi directamente si la expresión de la ecuación diferencial es:

$$\frac{\mathrm{d}^n y}{\mathrm{d}t^n} = f(t, y, y', \ldots, y^{(n-1)}), \text{ en } [a, b];$$

con las condiciones iniciales:

$$y(a) = \alpha;\ y'(a) = \beta; \ldots; y^{(n-1)}(a) = \gamma,$$

podemos aplicar un método como el Runge-Kutta de orden cuatro, pero con algunas transformaciones para poder adaptarlo al problema que debe resolverse. Estas transformaciones requieren un cambio de variable, como por ejemplo:

$$z_1(t) = y'(t); z_2(t) = z_1'(t); \ldots; z_n(t) = z_{n-1}'(t).$$

Supongamos que tenemos una ecuación diferencial de orden 2 expresada como:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = f(t, y, y'), \text{ en } [a, b];$$

con las siguientes condiciones iniciales:

$$y(a) = \alpha;\ y'(a) = \beta.$$

Para resolver esta ecuación por medio de alguno de los métodos estudiados, debemos transformar la ecuación diferencial en un sistema de dos ecuaciones diferenciales de primer orden, por medio de un cambio de variable; en este caso $y' = z$. Con este cambio de variable, y teniendo en cuenta que lo que buscamos es una aproximación de y que llamaremos u, por lo que también se cumple que $u' = z$, el esquema de resolución queda:

$$\begin{aligned} u'(t) &= z(t) \\ z'(t) &= f(t, u, z), \end{aligned}$$

con $u(a) = \alpha$ y $z(a) = \beta$.

A partir de aquí, se debe resolver el par de ecuaciones en forma simultánea para poder obtener el valor de $u(t)$ buscado. Si, por ejemplo, aplicamos el método de Euler para resolver las ecuaciones diferenciales de primer orden, el esquema iterativo queda de la siguiente manera:

$$\begin{aligned} u_{i+1} &= u_i + h \cdot z(t) \\ z_{i+1} &= z_i + h \cdot f(t_i, u_i, z_i), \end{aligned}$$

con el cual obtenemos en cada paso el nuevo valor de $u(t)$ y de $\frac{\mathrm{d}\,u}{\mathrm{d}t}$ representado por $z(t)$.

En general, las ecuaciones diferenciales de orden dos o superior no son de valores iniciales sino de valores de contorno o frontera. Es decir, no disponemos de todos los valores para $t = a$, sino que tenemos valores para $t = a$ y $t = b$. Por lo tanto, una ecuación diferencial de orden 2, por ejemplo, está dada por:

$$\frac{\mathrm{d}^2\,y}{\mathrm{d}t^2} = f(t, y, y'), \text{ en } [a, b];$$

con las condiciones en los extremos del intervalo:

$$y(a) = \alpha;\ y(b) = \beta.$$

Con estas condiciones no nos es posible utilizar los métodos estudiados, ni siquiera transformando la ecuación diferencial en un sistema de ecuaciones diferenciales. Debemos buscar otra forma para aproximar nuestra ecuación diferencial y obtener los resultados de la función $y(t)$.

¿Es necesario analizarlo con más detalle el procedimiento para resolverlas? Como dato importante, basta mencionar que buena parte de los problemas que debemos resolver los ingenieros, ya sean civiles, mecánicos, electrónicos,etc., están expresados en términos de ecuaciones diferenciales de orden superior. Un ejemplo que suele ser muy usado es el caso de la ecuación diferencial de equilibrio para una viga, dada por la expresión:

$$EI\frac{\mathrm{d}^4\,w}{\mathrm{d}x^4} - p(x) = 0.$$

que requiere de cuatro condiciones de contorno para ser resuelta. Estas condiciones pueden ser:

1. Condiciones de borde esenciales (Dirichlet);
2. Condiciones de borde naturales (Neumann);
3. Una combinación de ambas.

Por ejemplo, para una viga doblemente empotrada, de longitud L, como se ve en la figura, las condiciones de borde son:

$$w(0) = 0;\ w(L) = 0;\ w'(0) = 0;\ \text{y}\ w'(L) = 0.$$

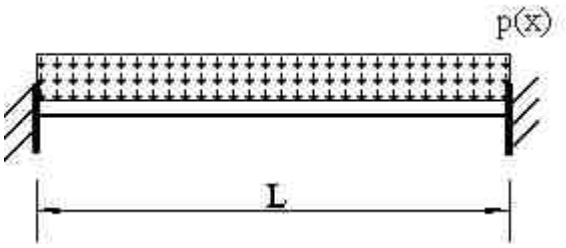


Figura 7.1: Viga doblemente empotrada

Este es el típico caso de condiciones de borde esenciales (o forzadas), puesto que las restricciones están asociadas a los desplazamientos y los giros en los extremos de la viga. Esta ecuación no es posible resolverla aplicando en forma directa los métodos mencionados anteriormente. En consecuencia, para poder aproximar una solución, debemos buscar alguna forma de adaptar los métodos vistos para tener en cuenta estas condiciones de frontera o de contorno.

Veremos a continuación dos métodos que pueden usarse para resolver este tipo de ecuaciones diferenciales. Empezaremos por el más sencillo, el método del disparo lineal, que hace uso de los métodos ya estudiados.

7.2.2. Método del tiro o disparo lineal

Supongamos que tenemos la siguiente ecuación diferencial:

$$y'' = -f(t, y),\ t \in [0; 1];$$

que debe cumplir con las condiciones:

$$y(0) = y_0;\ y(1) = y_1.$$

Como vemos, no tenemos dos condiciones iniciales, sino una para el valor inicial y otra para el valor final que debe tomar la función buscada.

Para encarar el problema haremos una modificación. Resolveremos el siguiente problema de valores iniciales, suponiendo que lo que buscamos es una aproximación a $y(t)$ que llamaremos $u(t_i)$. Entonces nuestro sistema quedará de la siguiente forma:

$$u'' = -f(t, u),\ u_1(0) = y_0 \text{ y } u_1'(0) = \alpha_1.$$

donde α_1 es el primer ensayo para $u_1'(0)$. Apliquemos para ello cualquiera de los métodos vistos anteriormente, por ejemplo el de Euler. Con él obtendremos un valor para $u_1(1)$ igual a β_1, que seguramente será distinto a y_1.

Nuevamente, resolvamos con Euler un sistema similar pero proponiendo que $u_2(0) = y_0$ y $u_2'(0) = \alpha_2$. Obtendremos otro valor para $u(1)$, es decir, un $u_2(1) = \beta_2$, probablemente distinto a y_1.

En consecuencia, tendremos dos aproximaciones de y_1. Para continuar, vamos a suponer que existe una relación lineal entre $u(t_i)$, $u_1(t_i)$ y $u_2(t_i)$. Esta relación lineal estará dada por:

$$\frac{u(t_i) - u_1(t_i)}{y_1 - \beta_1} = \frac{u_2(t_i) - y_0}{\beta_2 - y_0}.$$

Para calcular $u(t)$ debemos despejarla de la expresión anterior. Así obtenemos:

$$u(t_i) = u_1(t_i) + \frac{y_1 - \beta_1}{\beta_2 - y_0}\left[u_2(t_i) - y_0\right].$$

Para entender como opera el método, veamos un ejemplo práctico, resolviendo una ecuación diferencial de orden 2.

Ejemplo

Resolver la siguiente ecuación diferencial ordinaria con valores de frontera, aplicando el método de Euler:

$$y'' = 4(y - x);\ 0 \le x \le 1;$$

con los valores de contorno:

$$y(0) = 0;\ y(1) = 2.$$

Para resolver la ecuación por el método de Euler plantearemos primero que $y'(x) = z(x)$, con lo que tendremos que la ecuación diferencial se transforma en:

$$\begin{aligned} y'(x) &= z(x) \\ z'(x) &= 4(y - x) \end{aligned}$$

Si aplicamos el método de Euler, y hacemos $u_i = y(x_i)$ tendremos las siguientes ecuaciones:

$$\begin{aligned} u_{i+1} &= u_i + h \cdot z_i \\ z_{i+1} &= z_i + h \cdot 4(u_i - x_i) \end{aligned}$$

Tabla 7.1: Resultados obtenidos aplicando el método de Euler

$\mathbf{x_i}$	$\mathbf{z_{1,i}}$	$\mathbf{v_{1,i}}$	$\mathbf{z_{2,i}}$	$\mathbf{v_{2,i}}$	$\mathbf{u_i}$	$\mathbf{y(x_i)}$	$\mathbf{e}$
$0,00$	$0,000$	$0,000$	$1,000$	$0,000$	$0,000$	$0,000$	$0,0$
$0,10$	$0,000$	$0,000$	$1,000$	$0,100$	$0,252$	$0,156$	$9,7 \cdot 10^{-2}$
$0,20$	$-0,040$	$0,000$	$1,000$	$0,200$	$0,504$	$0,313$	$1,9 \cdot 10^{-1}$
$0,30$	$-0,120$	$-0,004$	$1,000$	$0,300$	$0,752$	$0,476$	$2,8 \cdot 10^{-1}$
$0,40$	$-0,242$	$-0,016$	$1,000$	$0,400$	$0,992$	$0,645$	$3,5 \cdot 10^{-1}$
$0,50$	$-0,408$	$-0,040$	$1,000$	$0,500$	$1,220$	$0,824$	$4,0 \cdot 10^{-1}$
$0,60$	$-0,624$	$-0,081$	$1,000$	$0,600$	$1,432$	$1,016$	$4,2 \cdot 10^{-1}$
$0,70$	$-0,896$	$-0,143$	$1,000$	$0,700$	$1,621$	$1,225$	$4,0 \cdot 10^{-1}$
$0,80$	$-1,234$	$-0,233$	$1,000$	$0,800$	$1,784$	$1,455$	$3,3 \cdot 10^{-1}$
$0,90$	$-1,647$	$-0,356$	$1,000$	$0,900$	$1,913$	$1,711$	$2,0 \cdot 10^{-1}$
$1,00$	$-2,150$	$-0,521$	$1,000$	$1,000$	$2,000$	$2,000$	$0,0$

Como vemos, debemos resolver dos ecuaciones para obtener el valor de u_{i+1}. Por ello, en primer término, vamos a resolver el sistema obteniendo, primero, valores para unas funciones $v_1(x)$ y adoptando las siguientes condiciones iniciales:

$$v_1(0) = 0;\ z_1(0) = 0$$

por lo que el sistema a resolver será:

$$\begin{aligned} v_{1_{i+1}} &= v_{1_i} + h \cdot z_{1_i} \\ z_{1_{i+1}} &= z_{1_i} + h \cdot 4(v_{1_i} - x_i) \end{aligned}$$

En segundo término, haremos lo mismo pero para las funciones $v_2(x)$ y $z_2(x) = v_2'(x)$ con los valores de contorno levemente distintos. Estos son:

$$v_2(0) = 0;\ z_2(0) = 1,$$

y el sistema a resolver será:

$$\begin{aligned} v_{2_{i+1}} &= v_{2_i} + h \cdot z_{2_i} \\ z_{2_{i+1}} &= z_{2_i} + h \cdot 4(v_{2_i} - x_i) \end{aligned}$$

Con los valores para cada una de las soluciones y por cada iteración, calcularemos los valores definitivos mediante la expresión:

$$u_i = v_{1_i} + \frac{y(1) - v_1(1)}{v_2(1) - y(0)} \left[v_{2_i} - y(0)\right]$$

En la tabla 7.1 podemos ver los resultados obtenidos.

En la penúltima columna podemos ver el valor *exacto* de la función buscada, dado que la solución analítica de la ecuación diferencial es:

$$y(x) = e^2 \left(e^4 - 1\right)^{-1} \left(e^{2x} - e^{-2x}\right) + x.$$

Los valores de $u(x_i)$ obtenidos no son muy precisos, dado que el método utilizado para resolver el sistema de ecuaciones es el de Euler, pero igualmente sirven como demostración de la efectividad al aplicar este método. Podemos ver que la última columna muestra el error absoluto entre el valor obtenido numéricamente y el valor exacto. Observemos que el error cometido es, del orden de 10^{-1}, un error razonable para este método. (Recordemos que el método de Euler tiene un error $O(h)$.)

7.2.3. Diferencias finitas

En el punto anterior hemos resuelto una ecuación diferencial lineal con condiciones de contorno utilizando un método de resolución que transforma las condiciones de contorno en condiciones iniciales. Sin embargo, este método tiene como desventaja que es inestable en ciertas ocasiones. Por lo que su utilización se ve reducida generalmente a unos pocos casos o problemas.

Uno de los métodos más aplicados para aproximar una solución de ecuaciones diferenciales de orden mayor o igual a dos, es el que reemplaza las derivadas por diferencias finitas mediante un cociente de diferencias, tal como vimos en diferenciación numérica. La aplicación de estas *diferencias finitas* generan un sistema de ecuaciones lineales del tipo $Ax = B$, sistema que puede resolverse mediante alguno de los métodos ya vistos. Está claro que estamos limitados en la elección de nuestro intervalo h, que no puede ser muy chico. Veamos en qué consiste el método, aplicándolo a nuestro ejemplo anterior.

Para aproximar las derivadas, tomaremos el método de las *diferencias centradas*, que permiten una mejor aproximación de las derivadas. Para empezar, desarrollemos $y(x_{i+1})$ y $y(x_{i-1})$ por Taylor hasta el cuarto término, por lo que tendremos:

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(iv)}(\xi_i^+),$$

para alguna ξ_i^+ en $(x_i; x_{i+1})$, y

$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(iv)}(\xi_i^-),$$

para alguna ξ_i^- en $(x_{i-1}; x_i)$. Demás está decir que se supone que $y(x) \in C^4[x_{i-1}; x_{i+1}]$. Si sumamos ambas expresiones y despejamos $y''(x_i)$, tendremos:

$$y''(x_i) = \frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] - \frac{h^2}{24}\left[y^{(iv)}(\xi_i^+) + y^{(iv)}(\xi_i^-)\right].$$

Si aplicamos el teorema del valor medio, podemos simplificar la expresión a:

$$y''(x_i) = \frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] - \frac{h^2}{12}y^{(iv)}(\xi_i),$$

para alguna ξ_i en $(x_{i-1}; x_{i+1})$.

Reemplacemos esta última expresión en nuestra ecuación diferencial:

$$\frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] - \underbrace{\frac{h^2}{12}y^{(iv)}(\xi_i)}_{O(h^2)} = 4\left[y(x_i) - x_i\right].$$

De esta manera, nuestra ecuación diferencial se transforma en:

$$\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] = 4h^2\left[y(x_i) - x_i\right],$$

y desarrollando algebraicamente, obtenemos:

$$\left[y(x_{i-1}) - 2(1 + 2h^2)y(x_i) + y(x_{i+1})\right] = -4h^2x_i,$$

por lo tanto, para cada i tenemos una ecuación lineal. Definamos, entonces, el intervalo o *paso* h como $\frac{b-a}{N}$ siendo $N > 0$; de esta manera obtendremos N intervalos para $i \in [0; N]$. Con i y h podemos armar nuestro sistema de ecuaciones para $i \in [1; N-1]$. La matriz resultante será:

$$A = \begin{bmatrix} 1 & -2(1+2h^2) & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -2(1+2h^2) & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -2(1+2h^2) & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & -2(1+2h^2) & 1 \end{bmatrix}.$$

Si hacemos que $y_i = y(x_i)$ tendremos que nuestras incógnitas son:

$$y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix}.$$

Nuestro vector de términos independientes será:

$$B = \begin{bmatrix} -4h^2x_1 \\ -4h^2x_2 \\ \vdots \\ -4h^2x_{N-2} \\ -4h^2x_{N-1} \end{bmatrix}.$$

Pero hemos armado un sistema con $N-2$ filas y N incógnitas y_i. Para completar el sistema debemos recordar que $y_0 = \alpha$ y $y_N = \beta$, por lo que nuestro sistema de ecuaciones lineales quedará como:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2(1+2h^2) & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -2(1+2h^2) & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -2(1+2h^2) & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 & -2(1+2h^2) & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} \alpha \\ -4h^2x_1 \\ -4h^2x_2 \\ \vdots \\ -4h^2x_{N-2} \\ -4h^2x_{N-1} \\ \beta \end{bmatrix}.$$

Armemos el sistema definitivo con $x \in [0;1]$, $y(0) = y_0 = 0$, $y(1) = y_N = 2$ y $N = 10$. Con estos parámetros tendremos que $h = \frac{1-0}{10} = 0,1$. Entonces, en la matriz A tendremos el coeficiente (además de 1):

$$-2\left[1 + 2(0,1)^2\right] = -2\left(1 + 0,02\right) = -2,04;$$

y en el vector de términos independientes:

$$-4(0+0,1)^2 \cdot 0,1i = -4(0,1)^2 \cdot 0,1i = -i \cdot 4(0,1)^3;$$

con $i \in [1, N-1]$. El sistema definitivo quedará con la matriz de coeficientes:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2,04 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2,04 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2,04 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2,04 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2,04 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2,04 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2,04 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2,04 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2,04 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

y con el vector de términos independientes:

$$B = \begin{bmatrix} 0 \\ -0,004 \\ -0,008 \\ -0,012 \\ -0,016 \\ -0,020 \\ -0,024 \\ -0,028 \\ -0,032 \\ -0,036 \\ 2 \end{bmatrix}.$$

Al resolver el sistema de ecuaciones por alguno de los métodos numéricos que hemos estudiado en el capítulo 2, obtenemos el siguiente vector solución:

$$y = \begin{bmatrix} 0,000 \\ 0,156 \\ 0,313 \\ 0,476 \\ 0,645 \\ 0,824 \\ 1,017 \\ 1,225 \\ 1,455 \\ 1,711 \\ 2,000 \end{bmatrix}.$$

Si lo comparamos con el vector y obtenido con el método de Euler, podemos observar que la solución por diferencias finitas es mucho más precisa, ya que los y obtenidos son *iguales* a los hallados por aplicación de la solución analítica.

7.2.4. El método de los elementos finitos

El método de los elementos finitos es un método numérico para aproximar soluciones de ecuaciones diferenciales, que surgió asociado al análisis estructural, que luego se expandió (y aún se expande)a otras ramas de la ingeniería. Orginalmente se basó en la aplicación del principio de los trabajos virtuales, pero luego evolucionó hacia la aplicación de principios variacionales.

Para explicar el método, nos ayudaremos de nuestro ejemplo de la viga doblemente empotrada. La ecuación diferencial de equilibrio que resuelve el problema es la siguiente:

$$\frac{\mathrm{d}^2\, M(x)}{\mathrm{d}x^2} + p(x) = 0,$$

donde $M(x)$ es la función *Momento flector* y $p(x)$ es la función de carga, distribuida a lo largo de la viga, que en nuestro ejemplo es uniforme (constante).

Por el principio de los trabajos virtuales, sabemos que si un sistema está en equilibrio, el trabajo de ese sistema a lo largo de un desplazamiento virtual es nulo. Con nuestra ecuación diferencial de equilibrio, el principio de los trabajos virtuales toma la siguiente forma:

$$\int_L \left[\frac{\mathrm{d}^2\, M(x)}{\mathrm{d}x^2} + p(x)\right] \delta w(x)\, \mathrm{d}x,$$

donde $\delta\, w(x)$ es la función desplazamiento virtual, que debe ser compatible con los vínculos. La intergral representa el trabajo virtual del sistema en equilibrio y por lo tanto debe ser nula.

También sabemos que para el caso de la viga sometida a flexión se cumple la siguiente relación:

$$M(x) = -EI\frac{\mathrm{d}^2\, w(x)}{\mathrm{d}x^2},$$

donde $w(x)$ es la función desplazamiento real del problema, también conocida como *elástica de deformación*, E es el módulo de elasticidad del material e I el momento de inercia estático.

Si reemplazamos en nuestra ecuación de los trabajos virtuales, tendremos:

$$\int_L \left[EI\frac{\mathrm{d}^4\, w(x)}{\mathrm{d}x^4} - p(x)\right]\delta w(x)\,\mathrm{d}x,$$

con lo cual nuestra ecuación diferencial original de orden 2 se transformó en una ecuación de orden 4. Esta tranformación supone, entonces, que se necesitan cuatro condiciones para resolverla y obtener la función desplazamiento $w(x)$ [4]. En nuestro caso las cuatro condiciones son de borde o de contorno, a saber:

$$w(0) = w(L) = 0;\ w'(0) = w'(L) = 0.$$

Para resolver esta ecuación, operemos algebraicamente en ella. Si distribuimos el producto del integrando, obtendremos lo siguiente:

$$\int_L EI\frac{\mathrm{d}^4\, w(x)}{\mathrm{d}x^4}\delta w(x)\,\mathrm{d}x - \int_L p(x)\,\delta w(x)\,\mathrm{d}x = 0.$$

Como paso siguiente, integremos por partes la primera integral. Así, nuestra ecuación quedará de la siguiente forma:

$$\int_L EI\frac{\mathrm{d}^2\, w(x)}{\mathrm{d}x^2}\frac{\mathrm{d}^2\delta w(x)}{\mathrm{d}x^2}\mathrm{d}x + EI\frac{\mathrm{d}^3\, w(x)}{\mathrm{d}x^3}\delta w(x)\Big|_L - EI\frac{\mathrm{d}^2\, w(x)}{\mathrm{d}x^2}\frac{\mathrm{d}\,\delta w(x)}{\mathrm{d}x}\Big|_L - \int_L p(x)\,\delta w(x)\,\mathrm{d}x = 0,$$

que suele escribirse también como

$$\int_L EI\, w''(x)\delta w''(x)\mathrm{d}x + EI\, w'''(x)\delta w(x)\big|_L - EI\, w''(x)\delta w'(x)\big|_L - \int_L p(x)\,\delta w(x)\,\mathrm{d}x = 0.$$

Si reagrupamos los términos, nos queda.

$$\int_L EI\, w''(x)\delta w''(x)\mathrm{d}x - \int_L p(x)\,\delta w(x)\,\mathrm{d}x + EI\, w'''(x)\delta w(x)\big|_L - EI\, w''(x)\delta w'(x)\big|_L = 0.$$

Como dijimos que la función desplazamiento virtual debe ser compatible con los vínculos, entonces debe cumplir que:

$$\delta w(0) = \delta w(L) = 0;\ \delta w'(0) = \delta w'(L) = 0.$$

Con estas condiciones podemos ver que los dos términos de la derecha se anulan, es decir, tenemos que:

$$EI\, w'''(0)\delta w(0) = 0; EI\, w'''(L)\delta w(L) = 0; EI\, w''(0)\delta w'(0) = 0; EI\, w''(L)\delta w'(L) = 0$$

.

Tengamos en cuenta que $EI\, w'''(0)$ es el esfuerzo de corte en $x = 0$ $(V(0))$, $EI\, w'''(L)$ es el esfuerzo de corte en $x = L$ $(V(L))$, $EI\, w''(0)$ es el momento flector en $x = 0$ $(M(0))$ y $EI\, w''(L)$ es el momento flector en $x = L$ $(M(L))$.

[4] Tengamos en cuenta que la ecuación original tiene dos condiciones, pero de difícil aplicación, puesto que dependen de la solución. No es posible definirlas «a priori».

7.3. Notas finales

Casi podría decirse que todos los problemas que debe enfrentar un ingeniero pueden formularse mediante ecuauciones diferenciales. Desde el análisis estructural hasta el diseño de un avión de pasajeros, las ecuaciones diferenciales intervenienen en forma explícitas (deben ser resueltas) o e forma implícita (se aplican soluciones análiticas de dichas ecuaciones).

Hasta la mitad del siglo XX, muchas de las limitaciones en los aspectos ingenieriles estaban dados por las pocas soluciones analíticas que se podían obtener de muchas de las ecuaciones diferenciales, y en consecuencia, se dependía de los ensayos en modelos físicos o en prototipos. Con el desarrollo de las computadoras, a partir de los años '50, y principalmente, con la aparición de las computadoras personales hace 25 años, obtener soluciones aproximadas de las ecuaciones diferenciales dejó de ser un escollo en cuanto a tiempo de cálculo. Prácticamente todas las disciplinas científicas y tecnológicas basan sus soluciones en la aplicación de métodos numéricos.

Dentro del conjunto de métodos numéricos para resolver ecuaciones diferenciales, los métodos de las diferencias finitas y de los elementos finitos, y en particular este último, son los más usados para encarar soluciones aproximadas. Y en los últimos años, la gran capacidad de cálculo de las computadoras han permitido adentrarse en la resolución aproximada de problemas con ecuaciones diferenciales no lineales, permitiendo el estudio de muchos fenómenos que antes se consideraban como «imposibles» de abordar. Basta con ver el avance en el campo de los estudios climáticos, el comportamiento de los ríos, el avance en la hidráulica marítima, etc., que han reemplazado el uso de modelos físicos (muy caros y lerdos) por modelos matemáticos (más baratos y rápidos).

Bibliografía

[1] Burden, R.L. & Faires, J.D. *Análisis Numérico*. Sexta Edición, International Thomson, 1998.

[2] Gavurin, M.K. *Conferencias sobre los métodos de cálculo*. Editorial Mir, 1973.

[3] Goldberg, D. *What every Computer Scientist should know about Floating-Point Arithmetic*. ACM Computing Surveys, March 1991.

[4] González, H. *Análisis Numérico, primer curso*. Primera Edición, Nueva Librería, 2002.

[5] Higham, N.J. *Accuracy and Stability of Numerical Algorithms*. SIAM, 1996.

[6] Higham, N.J. *How accurate is Gaussian Elimination*. Numerical Analysis 1989, Proceedings of the 13th Dundee Conference, volume 228 of Pitman research Notes in Mathematics.1990.

[7] Higham, N. J. *The numerical stability of barycentric Lagrange interpolation*. IMA Journal of Numerical Analysis. 2004.

[8] Marshall, G. *Solución numérica de ecuaciones diferenciales, Tomo I*. Editorial Reverté S.A., 1985.

[9] Saad, Y. *Iterative Methods for Sparse Linear Systems*. Second Edition, 2000.

[10] Samarski, A.A. *Introducción a los métodos numéricos*. Editorial Mir, 1986.

[11] Shewchuk, J. R. *An introduction to the Conjugate Gradient Method without the agonizing pain*. Edition $1\frac{1}{4}$. School of Computer Science. Carnegie Mellon University.

[12] Trefethen, L.N. *The Definition of Numerical Analysis*. SIAM News. November 1992.

[13] Trefethen, L.N. *Numerical Analysis*. Princeton Companion to Mathematics, to appear.

[14] Trefethen, L.N. & Berrut, J.P. *Barycentric Lagrange Interpolation*. 2004.

[15] Zill, D. G. *Ecuaciones diferenciales con aplicaciones de modelado*. Séptima Edición, International Thomson, 2002.