Invitación al Análisis Numérico

Carlos Enrique Mejía Salazar
Universidad Nacional de Colombia, Medellín
Escuela de Matemáticas

Julio 2002

Contenido

Prefacio

En estas notas ofrecemos una variedad de temas del análisis numérico, que pueden servir de motivación y de guía a estudiantes de nuestra Universidad y a otros interesados. Se pueden considerar como unas notas de clase, pero, para la selección de temas y enfoques, no se siguió un programa de curso dado.

Los más de 100 ejercicios están repartidos en pequeñas listas en medio de la exposición, ejercicios suplementarios y exámenes de entrenamiento. Hay cerca de 30 ejemplos resueltos en detalle, que casi siempre aparecen con su rutina MATLAB acompañante. Dichas rutinas se presentan únicamente con fines didácticos y no sustituyen software de calidad.

Pretendemos publicar estas notas en la WEB y mantenerlas en evolución. Por eso los aportes de usuarios son muy importantes. Comentarios, correcciones y sugerencias son bienvenidos en la dirección electrónica

cemejia@perseus.unalmed.edu.co.

Expresamos nuestro reconocimiento a la Universidad Nacional de Colombia, por habernos concedido el año sabático durante el cual escribimos estas notas.

Parte de este documento se utilizará como guía en un cursillo de la XIII Escuela Latinoamericana de Matemáticas, que se reunirá en Cartagena, Colombia de julio 29 a agosto 3 de 2002.

Medellín, julio 15 de 2002

1

Presentación y repaso

Motivación
Bosquejo del contenido
Repaso

1.1 Motivación

Los métodos numéricos son parte importante de la interacción entre matemáticas, ingeniería e industria. Se estudian cada día más, en gran parte debido a que la necesidad de modelamiento matemático en la ciencia y la técnica crece de forma vertiginosa.

Es común escuchar que los métodos numéricos se utilizan por no disponer de soluciones analíticas para la mayoría de los problemas de la matemática aplicada. Por supuesto ésto es verdad, pero es importante tener en cuenta que casi siempre las soluciones analíticas se utilizan discretizadas y/o truncadas. Generalmente es una pérdida de tiempo resolver un problema analíticamente para después obtener una aproximación de tal solución, en lugar de optar desde el principio por una solución numérica.

Definir el análisis numérico no es tarea fácil y sobre esta materia existen todavía discrepancias, descritas por Trefethen en el Apéndice a su libro Trefethen y Bau (1997) [34]. La definición que propone Trefethen, muy cercana a la propuesta por Henrici desde 1964 en Henrici (1964) [15], es la siguiente: *El análisis numérico es el estudio de algoritmos para la solución de problemas de la matemática continua.* Por matemática *continua*, Trefethen se refiere al análisis, real y complejo. Utiliza esta palabra como opuesta a *discreta.*

Por su parte, la *computación científica*, puede definirse como *el diseño e implementación de algoritmos numéricos para problemas de ciencias e ingeniería.* De manera que podemos decir que el análisis

numérico es un pre-requisito para la computación científica. En estas notas, hacemos de cuenta que el problema matemático a resolver ya ha sido definido. No nos ocupamos del problema físico o de ingeniería directamente. Aquí nos concentramos en algoritmos y uso de software. Además, es en medio de ejemplos concretos que presentamos las nociones de estabilidad y error, indispensables para evaluar la calidad de un método numérico. Para compensar por las obligatorias omisiones, ofrecemos bibliografía que permita profundizar en los temas a los interesados.

Frecuentemente, los métodos numéricos, muy en especial los que sirven para resolver ecuaciones diferenciales, conducen a problemas de álgebra lineal en los que las matrices tienen alguna estructura y la mayoría de sus elementos son nulos. Para estos problemas, el álgebra lineal numérica ofrece métodos especiales, en los que se está trabajando intensamente desde hace varios años. Basta ver, por ejemplo, las Memorias de Copper Mountain Conference del año 2000 [35], que contienen 6 artículos sobre precondicionamiento, 6 sobre cálculo de valores propios y 3 sobre métodos multigrid. Esta es una conferencia organizada anualmente por University of Colorado en cooperación con SIAM (Society for Industrial and Applied Mathematics), que trata cada dos años sobre métodos multigrid y en el año intermedio sobre métodos iterativos en general.

Los temas del álgebra lineal numérica son de gran importancia dentro del análisis numérico. Dentro de estas notas veremos problemas que al discretizarlos se convierten en problemas de álgebra lineal que aquí resolvemos utilizando a MATLAB. Pero este tema es tan importante, que amerita consideración especial. Esperamos en un futuro cercano complementar estas notas con unas acerca de álgebra lineal numérica. Lo único que haremos, por ahora, es que al final de estas notas, comentamos brevemente sobre software especializado que invitamos a conocer a los interesados.

1.2 Bosquejo del contenido

La organización de estas notas es la siguiente: Enseguida, en este mismo capítulo, presentamos una sección de repaso que hasta ahora solo con-

tiene material sobre normas en espacios vectoriales. Vislumbramos que la sección de repaso crecerá un poco posteriormente. En el próximo capítulo introducimos el software MATLAB, que es el que escogimos para ejemplos y ejercicios en estas notas. En el capítulo siguiente se presenta la solución de ecuaciones no lineales por varios métodos, incluyendo el método de Newton. En los dos capítulos que siguen nos ocupamos de interpolación y de integración numérica y en los dos siguientes nos concentramos en la solución numérica de ecuaciones diferenciales. Este tema lo iniciamos con métodos de un paso para problemas de valor inicial para ecuaciones diferenciales ordinarias. Enseguida continuamos con métodos numéricos para problemas con valores en la frontera para ecuaciones diferenciales ordinarias. Más tarde, hacemos una revisión de métodos numéricos para la solución de ecuaciones diferenciales parciales de tipo parabólico. En la parte final, hacemos referencia a software de álgebra lineal numérica, con énfasis en métodos iterativos no estacionarios para la solución de sistemas de ecuaciones lineales.

Al final de los capítulos que tratan sobre métodos numéricos, incluimos una lista de ejercicios suplementarios y un examen de entrenamiento. La mayoría de los problemas en estas secciones finales están basados en problemas y ejemplos de las referencias Cheney y Kincaid (1980) [8], Golub y Ortega (1993) [12] y Johnson y Riess (1982) [19].

1.3 Repaso: Normas en espacios vectoriales

Existen nociones topológicas y geométricas que son indispensables para estudiar análisis numérico. Posiblemente la más importante sea la idea de norma, que logra generalizar a espacios abstractos los conceptos escalares de valor absoluto y módulo.

Definición 1 *Sea V un espacio vectorial no vacío, real o complejo. Una norma en V es una función $N : V \to \mathbb{R}$ que cumple las siguientes condiciones:*

a. $N(x) \geq 0$ *para todo* $x \in V$ *y* $N(x) = 0$ *si y solo si* $x = 0$.

b. $N(\alpha x) = |\alpha|\, N(x)$ *para todo* $x \in V$ *y todo escalar* α.

c. $N(x + y) \leq N(x) + N(y)$ *para todo* $x, y \in V$.

A la norma $N(x)$ generalmente se le denota $\|x\|$. Se le agrega un subíndice si hay lugar a confusión.

1.3.1 Normas de vectores

Para el espacio vectorial $\mathbb{R}^n$, las siguientes son las normas más utilizadas:

Sea $x \in \mathbb{R}^n$ con componentes x_j, $j = 1, 2, ..., n$.

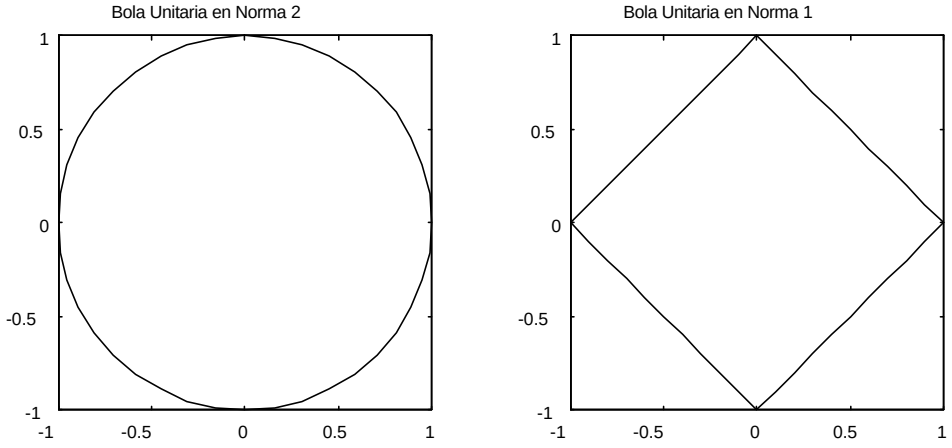
1. Norma infinito o norma uniforme: $\|x\|_\infty = \max_{j=1,2,...,n} |x_j|$.

2. Norma 1: $\|x\|_1 = \sum_{j=1}^{n} |x_j|$.

3. Norma 2 o euclideana: $\|x\|_2 = \left[\sum_{j=1}^{n} |x_j|^2 \right]^{\frac{1}{2}}$.

Las mismas definiciones son válidas en el espacio vectorial $\mathbb{C}^n$ de vectores de n componentes complejas. En este caso, $|x_j|$ significa módulo del número complejo x_j.

Ejemplo 2 *El vector* $x = [1\ 2\ 3\ 4]'$ *tiene* $\|x\|_\infty = 4$, $\|x\|_1 = 10$ *y* $\|x\|_2 = \sqrt{30}$. *Las bolas de centro en el origen y radio* 1, *en las normas* 2 *y* 1, *aparecen en la siguiente figura.*



Estas figuras fueron generadas con el siguiente M-archivo. El nombre M-archivo es debido a que los programas en lenguaje MATLAB se denominan en inglés M-files. Los nombres de archivos de programas en este lenguaje siempre tienen la letra m como extensión y son archivos de texto plano que se pueden abrir en procesadores de texto sin formato como Bloc de Notas en WINDOWS. MATLAB proporciona su propio editor de archivos en las versiones más recientes.

```
% normas.m Herramienta didactica, Carlos E. Mejia, 2002
% bolas unitarias con normas 1 y 2
t = 0:pi/20:2*pi;
subplot(121)
plot(sin(t),cos(t),'r')
axis([-1 1 -1 1]);axis square;
title('Bola Unitaria en Norma 2');
x=-1:.1:1;y=1-abs(x);
subplot(122)
plot(x,y,'r',x,-y,'r')
axis([-1 1 -1 1]);axis square
title('Bola Unitaria en Norma 1');
print -deps2 .\fig\normas.eps
```

1.3.2 Normas de matrices

Para el espacio vectorial $\mathbb{R}^{n\times n}$ de matrices cuadradas $n \times n$ con elementos reales, también se requiere el concepto de norma. La notación que se usa es la misma, pero, además de las propiedades a., b. y c. de la definición 1, se pide que se cumplan las siguientes dos condiciones:

d. $\|AB\| \leq \|A\| \, \|B\|$, para toda $A, B \in \mathbb{R}^{n\times n}$

e. Toda norma matricial debe ser *compatible* con alguna norma vectorial, es decir, $\|Ax\| \leq \|A\| \, \|x\|$, para toda $A \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n$.

Las condiciones para una norma en el espacio matricial $\mathbb{C}^{n\times n}$ son análogas a las de arriba y no las precisaremos aquí.

Un concepto necesario para la definición de una de las normas matriciales más utilizadas es el de radio espectral.

Definición 3 *Sea $B \in \mathbb{R}^{n\times n}$ con $L = \{\lambda_1, \lambda_2, ..., \lambda_r\}$ su espectro, es decir, el conjunto de sus valores propios distintos. El radio espectral de*

B es $\rho(B) = \max_{\lambda_j \in L} |\lambda_j|$.

Ejemplo 4 *La matriz*

$$A = \begin{bmatrix} -3 & 4 \\ 0 & -6 \end{bmatrix}$$

tiene radio espectral $\rho(A) = 6$, *pues sus valores propios son* -3 *y* -6.

Las normas matriciales más utilizadas son las siguientes:
Sea $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ (definiciones análogas para $A \in \mathbb{C}^{n\times n}$.)

1. Norma infinito: $\|A\|_\infty = \max_{i=1,2,\ldots,n} \sum_{j=1}^{n} |a_{ij}|$. (Compatible con norma infinito vectorial).

2. Norma 1: $\|A\|_1 = \max_{j=1,2,\ldots,n} \sum_{i=1}^{n} |a_{ij}|$. (Compatible con norma 1 vectorial).

3. Norma 2: $\|A\|_2 = \left[\rho\left(A^T A\right)\right]^{\frac{1}{2}}$. (Compatible con norma 2 vectorial).

4. Norma de Frobenius: $\|A\|_F = \left[\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2\right]^{\frac{1}{2}}$. (Compatible con norma 2 vectorial).

Ejemplo 5 *La matriz*

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \end{bmatrix}$$

tiene $\|A\|_\infty = 18$, $\|A\|_1 = 18$, $\|A\|_2 = \rho(A) = 13.4833$ *y* $\|A\|_F = \sqrt{184}$.

En este ejemplo la norma 2 de A es igual a su radio espectral por ser A una matriz simétrica. Esto no es cierto en general, como puede verse con matrices sencillas como

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

para la cual $\|A\|_2 = 1$ pero $\rho(A) = 0$.

2

Primeros pasos en MATLAB

La línea de comandos
Los M-archivos
Procesamiento de información y visualización

Según la empresa fabricante de MATLAB, llamada *The Mathworks,* MATLAB es un lenguaje de alto rendimiento para la computación técnica. Es que MATLAB posee herramientas de calidad que facilitan el trabajo en diferentes fases de la computación científica, como procesamiento de información, modelamiento, desarrollo de algoritmos, computación y simulación. Basta dar un vistazo a su página

$$http://www.mathworks.com/$$

para confirmarlo.

En esta corta introducción, ayudamos a la familiarización con este software de los que aún no lo están. Lo hacemos con unos cuantos ejemplos sencillos que sugieren la amplia gama de posibilidades que tiene este software y que preparan el camino para leer sin dificultad las rutinas de los capítulos posteriores. En todo el documento se trabajó con MATLAB 5.3, Release 11. Versiones posteriores podrían requerir pequeños cambios a las instrucciones dadas aquí para operación del software.

En Internet hay un gran número de guías para estudiar MATLAB y métodos numéricos con MATLAB. La amplia documentación en línea que trae el software es, por supuesto, una referencia obligada. Otra referencia útil y de aparición reciente es Higham y Higham (2000) [16]. Pero MATLAB es un software comercial. Aquellos que no tengan acceso a él, de todas maneras pueden seguir en gran medida las rutinas de estas notas, utilizando un paquete de software no comercial con sintaxis similar a la de MATLAB. Un tal paquete es OCTAVE, que se puede conseguir en su dirección

$$http://www.octave.org/.$$

2.1 La línea de comandos

La línea de la pantalla donde MATLAB espera comandos, empieza por el símbolo >>. A la línea se le llama *línea de comandos.* Para finalizar trabajo con MATLAB, se puede escoger la opción `Exit MATLAB` del menú `File` o entrar la palabra **quit** en la línea de comandos. Si lo que desea es abortar los cálculos que está haciendo, presione CTRL-C. La tecla de la flecha ↑ sirve para recordar las expresiones que se han escrito antes en la línea de comandos.

Hay cuatro sistemas de ayuda en línea: **help, lookfor, helpwin y helpdesk.** El último es el sistema más completo de todos, está en formato html, es apto para cualquier explorador de internet y posee enlaces para gran cantidad de archivos con formato PDF. El primero, **help,** lo explicamos por medio de un ejemplo: en la línea de comandos escriba **help linspace**. La respuesta del sistema es mostrar las primeras líneas precedidas por % del M-archivo linspace.m que listamos a continuación:

```
function y = linspace(d1, d2, n)
%LINSPACE Linearly spaced vector.
% LINSPACE(x1, x2) generates a row vector of 100 linearly
% equally spaced points between x1 and x2.
%
% LINSPACE(x1, x2, N) generates N points between x1 and x2.
%
% See also LOGSPACE, :.
% Copyright (c) 1984-98 by The MathWorks, Inc.
% $Revision: 5.6 $ $Date: 1997/11/21 23:29:09 $
if nargin == 2
n = 100;
end
if n~=1
y = d1:(d2-d1)/(n-1):d2;
else
y = d2;
end
```

Si el M-archivo por el que se pide ayuda (**help)** fue creado por un usuario y tiene líneas de comentarios en la primera parte, también las

mostrará. Así que la recomendación es incluir líneas de comentario en la parte de arriba de todo M-archivo que se cree. Es una costumbre que le ahorra tiempo y esfuerzo hasta a uno mismo.

Para conocer las variables que hay en un momento dado en el espacio de trabajo, hay dos comandos: **who** y **whos**. Ensáyelos y note la diferencia. Por cierto, también es bueno notar que en MATLAB todas las variables representan matrices. Los escalares son matrices 1×1. De hecho la palabra MATLAB es una sigla que significa *Matrix Laboratory.*

Finalmente, antes de considerar M-archivos, advertimos que el núcleo básico de MATLAB es para cálculo numérico y no simbólico. Por tanto, si **a**=5 y **b**=7+**a,** entonces **b**=12 y sigue siendo 12 aunque el valor de **a** cambie. MATLAB administra el cálculo simbólico a través de una de sus *cajas de herramientas,* que debe comprarse por aparte.

2.2 Los M-archivos

La línea de comandos deja de ser práctica cuando se desea hacer un proceso con varias órdenes o comandos. Se requiere entonces crear archivos con los comandos MATLAB que se requieren para cada tarea. Se les llama *M-archivos* (M-files en inglés.) Son de texto plano (ASCII) y MATLAB proporciona un editor con una agradable codificación de colores que ayuda mucho en el proceso de creación y depuración de dichos archivos. Los M-archivos que requieren argumentos de entrada y entregan valores de salida, se llaman *funciones.* Los que no tienen ni argumentos de entrada ni valores de salida se llaman *guiones (*script en inglés*.)* El M-archivo normas.m del ejemplo 2, que consideramos en la sección 1.3, es un guión.

Los M-archivos están en directorios. Para que MATLAB los encuentre, se debe cumplir al menos una de las siguientes condiciones:

i. El directorio en el que están los archivos pertenece al *search path* de MATLAB.

ii. El directorio en el que están los archivos es el *directorio actual* de MATLAB.

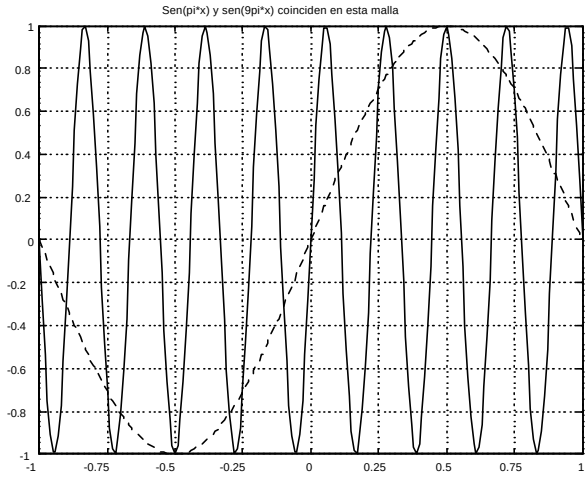
Para saber cúal es su *directorio actual,* use el comando **pwd**. Para hacer que el directorio c:\met sea su *directorio actual,* escriba la orden **cd c:\met.**

El *search path* se puede ver y actualizar por medio de la opción `Set`

`path`, del menú `File`.

Ejemplo 6 *Este es un ejemplo de M-archivo de tipo guión.*

```
% exa1.m Herramienta didactica, Carlos E. Mejia 2002
xx=-1:.01:1;
y=sin(pi*xx);z=sin(9*pi*xx);
plot(xx,y,'k--',xx,z,'r');
set(gca,'XTick',-1:.25:1)
grid on
title (' Sen(pi*x) y sen(9pi*x) coinciden en esta malla ');
print -deps2 .\fig\exa1.eps
```



De este sencillo ejemplo, destacamos que las funciones matemáticas elementales se pueden aplicar a vectores. La coincidencia de dos funciones distintas en los puntos de una malla, es un fenómeno que puede conducir a errores. En inglés se llama **aliasing**. Sobre todo lo que se quiera conocer mejor, la recomendación es acudir a la ayuda en línea.

Ejemplo 7 *El primer M-archivo de este ejemplo, es de tipo función y se ocupa de calcular, de forma ingenua, el n-ésimo término de la sucesión de Fibonacci. El segundo es de tipo guión y sirve para construir árboles binarios en los que las ramas se consiguen, parcialmente, de forma aleatoria. Utiliza la rutina rand.m que trae MATLAB.*

```
function x=fib(n);
% Herramienta didactica, Carlos E. Mejia 2002
% sucesion de Fibonacci, n<1477
% uso: x=fib(n)
x=ones(1,n);
for j=3:n
 x(j)=x(j-1)+x(j-2);
end
```

Al ejecutar x=fib(10) se obtiene

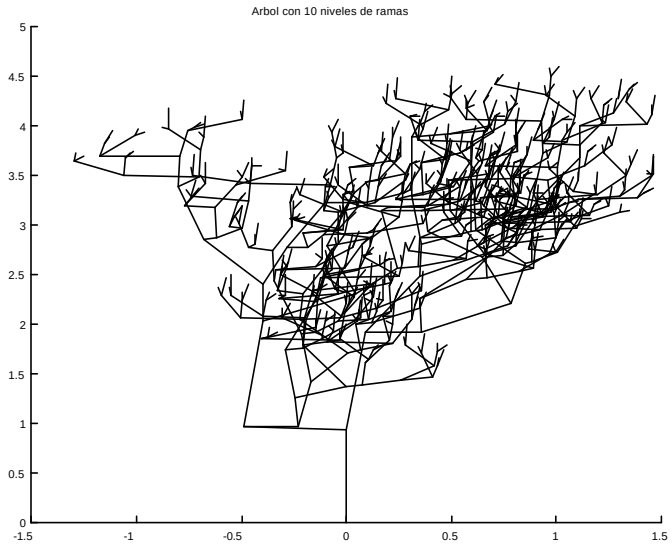
x =

1 1 2 3 5 8 13 21 34 55

Miremos ahora el generador de árboles:

```
% tree1.m, Herramienta didactica, Carlos E. Mejia, 2002
% generacion aleatoria de numeros para construir un arbol
clf
hold on
x=[0;0];y=[0;rand];line(x,y)
% a lo más 10 niveles de ramas
n=rand*10
for i=1:n,
 [m,longi]=size(x);
 angu1=(pi/2)*rand(1,longi);
 angu2=(pi/2)*rand(1,longi)+angu1;
 x=[x(2,:),x(2,:);x(2,:)+(2/i)*rand*cos(angu1),...
 x(2,:)+(3/i)*rand*cos(angu2)];
y=[y(2,:),y(2,:);y(2,:)+(2/i)*rand*sin(angu1),...
 y(2,:)+(3/i)*rand*sin(angu2)];
line(x,y);
end
title('Arbol con 10 niveles de ramas')
print -deps2 .\fig\tree1.eps
```

Uno de los árboles que pueden obtenerse con esta rutina, aparece en la siguiente figura.



Ahora es un buen momento para reposar y asimilar los comandos MATLAB que se han utilizado. También es buen momento para estudiar algunos otros por medio de la ayuda en línea y experimentación directa. Los siguientes ejercicios pueden servir de guía para estas tareas.

2.3 Ejercicios

1. Use el sistema de ayudas de MATLAB para encontrar información sobre las siguientes constantes especiales: *realmin, realmax, inf, NaN, pi, eps.*

2. Repase los M-archivos vistos hasta ahora y obtenga más información sobre los comandos utilizados en ellos, por medio de los sistemas de ayuda de MATLAB. En particular, consulte sobre **for, if, end, tres puntos (...), hold** y **line.**

3. Intentemos calcular de forma ingenua $\lim_{x\to 0} r\left(x\right)$,donde

$$r\left(x\right) = x^{-3}\left(\log\left(1-x\right) + x\exp\left(\frac{x}{2}\right)\right).$$

El resultado exacto es $-\frac{5}{24}$, se puede obtener por Regla de L'Hopital.

Ensaye el siguiente intento de aproximar este límite:

```
% exa2.m, Herramienta didactica, Carlos E. Mejia, 2002
% resultados exacto y numerico de limite complicado
format short e
x=(10.^(-(1:9)))';
% x es una sucesión decreciente de numeros positivos
res=x.^(-3).*(log(1-x)+x.*exp(x/2));
disp('Resultado numérico = ');disp(res);
```

Genera los siguientes resultados:

Resultado numérico =
-2.3341e-001
-2.1064e-001
-2.0856e-001
-2.0835e-001
-1.6282e-001
-2.8964e+001
5.2635e+004
-5.0248e+007
2.8282e+010

La discrepancia es enorme, definitivamente con este archivo no se puede aproximar ese límite. ¿Cómo explica esta discrepancia y qué estrategia recomienda para poder obtener resultados correctos?

Sugerencia: Como estrategia para hacer bien los cálculos, utilice expansiones en serie de Taylor para $\log(1-x)$ y $x\exp\left(\frac{x}{2}\right)$ alrededor de $x=0$. Con ellas, construya la serie de Taylor para $r(x)$ que resulta ser $-\frac{5}{24}+x\left(-\frac{11}{48}-\frac{379}{1920}x-...\right)$ donde el factor entre paréntesis es una serie que no requerimos en más detalle.

Como explicación de la discrepancia, piense en error de redondeo, sustracción de cantidades casi iguales y en el cociente de dos cantidades cercanas a cero. Reflexione sobre estas fuentes de error y observe que en este ejemplo aparecen juntas.

4. Use el sistema de ayuda en línea de MATLAB para estudiar sobre las siguientes formas de generar matrices: **linspace, zeros, eye** y **rand.** Practique con todos.

5. Escriba un M-archivo de tipo función que tenga como argumento de entrada, un número entero positivo n y como resultado, el promedio aritmético de las componentes de un vector $n \times 1$ generado aleatoriamente.

6. Utilice el sistema de ayuda en línea de MATLAB para estudiar sobre los comandos **num2str, clf, clear** y **disp**. Practique con ellos.

2.4 Procesamiento de información y visualización

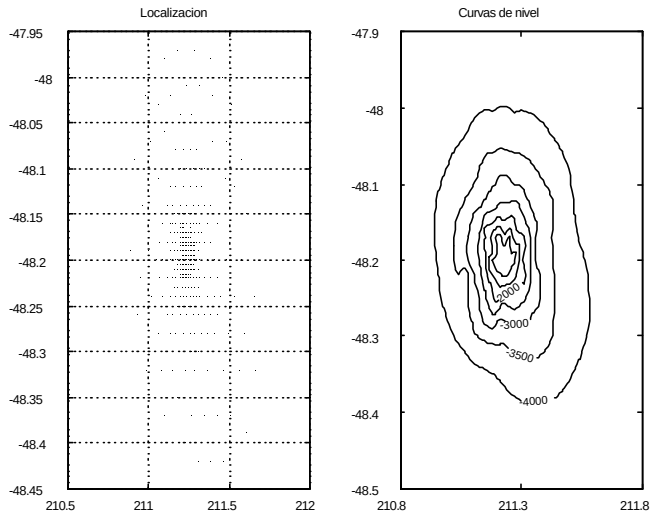
En esta sección damos un vistazo a distintas formas que tiene MATLAB de ayudar en pre y pos-procesamiento de información. Hacemos énfasis en las herramientas gráficas y en la capacidad de MATLAB de leer y almacenar información en archivos que pueden ser binarios, con extensión **mat,** o de texto plano, con la extensión que se desee.

Empezamos con el procesamiento de un archivo binario de datos llamado **seamount.mat** que trae MATLAB como ejemplo. Para ésto, preparamos el siguiente M-archivo:

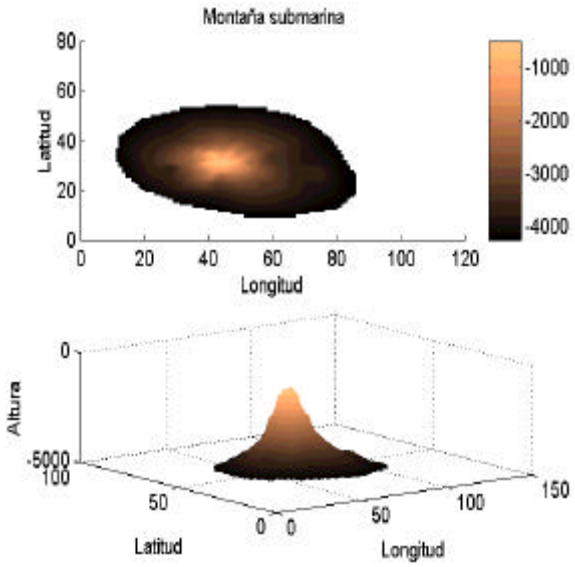
```
% exa3.m, Herramienta didactica, Carlos E. Mejia, 2002
% Referencia: ''Using MATLAB Version 5'', p. 5-21
% Datos de una montaña submarina en archivo seamount.mat
clear all;
load seamount;
subplot(1,2,1);
plot(x,y,'k.','markersize',15)
%xlabel('Longitud');ylabel('Latitud');
title(' Localizacion');
grid on
subplot(1,2,2);
[xi,yi]=meshgrid(210.8:.01:211.8,-48.5:.01:-47.9);
zi=griddata(x,y,z,xi,yi,'cubic');
[c,h]=contour(xi,yi,zi,'k-');
set(gca,'XTick',[210.8 211.3 211.8])
clabel(c,h,'manual');
title(' Curvas de nivel');
%xlabel('Longitud');ylabel('Latitud');
print -deps2 .\fig\exa3-1.eps
```

```
figure
subplot(2,1,1);
surface(zi);shading interp;
title(' Montaña submarina');
xlabel('Longitud');ylabel('Latitud');
colorbar('vert');
subplot(2,1,2);
surf(zi);
shading interp
xlabel('Longitud');ylabel('Latitud');
zlabel('Altura');
grid on
print -deps2 .\fig\exa3-2.eps
```

La primera figura, que incluye puntos de medición y curvas de nivel, es:



La segunda figura trae dos representaciones tri-dimensionales pero una de ellas se proyecta sobre un plano.

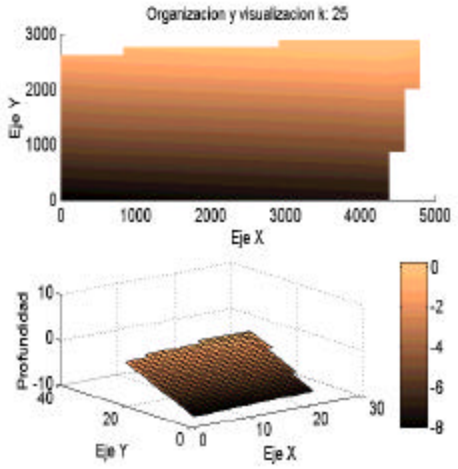


El siguiente ejemplo se basa en un archivo de datos generado con un M-archivo. Dicho archivo contiene cinco columnas x, y, z, r, v que representan coordenada x, coordenada y, profundidad, transmisividad óptica y velocidad respectivamente. Los números son todos ficticios, pero tienen cierto parecido con los que pueden medirse en un sector costero con aparatos especializados.

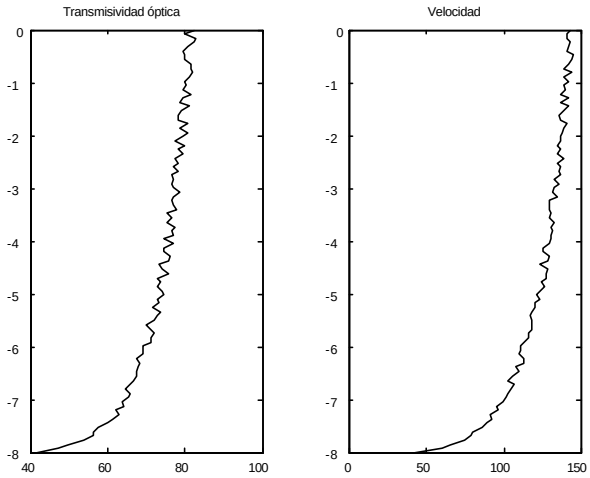
```
function exa4(k);
% Herramienta didactica, Carlos E. Mejia, 2002
% Datos generados por gen1 y grabados en mar1.dat
% columna 1: coordenada x, columna 2: coordenada y
% columna 3: profundidad (z)
% columna 4: transmisividad optica (r)
% columna 5: velocidad (v)
% se simulan mediciones de z, r, v en nodos de una
% cuadricula que cubre un rectangulo cerca de la costa
% todos los numeros en mar1.dat son ficticios
% k parametro de discretizacion
% uso exa4(k)
gen1
load mar1.dat;
```

```
% ahora tenemos una nueva variable mar1 que es una matriz

% con 5 columnas y tantas filas como filas tenga mar1.dat
x=mar1(:,1);y=mar1(:,2);z=mar1(:,3);nr=mar1(:,4);nv=mar1(:,5);

mx = min(x); Mx = max(x); my = min(y); My = max(y);
[xi,yi]=meshgrid(linspace(mx,Mx,k),linspace(my,My,k));
% se prepara malla 2-D
zi=griddata(x,y,z,xi,yi,'cubic');
% Interpolacion de datos a nueva malla
figure;
subplot(2,1,1);
surface(xi,yi,zi) % Visualizacion
title(['Organizacion y visualizacion',' k: ',num2str(k)]);

xlabel('Eje X'); ylabel('Eje Y');
shading interp;
subplot(2,1,2);
surf(zi);
xlabel('Eje X'); ylabel('Eje Y');
zlabel('Profundidad');colormap(copper)
colorbar('vert');
print -deps2 .\fig\exa4-1.eps
figure;
subplot(1,2,1)
plot(nr,z,'k');title('Transmisividad óptica');
subplot(1,2,2)
plot(nv,z,'k');title('Velocidad');
print -deps2 .\fig\exa4-2.eps
```

La primera figura trae representaciones tri-dimensionales del "área de estudio".



La segunda figura muestra los perfiles de transmisividad óptica y velocidad con respecto a la profundidad.



Para que el ejemplo quede completo, incluímos también el M-archivo **gen1.m.** En ninguna forma pretendemos que esta forma de generar datos ficticios sea la mejor. Aquí presta un buen servicio didáctico y promueve la consulta de la ayuda en línea para aclarar definiciones y usos de comandos que se incluyen ahora por primera vez.

```
% gen1.m Herramienta didactica, Carlos E. Mejia, 2002
```

```
% genera datos ficticios para procesamiento de informacion
% se piensa en sector rectangular cercano a la costa
% x: coordenada x, y: coordenada y
% z: profundidad,
% v: velocidad del agua, r: transmisividad optica
fi1=fopen('mar1.dat','w');
% transmisividad optica
A=40;B=80;step=.4;
r=A+step:step:B;lr=length(r);
p=linspace(-8,0,lr); %profundidad, variable z
s=rand(size(r));
nr=2*log(r-A)+0.1;
C=nr(1);D=nr(lr);
c1=(A-B)/(C-D);c2=(B*C-A*D)/(C-D);
nr=c1*nr+c2;
nr=nr+3*round(rand)*s;
% velocidad
v0=40;vf=140;
v=linspace(v0+1,vf,lr);
nv=3*log(v-v0)+2*rand;
cv=nv(1);dv=nv(lr);
d1=(v0-vf)/(cv-dv);d2=(vf*cv-v0*dv)/(cv-dv);
nv=d1*nv+d2;
s=rand(size(r));
nv=nv+6*round(rand)*s;
x=linspace(0,5000,10);y=linspace(0,3000,10);
l10=floor(lr/10);
nx=linspace(0,5000,lr);ny=linspace(0,3000,lr);
for j=1:lr-1
    m=mod(j,l10);q=(j-m)/l10;
 % j=l10*q+m
 if mod(j,l10)==0
 nx(j)=x(j-q*l10+1);
 else
 nx(j)=x(j-q*l10);
 end
 ny(j)=q*300;
```

```
end
for i=1:lr
out=[nx(i) ny(i) p(i) nr(i) nv(i)];
fprintf(fi1,'%8.2f %8.2f %8.2f %8.2f %8.2f\n',out);
end
fclose('all');
```

2.5 Ejercicios

1. Revise la ayuda en línea de MATLAB para saber más acerca de los comandos de visualización gráfica que hemos utilizado hasta ahora. Algunos de ellos son: **figure, plot, subplot, griddata, meshgrid, shading** y **surf.**

2. Por medio de la ayuda en línea, consulte acerca de los comandos **mod, round** y **floor.**

3. De nuevo, utilice la ayuda en línea para conocer más acerca del manejo de archivos en MATLAB. En particular, consulte sobre **fopen, fclose, load**, **save** y **diary.**

4. A continuación presentamos un M-archivo para jugar Punto y Fama. Es rudimentario pues el usuario debe entrar los números en forma de vectores pero es correcto desde el punto de vista lógico. También incluímos la salida en pantalla que proporcionó en un juego específico. Se consiguió al invocar el comando **diary pyf.dat.**

```
% punto y fama pyf1.m
% Herramienta didactica, Carlos E. Mejia, 2002
res=1;
while res ==1
nro=randperm(10);
nro=rem(nro(1:4),10);
nfa=0;
% disp([' El numero oculto es: ']); disp(nro);
cont=0;
disp([' Se trata de adivinar los 4 digitos distintos ']);
disp([' de un vector en el orden correcto ']);
disp([' Se da una fama por cada digito correctamente situado
']);
```

```
  disp([' y un punto por cada digito incorrectamente situado
']);
  while nfa~=4
  x=input(' Entre un vector con cuatro digitos: ');
  fa=find(nro-x==0);
  nfa=size(fa,2);
  npu=0;
  for i=1:4
  for j=1:4
  if nro(i)==x(j) & i~=j
  npu=npu+1;
  end
  end
  end
  disp([' Famas: ', num2str(nfa),' Puntos: ',num2str(npu)]);
  cont=cont+1;
  end
  disp([ ' Buen trabajo, felicitaciones! ']);
  disp([' Lo hizo en ', num2str(cont),' intentos ']);
  res=input(' Entre 1 si desea seguir jugando y...
   otro numero en caso contrario: ');
  end
```

```
  pyf1
  Se trata de adivinar los 4 digitos distintos
  de un vector en el orden correcto
  Se da una fama por cada digito correctamente situado
  y un punto por cada digito incorrectamente situado
  Entre un vector con cuatro digitos: [0 1 2 3]
  Famas: 1 Puntos: 1
  Entre un vector con cuatro digitos: [4 5 6 7]
  Famas: 1 Puntos: 1
  Entre un vector con cuatro digitos: [4 5 2 0]
  Famas: 0 Puntos: 2
  Entre un vector con cuatro digitos: [5 0 3 7]
  Famas: 0 Puntos: 0
  Entre un vector con cuatro digitos: [2 1 6 4]
  Famas: 4 Puntos: 0
```

```
Buen trabajo, felicitaciones!
Lo hizo en 5 intentos
Entre 1 para jugar mas y otro numero en caso contrario: 4
```

El ejercicio consiste en comprender el programa proporcionado y en tratar de mejorarlo para que su operación sea más sencilla para el usuario. En particular, MATLAB tiene la posibilidad de utilizar menús, botones, ventanas, etc. Nosotros no exploramos esas posibilidades en estas notas.

3

Ecuaciones no lineales

Método de bisección
Método de Newton
Iteración de Punto Fijo

Las ecuaciones no lineales de la forma $f(x) = 0$ aparecen frecuentemente en la computación científica y la mayoría exigen tratamiento numérico. El método más importante para su solución es el de Newton, que puede enunciarse para ecuaciones escalares o vectoriales. Para las primeras, lo estudiamos en la sección 3.2 y para las segundas lo presentamos en la sección 6.15 y lo utilizamos varias veces en estas notas.

Además del método de Newton, comentamos también un poco sobre el método de bisección, que aunque lento, tiene asegurada su convergencia en intervalos cerrados que contienen zeros de la función f. Además, también presentamos las iteraciones de punto fijo, de las cuales el método de Newton es un importante caso particular.

Sobre ecuaciones no lineales hay capítulos en los principales libros de análisis numérico. En estas notas seguimos a Stewart, 1996 [29], Atkinson, 1978 [4] y Kincaid y Cheney, 1994 [21]. Sin embargo, creemos que estas notas se pueden utilizar junto con cualquier otro libro de texto de análisis numérico.

3.1 Método de bisección

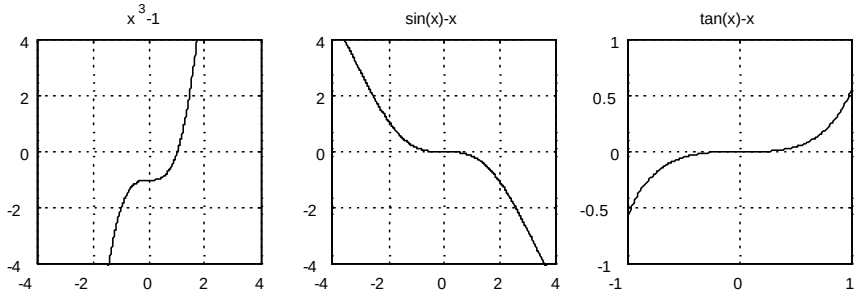
Uno de los problemas más sencillos de enunciar que más motivan el estudio de los métodos numéricos, es el de encontrar los números reales x que satisfagan una ecuación de la forma

$$f(x) = 0.$$

Por ejemplo, resolver en los números reales las ecuaciones

$$x^3 - 1 = 0, \quad sen(x) - x = 0 \quad \text{ó} \ \tan(x) - x = 0.$$

Las gráficas siguientes ilustran mejor la situación. Indican que un problema con enunciado tan simple puede ser difícil de resolver:



La primera gráfica sugiere que hay un único cero, precisamente donde ese cero está y las otras dos indican que hay alguno en cada caso, pero nada más. En realidad, $sen(x)$ y x se encuentran en un único punto, $x = 0$, pero en cambio $\tan(x)$ y x se encuentran en un número infinito de puntos. Ciertamente se requiere disponer de herramientas teóricas (teoremas) y prácticas (algoritmos) para resolver problemas de esta clase.

Una de las herramientas teóricas más importantes, que sirve de base teórica al *método de bisección*, es el *Teorema del Valor Intermedio,* que enunciamos enseguida.

Teorema 8 *(Valor Intermedio) Sea f una función continua definida en un intervalo cerrado $[a, b]$ y sean*

$$m = \min_{x \in [a,b]} f(x) \ \ y \ M = \max_{x \in [a,b]} f(x).$$

Entonces, para cada $s \in [m, M]$, existe al menos un $c \in [a, b]$ tal que $f(c) = s$.

El siguiente código en lenguaje de MATLAB sirve para resolver numéricamente, por el método de bisección, ecuaciones escalares de la forma $f(x) = 0$. En la iteración basada en la instrucción `while` de

esta rutina, puede apreciarse la idea central del método. Más detalles pueden encontrarse en un libro de texto.

```
function m1(fun,a,b)
% Herramienta didactica, Carlos E. Mejia, 2002
% Uso m1('fun',a,b)
% Se encuentra un cero de la funcion fun en el intervalo [a,b]
% por el Metodo de Biseccion
fprintf('Método de Bisección \n\n');
epsi=input('Valor para epsilon: ');
Nuit=input('valor para número maximo de iteraciones: ');
k=1;
fa=feval(fun,a); fb=feval(fun,b);
if sign(fa)==sign(fb) % chequeo inicial
disp('No se cumple condición f(a)*f(b)<0');
break;
end
% empieza búsqueda binaria
while (abs(b-a)>epsi & k < Nuit)
    c = (a+b)/2;
    if (c==a | c==b)
        fprintf('Epsilon es demasiado pequeño. ');
        break;
    end
    fc = feval(fun,c);
    if (fc==0)
        a = c; b = c;
        fa =fc; fb = fc;
        break;
    end
    if (sign(fc)~= sign(fb))
        a = c; fa = fc;
    else
        b = c; fb = fc;
    end
    k = k + 1;
end
disp(['La raiz encontrada es ',num2str(c)]);
```

```
disp(['Se requirieron ',num2str(k),' iteraciones']);
```

Un archivo con las funciones mencionadas arriba y una más con la que se trabajará más adelante, tiene la siguiente forma:

```
function y = f1(x)
y = x^3-1;
% y = sin(x)-x;
% y = tan(x)-x;
% y = cos(x)*cosh(x) +1;
% y = x*exp(x)-exp(x)+1;
% y = x^2-2;
```

Note el símbolo de comentario % delante de las definiciones que no se utilizan en un determinado momento.

Para usar m1.m con este archivo, se invoca el siguiente comando en la línea de comandos de MATLAB: `m1('f1', a, b)`, donde a y b son los extremos de intervalo y deben ser números reales.

Todo ésto parece sencillo pero se requiere tener mucho cuidado. No siempre es fácil determinar un intervalo en el que la función f es continua y tiene un cero. Además, cuando f tiene varios ceros en el mismo intervalo o cuando tiene uno que es una raiz múltiple de f, también se presentan dificultades.

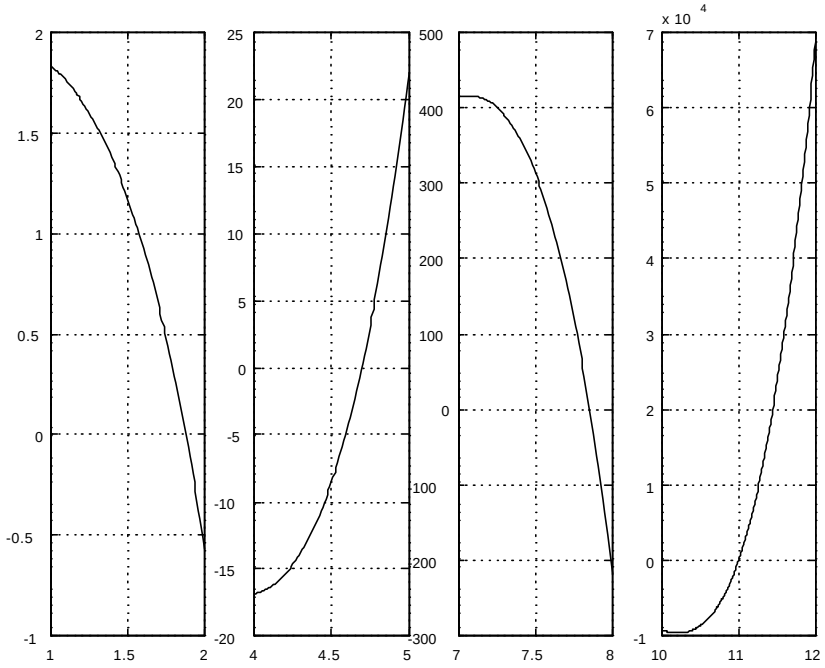
Veamos otro ejemplo:

Encontrar las primeras cuatro raíces positivas de

$$f(x) = \cos(x)\cosh(x) + 1.$$

La siguiente gráfica ayuda a encontrar intervalos apropiados para lla-

mar la rutina de bisección:



Las cuatro raíces pedidas son, aproximadamente, 1.8751, 4.6941, 7.8548 y 10.9955. Posiblemente ya notó que esta función oscila y su amplitud crece mucho. Funciones como ésta no delatan su comportamiento cuando se grafican en intervalos extensos. Utilice un computador para convencerse.

3.2 Método de Newton

En esta segunda sección sobre ecuaciones no lineales de la forma $f(x) = 0$, comentamos acerca del *Método de Newton,* su convergencia cuadrática (en muchos casos, no en todos desafortunadamente) y su derivación del *Teorema de Taylor.* El método en efecto, se debe a Newton (1642-1727), a quien se debe también, junto con Leibniz (1646-1716), la invención del cálculo.

Según el Teorema de Taylor, para f dos veces continuamente dife-

renciable en un intervalo que contiene a x y x_0,

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(\mu)(x - x_0)^2,$$

donde μ está entre x y x_0.

Supongamos que $\bar{x}$ es tal que $f(\bar{x}) = 0$. Si x_0 está cerca de $\bar{x}$ y $|f''(x_0)|$ no es demasiado grande, entonces la función

$$\bar{f}(x) = f(x_0) + f'(x_0)(x - x_0)$$

es una buena aproximación de $f(x)$ en una vecindad de $\bar{x}$. A $\bar{f}(x)$ se le llama *linealización* de $f(x)$ en x_0, pues es la ecuación de la recta tangente a la curva $y = f(x)$ en el punto $(x_0, f(x_0))$.

Finalmente, sería bueno que $\bar{x}$ se pudiera aproximar por medio de la solución de $\bar{f}(x) = 0$, que es muy fácil de obtener, siempre que $f'(x_0)$ sea no nula. Dicha solución es simplemente,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Afortunadamente, este razonamiento conduce a un efectivo método iterativo para obtener soluciones de la ecuación $f(x) = 0$. El método se llama de Newton o de Newton - Raphson y se define asi:

$$\begin{aligned} x_0 &: \text{primera aproximación} \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)},\ n = 0, 1, ... \end{aligned}$$

El siguiente M-archivo sirve para resolver ecuaciones de la forma $f(x) = 0$ por el método de Newton.

```
function out = m2(fun,fprima,x0)
% Herramienta didactica, Carlos E. Mejia, 2002
% Uso z = m2('fun','fprima',x0);
% Se aproxima un cero de f por el Metodo de Newton
% Las funciones f y f' estan dadas por los M-archivos fun
y
% fprima respectivamente. x0 es la primera aproximacion
fprintf('Método de Newton \n\n');
```

```
epsi=input('Valor para epsilon: ');
Nuit=input('Valor para número maximo de iteraciones: ');
v=feval(fun,x0);vp=feval(fprima,x0);
disp(' k x f(x) ');
out=[0,x0,v];
fprintf('%4.0f %8.3e %8.3e\n',out);
for k=1:Nuit
x1=x0-v/vp;
v=feval(fun,x1);vp=feval(fprima,x1);
out=[k,x1,v];
    fprintf('%4.0f %8.3e %8.3e\n',out);
if (abs(x1-x0)<epsi | abs(v)<epsi)
break
end
x0=x1;
end
```

Para ejecutar este archivo con cada una de las funciones definidas en el M-archivo f1.m, se requiere de otro M-archivo f2.m con las derivadas. El archivo f1.m que proponemos como ejemplo, lo listamos arriba en la sección 3.1. Nótese que utilizamos el signo de comentario % para inhabilitar las funciones que no se requieren en cada experimento.

El archivo de derivadas correspondiente a f1.m es

```
function y = f2(x)
% Derivadas de funciones en M-archivo f1.m
y = 3*x^2;
% y = cos(x)-1;
% y = sec(x)^2-1;
% y = cos(x)*sinh(x) - sin(x)*cosh(x);
% y = x*exp(x)+exp(x)-exp(x);
% y = 2*x;
```

Supongamos que queremos aproximar uno de los ceros de la función $f(x) = \cos(x)\cosh(x) + 1$. Entonces ponemos los signos de comentario % donde corresponden en f1.m y f2.m e invocamos z = m2('f1','f2',8); en la línea de comandos de MATLAB. Si proponemos una tolerancia epsi = 1.e-8 y un número máximo de iteraciones = 30, obtenemos:

k	x_k	$f(x_k)$
0	8.00000000e+000	-2.15864768e+002
1	7.87238132e+000	-2.31372508e+001
2	7.85506067e+000	-3.91280475e-001
3	7.85475753e+000	-1.18472167e-004
4	7.85475744e+000	-1.03985709e-011

3.2.1 Ejercicios

Trate de practicar con estas rutinas o con otras parecidas para resolver los siguientes ejercicios:

1. Resuelva el problema anterior con $x_0 = 7$. Obtiene lo que espera? Es correcto el resultado que obtiene?
2. Busque ceros de la función $f(x) = \tan(x) - x$ que sean cercanos a 100.

Respuesta: Los más cercanos son, aproximadamente, 98.9500628 y 102.091966. Trate de no usar esta información para que experimente en condiciones normales.

3.2.2 Análisis de Convergencia Local

Si la primera aproximación x_0 es suficientemente cercana al cero $\bar{x}$ de f y si $f'(\bar{x}) \neq 0$, entonces el método de Newton converge. La primera condición es la que nos obliga a decir que el análisis que presentamos es de caracter local. La segunda condición es válida únicamente para raíces simples de f, es decir, para raíces cuya multiplicidad algebraica es 1.

Para simplificar el análisis de convergencia, suponemos que f tiene derivadas de todos los órdenes. Sea

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Entonces

$$x_{k+1} = g(x_k). \tag{3.1}$$

La función g es la *función de iteración* del método de Newton. Cumple

$$\bar{x} = g(\bar{x}). \tag{3.2}$$

Por tal motivo, se dice que $\bar{x}$ es un punto fijo de g y que el método de Newton es un *método de punto fijo*. Posteriormente diremos más sobre iteraciones de punto fijo.

Definimos

$$e_k = x_k - \bar{x}.$$

Este es el error cometido al aproximar al cero exacto con el iterado x_k. La convergencia se da cuando

$$\lim_{k\to\infty} e_k = 0.$$

Además, en algunos casos, como el que nos ocupa, por ejemplo, se puede decir *qué tan rápida* es dicha convergencia.

De (3.1) y (3.2),

$$e_{k+1} = x_{k+1} - \bar{x} = g(x_k) - g(\bar{x})$$

y por Teorema de Taylor,

$$g(x_k) - g(\bar{x}) = g'(\beta_k)(x_k - \bar{x}),$$

con β_k un número entre x_k y $\bar{x}$. Es decir,

$$e_{k+1} = g'(\beta_k)\, e_k. \tag{3.3}$$

De otro lado,

$$g'(x) = \frac{f(x)\, f''(x)}{f'(x)^2}$$

pero como $f(\bar{x}) = 0$ y $f'(\bar{x}) \neq 0$, entonces $g'(\bar{x}) = 0$. Por continuidad de g' en $\bar{x}$, para todo $0 < C < 1$, existe $\delta > 0$ tal que si $0 < |x - \bar{x}| < \delta$ entonces $|g'(x) - g'(\bar{x})| = |g'(x)| \leq C < 1$. Por tanto, debemos escoger x_0 que cumpla $|x_0 - \bar{x}| < \delta$. Como β_0 está entre x_0 y $\bar{x}$, entonces

$$|e_1| = |g'(\beta_0)|\,|e_0| \leq C\,|e_0| < C\delta < \delta.$$

De aqui, $|x_1 - \bar{x}| < \delta$, lo que indica que este argumento se puede repetir. Llegamos a

$$|e_2| = |g'(\beta_1)|\,|e_1| \leq C\,|e_1| < C^2\,|e_0| < C^2\delta < \delta.$$

Por inducción,

$$|e_{k+1}| = |g'(\beta_k)|\,|e_k| \leq C\,|e_k| < C^{k+1}\,|e_0| < C^{k+1}\delta < \delta.$$

Esto prueba que los iterados pertenecen todos al intervalo con radio δ en torno a la raíz y que $\lim_{k\to\infty} |e_k| = 0$.

Para saber qué tan rápida es esta convergencia, usamos Teorema de Taylor para g, teniendo en cuenta que $g'(\bar{x}) = 0$. Es decir,

$$g(x_k) - g(\bar{x}) = \frac{1}{2} g''(\eta_k)\,(x_k - \bar{x})^2\,,$$

donde η_k está entre x_k y $\bar{x}$. Esto lo reescribimos asi:

$$e_{k+1} = \frac{1}{2} g''(\eta_k)\,e_k^2.$$

De aqui encontramos que

$$\lim_{k\to\infty} \frac{e_{k+1}}{e_k^2} = \frac{1}{2} g''(\bar{x}) = \frac{1}{2}\frac{f''(\bar{x})}{f'(\bar{x})}.$$

Un proceso iterativo en el que la sucesión de errores cumple

$$\lim_{k\to\infty} \frac{e_{k+1}}{e_k^2} = \text{constante},$$

se dice que es *cuadráticamente convergente.*

3.3 Iteraciones de punto fijo

En esta tercera parte sobre ecuaciones no lineales de la forma $f(x) = 0$, exponemos algunas ideas y proponemos algunos ejemplos sobre *iteraciones de punto fijo.* Tales iteraciones fueron brevemente introducidas a través del *Método de Newton,* que es un método de punto fijo. Recordemos que si

$$g(x) = x - \frac{f(x)}{f'(x)},$$

entonces

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

es la presentación como iteración de punto fijo del método de Newton.

Muchas veces, un problema de la forma $f(x) = 0$ se puede enunciar como un problema de punto fijo de la forma $x = g(x)$. Con un ejemplo sencillo se comprende mejor.

Ejemplo 9 *Se quieren encontrar los ceros de los polinomios* $p(x) = x^2 - 5x + 6$ *y* $q(x) = x^3 - 13x + 18$ *resolviendo un problema de punto fijo asociado.*

Para el polinomio p, se proponen las siguientes:

$$\begin{aligned} x &= g_1(x) = x^2 - 4x + 6 \\ x &= g_2(x) = 5 - \tfrac{6}{x} \\ x &= g_3(x) = (5x - 6)^{\frac{1}{2}} \\ x &= g_4(x) = \tfrac{x^2+6}{5}. \end{aligned}$$

Note que $x = g_j(x)$ si y solo si $p(x) = 0$.

El siguiente M-archivo ofrece la posibilidad de ensayar estas cuatro funciones de iteración.

```
function out = m3(f_fun,g_fun,s,x0)
% Herramienta didactica, Carlos E. Mejia, 2002
% Uso z = m3('f_fun','g_fun',s,x0);
% Se aproxima el cero s de f_fun por medio de una
% iteracion de punto fijo con funcion de iteracion g_fun
% x0 es la primera aproximacion
% El cero s es conocido pero no se utiliza en el proceso
% iterativo. Solo se usa para listar las dos ultimas
% columnas de la tabla de resultados
fprintf('Método de Punto Fijo \n\n');
epsi=input('Valor para epsilon: ');
Nuit=input('Valor para número maximo de iteraciones: ');
v=feval(f_fun,x0);e0=s-x0;
disp(' k x_k f(x_k) e_k e_k)/e_(k-1))');
out=[0,x0,v,e0];
fprintf('%4.0f %8.3e %8.3e %8.3e\n',out);
for k=1:Nuit
x1=feval(g_fun,x0);
```

```
v=feval(f_fun,x1);
e1=s-x1;
out=[k,x1,v,e1,e1/e0];
    fprintf('%4.0f %8.3e %8.3e %8.3e %8.3e\n',out);
if (abs(x1-x0)<epsi | abs(v)<epsi)
break
end
x0=x1;e0=e1;
end
```

Las funciones f y g_j están dadas en M-archivos como los siguientes:

```
function z = f(x);
% Funcion f_fun iteracion de punto fijo
% z=x^2-5*x+6;
z=x^3-13*x+18;
```

```
function z = g(x);
% funcion g_fun de iteracion de punto fijo
% z = x^2-4*x+6;
% z = 5-6/x;
% z = (5*x-6)^(1/2);
% z = (x^2+6)/5;
%
% z = x^3-12*x+18;
% z = (x^3+18)/13;
% z = (13*x-18)^(1/3);
z = (13*x-18)/x^2;
```

Para aproximar el cero $s = 2$ del primer polinomio, utilizamos el M-archivo propuesto. Siempre utilizamos una tolerancia $epsi = 10^{-8}$.

Para la función g_1 con un máximo de 10 iteraciones, los resultados son:

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}
0	0.00e+000	6.00e+000	2.00e+000	
1	6.00e+000	1.20e+001	-4.00e+000	-2.00e+000
2	1.80e+001	2.40e+002	-1.60e+001	4.00e+000
3	2.58e+002	6.53e+004	-2.56e+002	1.60e+001
4	6.55e+004	4.29e+009	-6.55e+004	2.56e+002
5	4.29e+009	1.84e+019	-4.29e+009	6.55e+004
6	1.84e+019	3.40e+038	-1.84e+019	4.29e+009
7	3.40e+038	1.16e+077	-3.40e+038	1.84e+019
8	1.16e+077	1.34e+154	-1.16e+077	3.40e+038
9	1.34e+154	Inf	-1.34e+154	1.16e+077
10	Inf	NaN	-Inf	Inf

Ciertamente no hay convergencia. Para la función g_2 con un máximo de 20 iteraciones, se encuentra:

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}
0	0.00e+000	6.00e+000	2.00e+000	
1	-Inf	Inf	Inf	Inf
2	5.00e+000	6.00e+000	-3.00e+000	0.00e+000
.	...	...	...	...
10	3.03e+000	2.74e-002	-1.03e+000	9.87e-001
.	...	...	...	...
15	3.00e+000	3.45e-003	-1.00e+000	9.98e-001
.	...	...	...	...
20	3.00e+000	4.52e-004	-1.00e+000	1.00e+000

Es decir, hay convergencia, pero a otro cero de p. Otro ensayo con g_2 y un máximo de 10 iteraciones:

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}
0	1.99e+000	1.01e-002	1.00e-002	
4	1.95e+000	5.56e-002	5.28e-002	1.53e+000
8	1.66e+000	4.56e-001	3.40e-001	1.67e+000
10	6.69e-001	3.10e+000	1.33e+000	2.17e+000

Definitivamente, aproximar el cero $s = 2$ por este medio no parece fácil. Tratemos con g_3 y un máximo de 10 iteraciones:

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}
0	0.00e+000	6.00e+000	2.00e+000	
4	3.58e+000	-2.11e+000	-1.58e+000	8.17e-001
8	3.43e+000	4.21e-001	-1.43e+000	9.16e-001
10	3.29e+000	3.13e-001	-1.29e+000	9.43e-001

Tampoco se ve promisorio. Finalmente, con g_4 y un máximo de 30 iteraciones, se obtiene:

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}
0	0.00e+000	6.00e+000	2.00e+000	
5	1.81e+000	2.33e-001	1.95e-001	7.48e-001
10	1.95e+000	5.75e-002	5.45e-002	7.86e-001
20	1.99e+000	5.55e-003	5.52e-003	7.99e-001
25	2.00e+000	1.80e-003	1.80e-003	8.00e-001
30	2.00e+000	5.89e-004	5.89e-004	8.00e-001

Aqui se observa convergencia.

Este ejemplo simple nos invita a reflexionar acerca de condiciones suficientes para convergencia de una iteración de punto fijo y también acerca del cociente de errores consecutivos presentado en la última columna, que nos sugiere tendencia a ser constante en caso de haber convergencia. Todo ésto lo aclararemos próximamente, pero antes proponemos el siguiente ejercicio:

3.4 Ejercicio

Para el polinomio $q(x) = x^3 - 13x + 18$ presentado antes, considere las siguientes funciones de iteración de punto fijo y decida cuáles permiten

aproximar el cero $s = 2$. Realice 5 iteraciones de punto fijo con cada una de las funciones admisibles.

$$\begin{aligned} x &= g_1(x) = x^3 - 12x + 18 \\ x &= g_2(x) = \tfrac{x^3+18}{13} \\ x &= g_3(x) = (13x - 18)^{\frac{1}{3}} \\ x &= g_4(x) = \tfrac{13x-18}{x^2}. \end{aligned}$$

3.5 Análisis de convergencia local

En esta sección sobre ecuaciones no lineales de la forma $f(x) = 0$, exponemos teoría de convergencia local para *iteraciones de punto fijo*. Recordemos que los ejemplos presentados antes, nos invitan a reflexionar acerca de condiciones suficientes para convergencia de una iteración de punto fijo y también acerca del cociente de errores consecutivos presentado en la última columna, que nos sugiere tendencia a ser constante en caso de haber convergencia lineal.

En lugar de empezar con una ecuación de la forma $f(x) = 0$, lo hacemos con una función g que tiene un punto fijo s. Es decir, tal que $s = g(s)$. Nos preguntamos cuándo y cómo converge la iteración

$$\begin{aligned} &x_0, \text{ primera aproximación} \\ &x_{k+1} = g(x_k), \ k = 0, 1, ... \end{aligned} \tag{3.4}$$

Para nadie es sorpresa que es la derivada de g en el punto fijo la que determina si hay o no convergencia. Para hacer el análisis más sencillo, suponemos que g tiene un número *suficiente* de derivadas continuas. El resultado central es el siguiente:

Teorema 10 *(Teorema de Convergencia Local) Si* $|g'(s)| < 1$, *entonces hay un intervalo* $I_\delta = [s - \delta, s + \delta]$ *tal que la iteración (3.4) converge a* s *siempre que* $x_0 \in I_\delta$. *Si* $g'(s) \neq 0$, *entonces la convergencia es lineal con rata de convergencia* $g'(s)$. *Si*

$$0 = g'(s) = g''(s) = ... = g^{(p-1)}(s) \neq g^{(p)}(s), \tag{3.5}$$

entonces la convergencia es de orden p.

dem:

No es sorprendente que el argumento presentado para demostrar la convergencia del Método de Newton, que es un método de Punto Fijo, nos sea útil de nuevo aquí. Lo repetimos por considerarlo de interés. Definimos $e_k = x_k - s$. Por Teorema de Taylor,

$$x_{k+1} - s = g(x_k) - g(s) = g'(\mu_k)(x_k - s),$$

donde μ_k está entre x_k y s. Por tanto,

$$|e_{k+1}| = |g'(\mu_k)|\,|e_k|. \tag{3.6}$$

Sea C una constante positiva tal que $|g'(s)| < C < 1$. Por continuidad de g', para $\varepsilon = C - |g'(s)|$, existe δ tal que $|g'(x) - g'(s)| \leq \varepsilon$ siempre que $0 < |x - s| < \delta$. De aqui obtenemos $|g'(x)| \leq \varepsilon + |g'(s)| = C < 1$ para todo $x \in I_\delta$.

Sea $I_\delta = [s - \delta, s + \delta]$. Veamos que si $x_0 \in I_\delta$, la iteración de punto fijo $x_{k+1} = g(x_k)$ es convergente. Por (3.6),

$$|e_1| \leq |g'(\mu_0)|\,|e_0|.$$

Como μ_0 está entre x_0 y s, entonces está en I_δ. Luego

$$|e_1| \leq C\,|e_0| < C\delta < \delta. \tag{3.7}$$

Esto indica que $x_1 \in I_\delta$.

Similarmente,

$$|e_2| \leq |g'(\mu_1)|\,|e_1|,$$

con μ_1 entre x_1 y s, o sea en I_δ. Luego de (3.7),

$$|e_2| \leq C\,|e_1| \leq C^2\,|e_0| < C^2\delta < \delta.$$

Esto indica que $x_2 \in I_\delta$. Con este procedimiento, llegamos a que

$$|e_k| \leq C^k\,|e_0|$$

y por tanto

$$0 \leq \lim_{k\to\infty} |e_k| \leq \lim_{k\to\infty} C^k\,|e_0| = 0,$$

o sea, la iteración es convergente. Más aun, si $g'(s) \neq 0$, hay una vecindad de s en la que g' no se anula. Sin pérdida de generalidad,

podemos suponer que tal vecindad es el intervalo I_δ con el que hemos venido trabajando. Entonces podemos escribir

$$0 \leq \lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|} \leq \lim_{k\to\infty} |g'(\mu_k)|,$$

para algún μ_k entre x_k y s. Por convergencia, $\lim_{k\to\infty} \mu_k = s$ y de nuevo por continuidad de g', $\lim_{k\to\infty} g'(\mu_k) = g'(s)$. Por tanto

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|} = |g'(s)|.$$

Qué pasa si existe $p > 1$ tal que

$$0 = g'(s) = g''(s) = ... = g^{(p-1)}(s) \neq g^{(p)}(s)?$$

De nuevo, invocamos el Teorema de Taylor para obtener

$$x_{k+1} - s = g(x_k) - g(s) = \frac{1}{p!} g^{(p)}(\mu_k)(x_k - s)^p,$$

con μ_k entre x_k y s. Con el procedimiento que usamos arriba, vemos que

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = \frac{1}{p!} \left|g^{(p)}(s)\right|.$$

Es decir, la convergencia es de orden p.

3.6 Ejemplos

La iteración de Newton para raíces simples tiene convergencia al menos, cuadrática.

En efecto, si s es una raíz simple de f, entonces $f(s) = 0$ pero $f'(s) \neq 0$. Sea

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Entonces,

$$g'(s) = 0 \text{ y } g^{(2)}(s) = \frac{f^{(2)}(s)}{f'(s)}.$$

De aqui se concluye que el orden de convergencia es 2 si $f^{(2)}(s) \neq 0$ y es más de 2 si $f^{(2)}(s) = 0$.

El M-archivo m4.m contiene el cálculo de errores y cocientes de errores consecutivos para la iteración de Newton. Es una modificación menor del M-archivo m2.m.

```
function out = m4(f_fun,f_prima,s,x0)
% Herramienta didactica, Carlos E. Mejia, 2002
% Uso z = m4('f_fun','f_prima',s,x0);
% Se aproxima el cero s de f_fun por medio de la
% iteracion de punto fijo con funcion de
% iteracion dada por Metodo de Newton
% x0 es la primera aproximacion
% El cero s es conocido pero no se utiliza en el proceso
% iterativo. Solo se usa para listar las dos ultimas
% columnas de la tabla de resultados
% ultima columna ilustra la convergencia cuadratica
fprintf('Método de Newton y convergencia cuadratica\n\n');
epsi=input('Valor para epsilon: ');
Nuit=input('Valor para número maximo de iteraciones: ');
v=feval(f_fun,x0);vp=feval(f_prima,x0);e0=s-x0;
disp(' k x_k f(x_k) e_k e_k/(e_(k-1)) e_k/(e_(k-1))^2');
out=[0,x0,v,e0];
fprintf('%4.0f %8.4e %8.4e %8.4e\n',out);
for k=1:Nuit
x1=x0-v/vp;
v=feval(f_fun,x1);vp=feval(f_prima,x1);
e1=s-x1;
out=[k,x1,v,e1,e1/e0,e1/e0^2];
    fprintf('%4.0f %8.4e %8.4e %8.4e %8.4e %8.4e\n',out);
if (abs(x1-x0)<epsi | abs(v)<epsi)
break
end
x0=x1;e0=e1;
end
```

Veamos su ejecución para tres ejemplos de los listados en el M-archivo f1.m. En todos los casos, usamos 1.e-6 como valor de epsilon y un

máximo de 20 iteraciones.

1. Aproximación del cero $s = \sqrt{2}$ de la función $f(x) = x^2 - 2$ por el Método de Newton.

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}	$e_k/(e_{k-1})^2$
0	50	2.5e+003	-4.9e+001		
1	25.0200	6.2e+002	-2.4e+001	4.9e-001	-1.0e-002
2	12.5500	1.6e+002	-1.1e+001	4.7e-001	-2.0e-002
3	6.3547	3.8e+001	-4.9e+000	4.4e-001	-4.0e-002
4	3.3347	9.1e+000	-1.9e+000	3.9e-001	-7.9e-002
5	1.9672	1.9e+000	-5.5e-001	2.9e-001	-1.5e-001
6	1.4919	2.3e-001	-7.8e-002	1.4e-001	-2.5e-001
7	1.4162	5.7e-003	-2.0e-003	2.6e-002	-3.4e-001
8	1.4142	4.1e-006	-1.4e-006	7.1e-004	-3.5e-001
9	1.4142	2.1e-012	-7.4e-013	5.1e-007	-3.5e-001

La última columna sugiere convergencia cuadrática. Pero solamente el análisis teórico puede asegurarla.

2. Aproximación del cero $s = 0$ de la función $f(x) = xe^x - e^x + 1$. Nótese que $s = 0$ es un cero de multiplicidad 2 y que no debemos esperar convergencia cuadrática.

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}	$e_k/(e_{k-1})^2$
0	2.0000	8.4e+000	-2.0e+000		
1	1.4323	2.8e+000	-1.4e+000	7.2e-001	-3.6e-001
2	0.9638	9.1e-001	-9.6e-001	6.7e-001	-4.7e-001
.	...	...	...	...	...
8	0.0276	3.9e-004	-2.8e-002	5.1e-001	-9.4e+000
9	0.0139	9.8e-005	-1.4e-002	5.0e-001	-1.8e+001
10	0.0070	2.5e-005	-7.0e-003	5.0e-001	-3.6e+001
11	0.0035	6.1e-006	-3.5e-003	5.0e-001	-7.2e+001
12	0.0018	1.5e-006	-1.8e-003	5.0e-001	-1.4e+002
13	0.0009	3.8e-007	-8.8e-004	5.0e-001	-2.9e+002

La penúltima columna sugiere convergencia lineal. La última no invita a pensar en convergencia cuadrática. De todas maneras, como lo dijimos antes, los experimentos nunca son la última palabra.

Estos dos ejemplos sugieren la preparación de tablas con cocientes de errores consecutivos para tratar de encontrar patrones que después conduzcan a afirmaciones matemáticas generales. Tales patrones no siempre se encuentran, como en el siguiente caso:

3. Aproximar el cero $s = 1$ de la función $x^3 - 1$.

k	x_k	$f(x_k)$	e_k	e_k/e_{k-1}	$e_k/(e_{k-1})^2$
0	3.0000	2.6e+001	-2.0e+000		
1	2.0370	7.5e+000	-1.0e+000	5.2e-001	-2.6e-001
2	1.4384	2.0e+000	-4.4e-001	4.2e-001	-4.1e-001
3	1.1200	4.1e-001	-1.2e-001	2.7e-001	-6.2e-001
4	1.0124	3.8e-002	-1.2e-002	1.0e-001	-8.6e-001
5	1.0002	4.5e-004	-1.5e-004	1.2e-002	-9.8e-001
6	1.0000	6.9e-008	-2.3e-008	1.5e-004	-1.0e+000

Ninguna de las dos últimas columnas ofrece mayor información como para buscar un orden dado de convergencia. En este caso, la convergencia es cuadrática, pues se trata de un cero simple.

3.7 Ejercicios suplementarios

1. Trate de encontrar todos los ceros de $f(x) = \cos(x) - \cos(3x)$ por un procedimiento gráfico o analítico. Enseguida utilice un método numérico para aproximar los ceros que se encuentran en el intervalo $[-2\pi, 2\pi]$.

2. Argumente por medios gráficos que la ecuación $x = \tan(x)$ tiene infinitas soluciones. Conjeture el valor de dos de esas soluciones y confirme su conjetura con un método numérico.

3. Encuentre todas las raíces de $\log\left(\dfrac{1+x}{1-x^2}\right)$.

   Sugerencia: Con una buena gráfica conjeture que hay una sola

raíz. Después, por un procedimiento analítico, demuestre que, en efecto, puede haber a lo más una. Finalmente, con un procedimiento gráfico o un método numérico, aproxime la raíz. En este caso, el siguiente teorema puede servir como el procedimiento analítico requerido:
Teorema: Sea f una función continuamente diferenciable en (a,b). Si $f'(x) > 0$ para todo $x \in (a,b)$, entonces f tiene a lo más un cero en el intervalo (a,b).
Este teorema es una consecuencia del Teorema del Valor Medio para Derivadas. Trate de demostrarlo.

4. Encuentre el punto de corte de las gráficas $y = 3x$ y $y = e^x$.

5. Utilice medios gráficos para aproximar la localización de los ceros de la función $\log(1+x) + \tan(2x)$. Utilice un método numérico para aproximar dos de esos ceros.

6. Escriba el método de Newton para determinar el recíproco de la raíz cuadrada de un número positivo. Realice dos iteraciones del método para aproximar $1/\pm\sqrt{4}$ empezando en $x_0 = 1$ y $x_0 = -1$.
Sugerencia: Enuncie el método de Newton para encontrar los ceros de $f(x) = x^2 - \dfrac{1}{4}$ y simplifíquelo.

7. Dos de los ceros de $x^4 + 2x^3 - 7x^2 + 3$ son positivos. Aproxímelos por el método de Newton.

8. La ecuación $x - Rx^{-1} = 0$ tiene a $x = \pm R^{1/2}$ como solución. Establezca el método de Newton para esta ecuación, simplifíquelo y calcule los primeros 5 iterados para $R = 25$ y $x_0 = 1$.

9. ¿Cuál es la recta $y = ax + b$ que más se aproxima a $sen(x)$ cerca de $x = \pi/4$?
¿Por qué está relacionada esta pregunta con el método de Newton?
Sugerencia: Considere la expansión en serie de Taylor de $sen(x)$ alrededor de $x = \pi/4$.

10. Utilice dos métodos numéricos diferentes para aproximar $\log(2)$.
Sugerencia: Piense en la ecuación $e^x - 2 = 0$.

11. Considere la ecuación $x - 2sen\,(x) = 0$.
    a. Muestre gráficamente que esta ecuación tiene exactamente tres raíces en el intervalo $(-2, 2)$: Son 0 y una en cada uno de los intervalos $(\pi/2, 2)$ y $(-2, -\pi/2)$.
    b. Muestre que la iteración de punto fijo $x_{n+1} = 2sen\,(x_n)$ converge a la raíz en $(\pi/2, 2)$ para cualquier x_0.
    c. Utilice el método de Newton para aproximar este mismo cero. Compare la rapidez de convergencia en las partes a. y b.

12. Considere la ecuación $x^2 - 2x + 2 = 0$. Por medio de ensayos con varios valores x_0, estudie el comportamiento del método de Newton ante esta igualdad.
    Comentario: Como $x^2 - 2x + 2 = (x-1)^2 + 1 > 0$, no hay un número real que cumpla esta igualdad.

3.8 Examen de entrenamiento

Resuelva estos ejercicios sin consultar libros ni notas de clase para darse una idea del nivel de preparación que ha obtenido hasta ahora. Una calculadora sencilla es lo mínimo que debe tener a disposición y es suficiente para poder responder todos los ejercicios del examen. Claro que es preferible si dispone de una calculadora graficadora o un computador.

1. Considere la función $f\,(x) = e^{2x} + 3x + 2$.
   a. Demuestre que f tiene una raíz en $[-2, 0]$.
   b. Demuestre que este es el único cero de f.
   Sugerencia: Utilice el teorema que se sugirió en el Ejercicio Suplementario número 3.
   c. Con $x = -1$ como primera aproximación, calcule los primeros tres iterados del método de bisección y del método de Newton para aproximar la raíz de f.

2. Sea $f\,(x) = (x-1)^2$. Muestre que la derivada de f en 1 es 0, pero los iterados del método de Newton convergen a 1. ¿Qué puede decir de la rata de convergencia?

3. Evalúe

$$s = \sqrt[3]{6 + \sqrt[3]{6 + \sqrt[3]{6 + ...}}}$$

Sugerencia: Sea $x_0 = 0$ y considere $g(x) = \sqrt[3]{6 + x}$. El número s es un punto fijo de la función g.

4. Como un caso extremo de una función para la cual el método de Newton converge lentamente, considere $f(x) = (x - a)^n$, para n un entero positivo y a un número real. Muestre que el método de Newton genera la sucesión

$$x_{n+1} = \left(1 - \frac{1}{n}\right) x_n + \frac{a}{n}$$

y que dos errores consecutivos satisfacen la relación

$$x_{n+1} - a = \left(1 - \frac{1}{n}\right)(x_n - a).$$

¿Por qué motivo esta relación sugiere que la convergencia es lenta?

4

Interpolación

Tablas
Teorema fundamental
Forma de Newton (Diferencias divididas)
Forma de Lagrange

4.1 Tablas

Uno de los procedimientos más útiles en el procesamiento de información es el de construir curvas con base en unos cuantos puntos de referencia P_j con coordenadas (x_j, y_j) dados en una tabla de la forma

j	0	1	$\cdots$	n
x_j	x_0	x_1	$\cdots$	x_n
y_j	y_0	y_1	$\cdots$	y_n

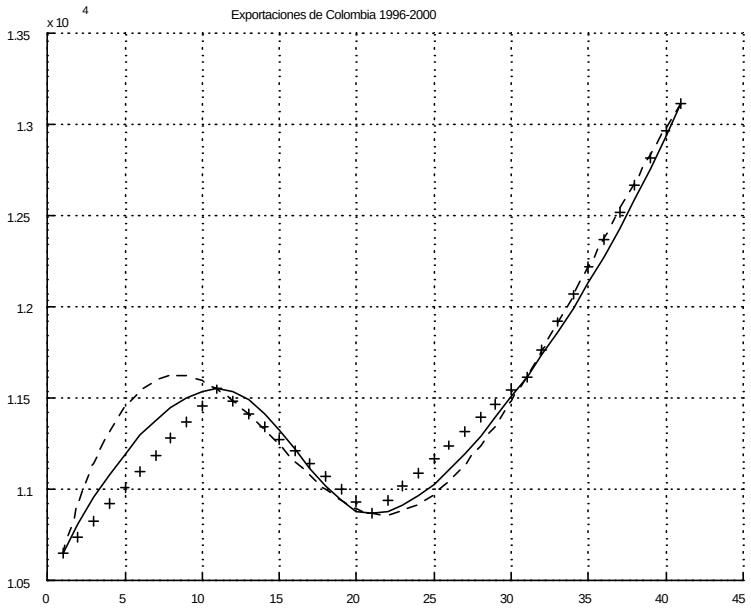
(4.1)

Las curvas obtenidas representan y aproximan la información puntual y con las debidas precauciones, se usan frecuentemente en lugar de los datos originales. Si la curva que se construye pasa por todos los puntos de la tabla, se denomina una *interpolación*. Si la curva no tiene que pasar por los puntos pero satisface un cierto *criterio de cercanía*, se denomina una *aproximación* a los datos de la tabla. Los criterios de cercanía son procedimientos de optimización entre los cuales el más común es posiblemente el de *mínimos cuadrados.*

Ejemplo 11 *El Departamento Nacional de Estadística de la República de Colombia, DANE, publica en su página <http://www.dane.gov.co>, que el total de exportaciones del país, en millones de dólares, entre 1996 y 2000, está dado por la siguiente tabla:*

Año	*1996*	*1997*	*1998*	*1999*	*2000*
Total	*10648*	*11549*	*10866*	*11617*	*13115*

Tres diferentes interpolaciones obtenidas por medio del M-archivo m5.m, están dadas en la siguiente figura:



Por su parte, el M-archivo m5.m es:

```
% Herramienta didactica, Carlos E. Mejia, 2002
% uso: m5
% tres clases de interpolacion ofrecidas por MATLAB
x=1996:2000;xp=1996:.1:2000;
y=[10648 11549 10866 11617 13115];
y1=interp1(x,y,xp,'linear');
grid on;hold on;
plot(y1,'k+');
title('Exportaciones de Colombia 1996-2000');
y2=interp1(x,y,xp,'cubic');
plot(y2,'k');
y3=interp1(x,y,xp,'spline');
plot(y3,'k--');
print -deps2 .\fig\m5.eps
```

Los tres métodos de interpolación utilizados son: *linear*, que significa que los puntos se unen por medio de rectas, *cubic*, que construye un polinomio cúbico que pasa por los puntos y *spline*, que es una interpolación cúbica segmentaria. Cada segmento es un polinomio cúbico que se une *suavemente* con el vecino, o sea que su valor y los de sus dos primeras derivadas en el punto de encuentro, concuerdan con los del vecino.

4.2 Teorema fundamental

Pasamos ahora a considerar algunas ideas teóricas sobre interpolación, encabezadas por el teorema fundamental de interpolación polinómica que garantiza existencia y unicidad del polinomio interpolante. Cualquier libro de texto de análisis numérico incluye este material. Aquí seguimos a Stewart, 1996 [29], Atkinson, 1978 [4] y Kincaid y Cheney, 1994 [21].

La idea de aproximar es central en el modelamiento matemático. Aquí queremos presentar una aproximación particular que es interpolatoria de una tabla dada, es decir, la función resultante satisface la tabla. Pero antes veamos que hay aproximaciones no interpolatorias y que en muchas ocasiones son las únicas disponibles.

Consideremos la siguiente tabla:

j	0	1	2	3
x_j	1	2	3	4
y_j	1	1/2	1/3	1/4

(4.2)

entonces podemos pretender conseguir funciones de muy diversas clases para aproximarla.

4.2.1 Intento 1

Una recta, llamada recta de regresión, que se consigue por aproximación de mínimos cuadrados.

Nótese que buscar interpolación con una recta es imposible, pues tal recta debería ser de la forma $f(x) = ax + b$ con $f(x_j) = y_j$ para

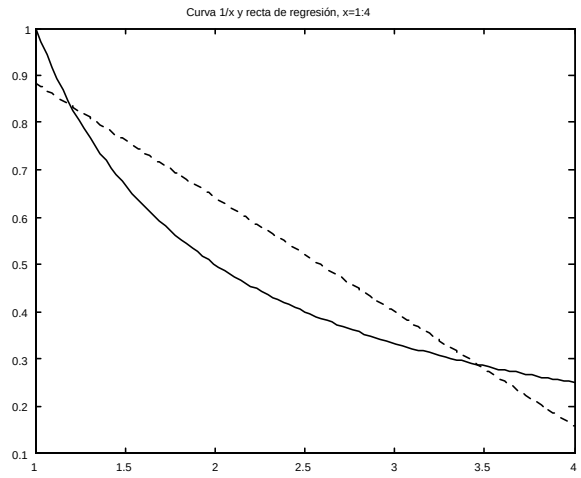
$j = 0, 1, 2, 3$. Esto lleva al sistema

$$\begin{aligned} ax_0 + b &= y_0 \\ ax_1 + b &= y_1 \\ ax_2 + b &= y_2 \\ ax_3 + b &= y_3 \end{aligned}$$

que es contradictorio.

Se busca entonces una recta $f(x) = ax+b$ tal que $\sum_{j=0}^{3} [f(x_j) - y_j]^2$ sea mínimo. Esta recta se consigue con el siguiente M-archivo, que produce la figura que se incluye a continuación.

```
% Herramienta didactica, Carlos E. Mejia, 2002
% sistema redundante
% uso: redun
A=[1 1; 2 1;3 1; 4 1];
y=[1; 1/2;1/3; 1/4];
x=A\y
p=x' % en forma de polinomio
z=1:3/100:4;
plot(z,1./z,'r',z,polyval(p,z),'k--');
title('Curva 1/x y recta de regresión, x=1:4');
print -deps2 .\fig\redun.eps
```



Digno de mencionarse es que, para MATLAB, un polinomio es un vector fila, el vector de sus coeficientes. Las operaciones entre polinomios, están definidas en MATLAB con esa convención. Por ejemplo, la instrucción `polyval` que se usa en el M-archivo de arriba, sirve para evaluar un polinomio en los elementos de un vector. También es importante saber más de la instrucción $x = A\backslash y$. Cuando los tamaños de las matrices involucradas son compatibles, esta instrucción entrega la solución del sistema lineal

$$Ax = y,$$

ya sea la única, en caso de haber una sola o la de mínimos cuadrados. La simplicidad de esta instrucción y la calidad de las rutinas MATLAB que la respaldan, son dos de las razones para que este software sea tan exitoso. La ayuda en línea de MATLAB puede ayudar a completar esta información.

4.2.2 Intento 2

Un polinomio de grado mínimo, en este caso 3, que sea interpolante. El grado está determinado por el número de parejas de datos disponibles. Como hay 4, se busca un polinomio de grado 3 que tiene 4 coeficientes por encontrar. Más precisamente:

Queremos hallar a, b, c, d tales que el polinomio

$$p(x) = a + b(x-1) + c(x-1)(x-2) + d(x-1)(x-2)(x-3) \quad (4.3)$$

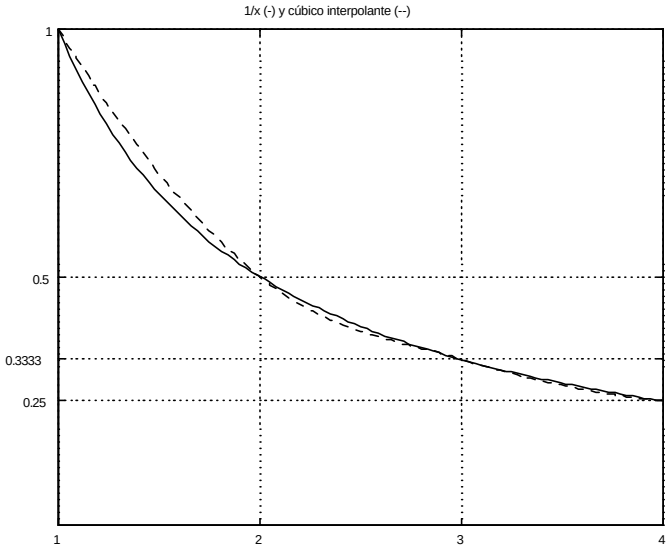
sea tal que $p(x_j) = y_j$, $j = 0, 1, 2, 3$. La escritura del polinomio interpolante en esta forma es cómoda para evaluaciones y es la que después llamaremos *forma de Newton.* Para resolver a mano este problema, acudimos al *método de los coeficientes indeterminados.* Consiste en utilizar evaluaciones del polinomio (4.3) en las abscisas de la tabla para formar un sistema de ecuaciones en el que las incógnitas son los coeficientes a, b, c, d. Veamos el uso del método para el polinomio

interpolante de la tabla (4.2).

$$\begin{aligned}
&\text{Para 1:} && 1 = p\,(1) = a, && \text{luego } a = 1\\
&\text{Para 2:} && \frac{1}{2} = p(2) = a + b, && \text{luego } b = -\frac{1}{2}\\
&\text{Para 3:} && \frac{1}{3} = p(3) = a + 2b + 2c, && \text{luego } c = \frac{1}{6}\\
&\text{Para 4:} && \frac{1}{4} = p(4) = a + 3b + 6c + 6d, && \text{luego } d = -\frac{1}{24}.
\end{aligned}$$

Para resolver el problema con MATLAB, preparamos el M-archivo pol3.m, que incluimos a continuación junto con la gráfica que produce.

```
% pol3, Herramienta didactica, Carlos E. Mejia, 2002
z=1:3/100:4;
x=[1 2 3 4]';y=1./x;
polin=interp1(x,y,z,'cubic');
plot(z,1./z,'k',z,polin,'k--');
set(gca,'XTick',x'),grid on
axis([1 4 0 1]),
set(gca,'YTick',[1/4 1/3 1/2 1]),
title('1/x (-) y cúbico interpolante (--)');
print -deps2 .\fig\pol3.eps
```



La interpolación polinomial esta respaldada por el siguiente teorema de existencia y unicidad.

Teorema 12 *Dada una tabla con abscisas x_j (todas diferentes) y ordenadas y_j, $j = 0, 1, 2, ..., n$, existe un único polinomio p de grado a lo más n tal que*

$$p(x_j) = y_j. \tag{4.4}$$

dem:

Unicidad

Supongamos que hay dos polinomios q y r que satisfacen (7.2). El polinomio $q - r$ es tal que $(q - r)(x_j) = 0$ para $j = 0, 1, ..., n$. Es decir, tiene grado a lo más n y sin embargo cuenta con $n + 1$ raíces. Esto obliga a que sea el polinomio nulo o sea que $q = r$.

Existencia

Inducción sobre n :

$n = 0$: La función $x_0 \to y_0$ se interpola con el polinomio constante $p(x) = y_0$.

Supongamos que hay polinomio interpolante p_{k-1} de grado $k - 1$ para los datos (x_j, y_j), $j = 0, 1, ..., k - 1$. Es decir, $p_{k-1}(x_j) = y_j$, $j = 0, 1, ..., k - 1$.

Consideremos el polinomio

$$p_k(x) = p_{k-1}(x) + c(x - x_0)(x - x_1)...(x - x_{k-1}).$$

Por supuesto es interpolante de los datos hasta $k-1$, pues p_{k-1} lo es. Para que también interpole en x_k, pedimos que

$$y_k = p_k(x_k) = p_{k-1}(x_k) + c(x_k - x_0)(x_k - x_1)...(x_k - x_{k-1})$$

lo cual proporciona un único valor para c. El despeje de c se puede hacer pues los x_j son todos distintos por hipótesis.

4.2.3 Cálculo de coeficientes

La idea en la demostración de existencia nos lleva a considerar el siguiente procedimiento recurrente:

$$\begin{aligned} p_0(x) &= y_0 = c_0 \\ p_1(x) &= p_0(x) + c_1(x - x_0) \\ p_2(x) &= p_1(x) + c_2(x - x_0)(x - x_1) \\ &\vdots \\ p_k(x) &= \sum_{j=0}^{k} c_j \prod_{s=0}^{j-1}(x - x_s) \end{aligned}$$

con la convención $\prod_{s=0}^{m}(x - x_s) = 1$ si $m < 0$.

Para evaluar los p_k y por tanto, para calcular los c_k, se usa la multi-

plicación anidada. El algoritmo es como sigue:

$$
\begin{aligned}
& c_0 = y_0 \\
\text{Para } & k = 1, 2, ..., n \\
& d \leftarrow x_k - x_{k-1} \\
& u \leftarrow c_{k-1} \\
\text{Para } & i \geq 0 \text{ e } i = k-2, k-3, ..., 0 \\
& u \leftarrow u(x_k - x_i) + c_i \\
& d \leftarrow d(x_k - x_i) \\
& \text{Fin } i \\
& c_k \leftarrow \frac{y_k - u}{d} \\
& \text{Fin } k
\end{aligned}
$$

Continuamos considerando la tabla del teorema anterior, es decir,

j	0	1	$\cdots$	n
x_j	x_0	x_1	$\cdots$	x_n
y_j	y_0	y_1	$\cdots$	y_n

(4.5)

en la que todos los nodos x_j son distintos. Supondremos además que $y_j = f(x_j)$, donde f es la función que se quiere interpolar. Según el teorema anterior, hay un único polinomio, de grado a lo más n, que interpola esta tabla. A continuación presentamos dos maneras de conseguirlo que son muy comunes.

4.3 Forma de Newton

Volvamos a la idea de la demostración del teorema fundamental de la interpolación. Recordemos que se trata de una demostración por inducción en la que se van construyendo los polinomios de acuerdo con la siguiente estrategia:

$$
\begin{aligned}
p_0(x) &= y_0 = c_0 \\
p_1(x) &= p_0(x) + c_1(x - x_0) \\
p_2(x) &= p_1(x) + c_2(x - x_0)(x - x_1) \\
&\vdots \\
p_k(x) &= \sum_{j=0}^{k} c_j \prod_{s=0}^{j-1}(x - x_s)
\end{aligned}
$$

con la convención $\prod_{s=0}^{m}(x - x_s) = 1$ si $m < 0$.

Esta es la forma de Newton del polinomio interpolante de la tabla dada. Los coeficientes c_k, que se consiguen con el algoritmo visto antes, se conocen como diferencias divididas. Para ellos se utiliza generalmente la siguiente notación, que viene acompañada de la forma de obtenerlos por recurrencia:

$$
\begin{aligned}
c_0 &= y_0 = f[x_0] \\
c_1 &= \frac{y_1 - y_0}{x_1 - x_0} = f[x_0, x_1] \\
c_2 &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = f[x_0, x_1, x_2] \\
&\vdots \\
c_k &= \frac{f[x_1, x_2, \cdots x_k] - f[x_0, x_1, \cdots x_{k-1}]}{x_k - x_0} = f[x_0, x_1, \cdots, x_k]
\end{aligned}
$$

En general,

$$
f[x_k, x_{k+1}, \cdots, x_{k+j}] = \frac{f[x_{k+1}, x_{k+2}, \cdots x_{k+j}] - f[x_k, x_{k+1}, \cdots x_{k+j-1}]}{x_{k+j} - x_k} \tag{4.6}
$$

En resumen, la forma de Newton del polinomio interpolante de la tabla dada es

$$
p_n(x) = \sum_{j=0}^{n} f[x_0, x_1, \cdots, x_j] \prod_{s=0}^{j-1}(x - x_s).
$$

Las diferencias divididas se presentan generalmente en tablas como la siguiente:

$$\begin{array}{llllll}
x_0 & f[x_0] & f[x_0,x_1] & \cdots & f[x_0,\cdots,x_{n-1}] & f[x_0,x_1,\cdots x_n] \\
x_1 & f[x_1] & f[x_1,x_2] & \cdots & f[x_1,\cdots,x_n] & \\
x_2 & f[x_2] & \vdots & & & \\
\vdots & \vdots & & & & \\
x_{n-1} & f[x_{n-1}] & f[x_{n-1},x_n] & & & \\
x_n & f[x_n] & & & &
\end{array}$$

Cuando las tablas de diferencias divididas se organizan de esta manera, los coeficientes del polinomio interpolante quedan situados en la primera fila.

Note también que la diferencia dividida (4.6) se encuentra en la fila k y la columna j de la matriz $(n+1)\times(n+1)$ indizada de 0 a n en las dos direcciones, que se obtiene de la tabla de diferencias descartando la columna de las x_j. El numerador de (4.6) se consigue por substracción de los elementos $(k+1, j-1)$ y $(k, j-1)$ de esta matriz y el numerador se obtiene por substracción de x_{k+j} y x_k.

El siguiente M-archivo lo presentamos con el ánimo de mostrar el procedimiento de construcción de tablas de diferencias divididas. Advertimos que dista mucho de ser útil, por la cantidad de información que debe entrarse con el teclado. Como dijimos al principio, las rutinas presentadas en estas notas son solamente ejemplos de clase que no pretenden sustituir rutinas hechas por profesionales que son las que deben usarse siempre que estén disponibles.

```
% Herramienta didactica, Carlos E. Mejia, 2002
% uso: difdiv
% diferencias divididas
n=input(' Entre numero de abscisas: ');
x=input(' Entre vector de abscisas: ');
y=input(' Entre vector de ordenadas: ');
d=zeros(n);
d(:,1)=y(:);
for j=2:n
    for i=1:n-j+1
```

```
    d(i,j)=(d(i+1,j-1)-d(i,j-1))/(x(i+j-1)-x(i));
    end
end
d
```

Veamos unos resultados obtenidos con esta rutina. Ofrecemos los datos correspondientes al polinomio x^3 en una tabla con 4 abscisas. Los resultados son:

» difdiv
Entre numero de abscisas: 4
Entre vector de abscisas: [1 2 4 6]
Entre vector de ordenadas: [1 8 64 216]
d =

1	7	7	1
8	28	12	0
64	76	0	0
216	0	0	0

El resultado que ofrece difdiv.m es que el polinomio interpolante, que tiene que ser el propio x^3 por unicidad, está dado por

$$1 + 7\,(x-1) + 7\,(x-1)\,(x-2) + (x-1)\,(x-2)\,(x-4)\,.$$

Es decir, se obtiene el resultado correcto.

4.4 Forma de Lagrange

Hay una expresión equivalente para el polinomio interpolante de grado a lo más n que interpola la tabla (4.5). Se conoce como la forma de Lagrange. Está dada por

$$\begin{aligned} p_n(x) &= \sum_{k=0}^{n} y_k l_k(x) \\ &= \sum_{k=0}^{n} f\,(x_k)\,l_k(x), \end{aligned} \tag{4.7}$$

donde

$$l_k(x) = \prod_{s=0,\ s\neq k}^{n} \frac{(x - x_s)}{(x_k - x_s)}. \tag{4.8}$$

A los polinomios l_k se les llama polinomios básicos de Lagrange o funciones cardinales. Estos polinomios son de grado n y entonces el polinomio interpolante se consigue como una suma ponderada de polinomios de grado n.

Mencionamos esta forma porque es de interés teórico. Nos será de utilidad en la sección 5.5 sobre cuadraturas de Newton-Cotes y en la sección 5.11 sobre polinomios ortogonales y cuadratura gaussiana.

Una de las propiedades que utilizaremos más adelante, es

$$l_k(x_j) = \delta_{kj}, \tag{4.9}$$

que es una consecuencia directa de (4.8). Aquí δ_{kj} es el delta de Kronecker, que toma el valor 1 si $k = j$ y el valor 0 si $k \neq j$.

Con riesgo de sonar repetitivo, volvemos a insistir en que se recurra a software de calidad siempre que esté disponible. En particular, los cálculos con la forma de Lagrange, son más susceptibles de ser inestables numéricamente que los cálculos con diferencias divididas.

4.5 Ejercicios

1. De acuerdo con la revista Newsweek de junio 17 de 2002, pag. 27, el porcentaje de sillas vacías en la primera fase del Mundial de Fútbol, está dado por la siguiente tabla:

Año	1990	1994	1998
Porcentaje	18	8.7	10

a. Construya el polinomio interpolante de esta tabla de grado a lo más 2, utilizando diferencias divididas.

b. Evalúe este polinomio en 2002. Quisiéramos llamar a este número, el porcentaje correspondiente al Mundial de 2002. Esta clase de evaluación se llama *extrapolación.* No tiene que ofrecer resultados correctos, a menos que haya razones adicionales para tener en cuenta.

c. La misma revista informa que el porcentaje de sillas vacías en la primera fase del Mundial del 2002, es 22%. ¿Qué tan bueno fue el estimado obtenido en b.?

d. Aumente la tabla con la información para el año 2002. Construya el polinomio interpolante de esta tabla de grado a lo más 3, utilizando

diferencias divididas. Note que sólo tiene que completar el trabajo que ya hizo en la parte a., no necesita empezar desde el principio.

2. Use el método de coeficientes indeterminados para encontrar un polinomio cuadrático $p(x)$ tal que $p(1) = 0$, $p'(1) = 7$ y $p(2) = 10$. (Note la evaluación de la derivada.) La interpolación polinomial que incluye información sobre derivadas, es importante pero no hace parte de estas notas. Sugerimos a los interesados que estudien *interpolación de Hermite*, para darse una idea sobre el tema.

3. Una función *spline* de grado k con nodos x_0, $x_1, ...,$ x_n es una función S que cumple:

a. En cada intervalo $[x_{j-1}, x_j)$, S es un polinomio de grado $\leq k$.

b. S tiene derivadas continuas hasta de orden $k-1$ en $[x_0, x_n]$.

Ejemplos: Los splines de grado 0 son funciones constantes a trozos, los de grado 1 son funciones *continuas* lineales a trozos y los de grado 2 son continuas, con primera derivada continua y definidos en cada subintervalo por un polinomio cuadrático.

Determine los valores de a, b, c y d para que la función

$$f(x) = \begin{cases} 3 + x - 9x^3, & x \in [0,1] \\ a + b(x-1) + c(x-1)^2 + d(x-1)^3, & x \in [1,2] \end{cases}$$

sea un spline cúbico con nodos 0, 1 y 2.

Los ejercicios 2. y 3. abren un espacio para temas no tratados en detalle en estas notas. Sugerimos consultar Kincaid y Cheney (1994), cap. 6.

4.6 Error al interpolar

Si la función f que se está interpolando es suficientemente suave, las diferencias divididas están relacionadas con las derivadas de f y es posible acotar el error cometido al interpolar. Más precisamente, el teorema siguiente afirma que la diferencia dividida de f en $n+1$ puntos distintos, se puede obtener por evaluación de $f^{(n)}$ en un cierto punto.

Teorema 13 *Si $f \in C^n[a,b]$ y si x_0, $x_1, ...,$ x_n son $n+1$ puntos distintos de $[a,b]$, entonces existe un punto $z \in (a,b)$ tal que*

$$f[x_0, x_1, \cdots x_n] = \frac{f^{(n)}(z)}{n!}$$

El error al interpolar también puede expresarse en términos de derivadas de f cuando f es suficientemente suave.

Teorema 14 *Sean $f \in C^{n+1}[a,b]$ y p_n un polinomio de grado $\leq n$ que interpola a f en $n+1$ puntos distintos $x_0, x_1, ..., x_n$ de $[a,b]$. Para cada $x \in [a,b]$, corresponde un punto $z_x \in (a,b)$ tal que*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(z_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j).$$

En particular, si x no es uno de los nodos x_j, $j = 0, 1, ..., n$, el error al interpolar también puede escribirse así:

$$f(x) - p_n(x) = f[x_0, x_1, \cdots x_n, x] \prod_{j=0}^{n} (x - x_j).$$

4.7 Ejercicios suplementarios

1. Encuentre los polinomios interpolantes de grado mínimo (a lo más 3), que interpolan las siguientes tablas. Utilice forma de Lagrange para la primera tabla, forma de Newton en la segunda y repita los cálculos de ambas tablas por el método de coeficientes indeterminados.

Tabla 1				
x	-2	-1	0	1
y	2	14	4	2

Tabla 2				
x	-1	0	2	3
y	-1	3	11	27

2. Encuentre el polinomio de grado a lo más 4, que satisface la siguiente tabla:

x	-2	-1	0	1	2
y	2	14	4	2	2

3. Forme una tabla de diferencias divididas para los datos siguientes y explique lo que pasa.

x	-2	-1	0	-2
y	2	14	4	6

4. Verifique directamente que si x_1, x_2, x_3 son puntos distintos, entonces
$$f[x_1, x_2, x_3] = f[x_3, x_1, x_2] = f[x_2, x_3, x_1].$$

5. Encuentre el polinomio de grado a lo más 4, que satisface la siguiente tabla:

x	0	1	3	2	5
y	2	1	5	6	-183

6. La función $\cos(x)$ se puede aproximar con un polinomio interpolante de grado n usando $n+1$ nodos igualmente espaciados en el intervalo $[0,1]$. Exprese un estimado del error que se comete en términos de n. ¿Para qué valores de n es este error menor que 10^{-7}?

7. (Continuación) Realice el mismo experimento del ejercicio anterior usando como nodos, los llamados nodos de Chebyshev, que son $x_j = 5\cos(j\pi/20)$, $j = 0, 1, ..., 20$.

8. Encuentre un polinomio interpolante de grado 15 para la función $f(x) = arcsen(x)$ en el intervalo $\left[-1/\sqrt{2}, 1/\sqrt{2}\right]$. Afirme si considera que esta aproximación es buena o no con justificaciones.

9. Suponga que estamos hallando un polinomio interpolante para una función en $[0,1]$, usando solamente dos puntos x_0 y x_1. ¿Cuáles deben ser x_0 y x_1 para que el factor $(x - x_0)(x - x_1)$ del error al interpolar sea mínimo?

10. Se propone encontrar un polinomio interpolante de la función $f(x) = (1 + 25x^2)^{-1}$ en el intervalo $[-1,1]$, de las siguientes formas:
a. Utilizando los puntos $x_{jn} = -1 + 2j/n$, $j = 0, 1, ..., n$, para $n = 5, 10, 25$.
b. Utilizando los puntos $x_{jn} = \cos\left(\frac{2j\pi}{n}\right)$.
Haga gráficas de las funciones de error $f(x) - p(x)$ en todos los casos. Comente sus resultados.

11. Para la tabla

x	-2	-1	0	1	2	3	4
$f(x)$	-39	1	1	3	25	181	801

se construyó su tabla de diferencias divididas

-2	-39	40	-20	7	-1	1	0
-1	1	0	1	3	4	1	
0	1	2	10	19	9		
1	3	22	67	55			
2	25	156	232				
3	181	620					
4	801						

Utilice esta tabla para construir el polinomio interpolante de $f(x)$ en cada uno de los siguientes conjuntos de nodos: $\{-1,0,1\}$, $\{-1,0,1,2\}$, $\{-1,0,1,2,3\}$, $\{0,1\}$ y $\{0,1,2,3,4\}$.

4.8 Examen de entrenamiento

Resuelva estos ejercicios sin consultar libros ni notas de clase para darse una idea del nivel de preparación que ha obtenido hasta ahora. Una calculadora sencilla es lo mínimo que debe tener a disposición y es suficiente para poder responder todos los ejercicios del examen. Claro que es preferible si dispone de una calculadora graficadora o un computador.

1. Calcule el polinomio p de grado 2 que satisface la tabla

x	0	1	2
$p(x)$	0	1	0

utilizando los tres métodos estudiados, es decir, la forma de Lagrange, la forma de Newton y el método de coeficientes indeterminados.

2. Sea $f(x) = sen\left(\frac{\pi x}{2}\right)$ y sea p el polinomio p del ejercicio anterior que coincide con f en los puntos 0, 1 y 2. Calcule una cota para el error $|f(x) - p(x)|$ en $[0,2]$. Compare esta cota con el error verdadero en los puntos $x = 1/4$ y $x = 3/4$.

3. Encuentre el polinomio p de grado 3 que interpola a $\sqrt{x}$ en los nodos 0, 1, 3, y 4. Compare $p(2)$ con $\sqrt{2} = 1.414216$.

4. Sea p el polinomio que satisface la tabla

x	-2	-1	0	1	2	3
$p(x)$	-5	1	1	1	7	25

   ¿Qué puede decir del grado de p?

5

Integración numérica

Fórmulas básicas
Error en la cuadratura
Cuadraturas compuestas
Fórmulas de Newton-Cotes
Coeficientes indeterminados
Cuadratura de Gauss
Polinomios ortogonales

La integral definida $\int_a^b f(x)\,dx$ es un número y el proceso de calcularlo, con base en valores de la función f, se conoce como *integración numérica* o *cuadratura*. (Esta última palabra se refiere a encontrar un cuadrado cuya área sea igual al área bajo una curva.) Más precisamente, toda fórmula de integración numérica para la integral $\int_a^b f(x)\,dx$ es una suma de la forma

$$\sum_{j=0}^{n} A_j f(x_j).$$

Las formas de escoger los x_j y los A_j, generan los distintos tipos de cuadratura.

Tal como dijimos al principio de estas notas, frecuentemente lo que se requiere para cálculos numéricos son aproximaciones numéricas en lugar de soluciones analíticas. En este caso, podemos decir que utilizar integración numérica es tan importante cuando no se conoce antiderivada para la función como cuando se conoce.

Para presentar la integración numérica, seguimos a Stewart (1996) [29] y Kincaid y Cheney (1994) [21]. Nos dedicamos únicamente a integrales unidimensionales definidas en intervalos cerrados. No consideramos otros dominios de integración ni funciones con singularidades. Para tratamiento avanzado del tema, sugerimos Isaacson y Keller (1994) [17] y el libro especializado Davis y Rabinowitz (1984) [9].

5.1 Fórmulas básicas

Una fórmula típica es la *Regla de Simpson:*

$$\int_0^1 f(x)\,dx \cong \frac{1}{6}\left[f(0) + 4f\left(\frac{1}{2}\right) + f(1)\right].$$

El símbolo $\cong$ se lee: *es aproximadamente igual a.* Empecemos usando esta regla y la más elemental de las rutinas de cuadratura que ofrece MATLAB.

Ejemplo 15 *Utilizar la rutina básica de cuadratura proporcionada por MATLAB, llamada quad, y la regla de Simpson básica para aproximar las integrales entre* 0 *y* 1 *de las siguientes funciones:* $f(x) = |\sin(2\pi x)|$, $f(x) = \frac{2}{\sqrt{\pi}}\exp(-x^2)$ *y* $f(x) = x(x-0.5)(x-1)+5$.

Solución: Las soluciones exactas se conocen. En el segundo ejemplo no existe antiderivada, pero la función de error

$$\operatorname{erf}(z) = \int_0^z \frac{2}{\sqrt{\pi}}\exp\left(-x^2\right)dx$$

está tabulada. Las tablas para erf y muchas otras funciones definidas en términos de integrales, son indicativo de lo importante que es tener buenas rutinas para integración numérica. Dichas tablas se pueden consultar, por ejemplo, en la compilación Abramowitz y Stegun (1970) [1].

Preparamos el siguiente M-archivo para hacer los cálculos:

```
function simpson(fun,a,b)
% Herramienta didactica, Carlos E. Mejia, 2002
% uso de quad, la integracion numerica de bajo orden
% que ofrece MATLAB junto con regla de Simpson
% quad esta basada en Simpson y es adaptativa
% fun: integrando, a, b: limites de integracion
% Uso: simpson('fun',a,b), fun puede ser fsim.m
i1=quad(fun,0,1);
c=(b-a)/6;
fa=feval(fun,a);
f2=feval(fun,(a+b)/2);
```

```
fb=feval(fun,b);
i2=c*(fa+4*f2+fb);
disp(['Resultado quad: ',num2str(i1)]);
disp(['Resultado Simpson básico: ',num2str(i2)]);
```

Los resultados obtenidos se comparan con los exactos en la siguiente tabla:

Integral	quad.m	Simpson	Exacto
$\int_0^1 \lvert\sin(2\pi x)\rvert\, dx$	0.6366	1.2246×10^{-16}	0.6366
$\int_0^1 \frac{2}{\sqrt{\pi}} \exp\left(-x^2\right) dx$	0.8427	0.8431	0.8427
$\int_0^1 (x\,(x-0.5)\,(x-1)+5)\, dx$	5	5	5

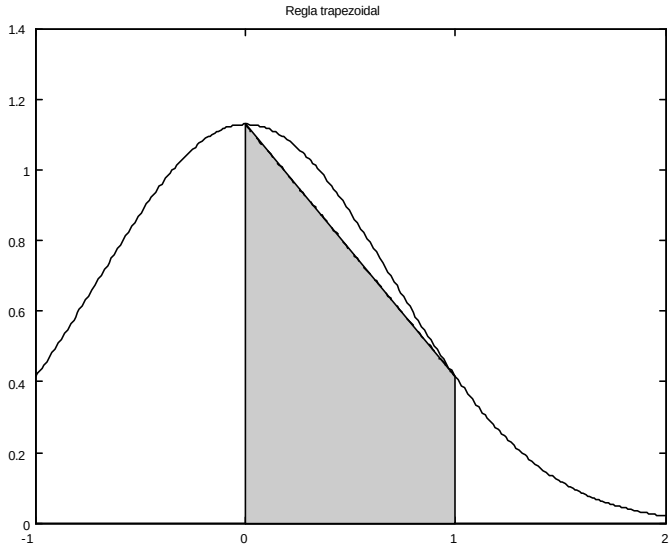
El fracaso de la regla de Simpson en la aproximación de la primera integral se debe a que en $x = 0$, 0.5 y 1, el valor del integrando es cero. La rutina quad.m acierta en los tres ejemplos. La regla de Simpson no fracasa en la segunda integral, simplemente ofrece una aproximación menos buena que quad.m. El éxito de ambas rutinas en el tercer ejemplo es obligatorio, pues el integrando es un polinomio de grado 3 y la regla de Simpson es exacta para polinomios hasta de grado 3, como veremos enseguida.

Otra fórmula básica es la regla trapezoidal, definida así:

$$\int_0^1 f(x)\, dx \cong \frac{1}{2}\left[f(0) + f(1)\right].$$

Si f es una función no negativa, la regla trapezoidal consiste en aproximar el área bajo la curva por el área del trapecio con bases $f(0)$ y $f(1)$ y altura 1. Para la función $\frac{2}{\sqrt{\pi}} \exp\left(-x^2\right)$, las dos áreas se pueden

ver en la siguiente gráfica:



5.1.1 Ejercicio

Repetir el ejemplo 15 con la regla trapezoidal en lugar de la regla de Simpson. Comentar acerca de los resultados obtenidos.

5.1.2 Cambio de intervalo

Las dos fórmulas básicas que hemos presentado fueron dadas para el intervalo $[0, 1]$ pero el M-archivo simpson.m, lo escribimos con base en un intervalo genérico $[a, b]$. Los cambios de intervalo son tan fáciles de hacer, que a menudo se presentan las fórmulas únicamente para $[0, 1]$.

Para cambiar de intervalo, basta escribir la variable x como función *lineal* de otra variable y, así:

$$x = p + qy.$$

Enseguida pedimos $x = a$ cuando $y = 0$ y $x = b$ cuando $y = 1$, lo cual exige tomar $p = a$ y $q = b - a$. La expresión anterior se cambia por

$$x = a + (b - a)\, y$$

y además, también escribimos

$$g(y) = f(a + (b - a)\, y).$$

Finalmente, $dx = (b-a)dy$ y la integral $\int_a^b f(x)\,dx$ la podemos evaluar así:

$$\int_a^b f(x)\,dx = (b-a)\int_0^1 g(y)\,dy.$$

Como

$$g(0) = f(a),\ g\left(\frac{1}{2}\right) = f\left(\frac{a+b}{2}\right) \text{ y } g(1) = f(b),$$

entonces en el intervalo $[a,b]$ la regla de Simpson es

$$\int_a^b f(x)\,dx \cong \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

y la regla trapezoidal es

$$\int_a^b f(x)\,dx \cong \frac{b-a}{2}\left[f(a) + f(b)\right].$$

5.2 Error en la cuadratura

Sean

$$S(f,a,b) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

y

$$T(f,a,b) = \frac{b-a}{2}\left[f(a) + f(b)\right].$$

Si f tiene suficientes derivadas, los errores en cada una de estas cuadraturas están dados por

$$\int_a^b f(x)\,dx - S(f,a,b) = -\frac{1}{90}(b-a)^5 f^{(4)}(\tau)$$

y

$$\int_a^b f(x)\,dx - T(f,a,b) = -\frac{1}{12}(b-a)^3 f^{(2)}(\eta),$$

donde τ y η están entre a y b. La justificación de estas igualdades puede verse en Kincaid y Cheney (1994) [21].

Una consecuencia de estas fórmulas de error, es que la regla de Simpson es exacta para polinomios hasta de grado 3 y que la regla trapezoidal es exacta para polinomios hasta de grado 1. En efecto, la cuarta derivada de un polinomio de grado 3 o menor, es cero y la segunda derivada de un polinomio de grado 1 es cero. Estas ideas se generalizan en la sección 5.5 sobre cuadraturas de Newton-Cotes.

Otra consecuencia de estas fórmulas, es que sobre intervalos grandes, es prácticamente imposible obtener buenos resultados con estas cuadraturas. Por eso se recurre a las cuadraturas compuestas, que se consideran en la siguiente sección.

5.3 Cuadraturas compuestas

Las cuadraturas trapezoidal y de Simpson normalmente se aplican sobre intervalos pequeños que conforman uno grande, pues de esa manera se obtienen errores menores. Supongamos que queremos aproximar $I = \int_a^b f(x)\,dx$. Entonces dividimos el intervalo $[a,b]$ en n subintervalos, cada uno de longitud $h = \dfrac{b-a}{n}$. Sean $x_j = a + jh,\ j = 0, 1, ..., n$. Sabemos que

$$T(f, x_{j-1}, x_j) = \frac{h}{2}\left(f(x_{j-1}) + f(x_j)\right)$$

y al sumar todas estas aproximaciones, obtenemos una aproximación para I. Tal suma la denotamos $CT(f,a,b)$ y la llamamos *regla trapezoidal compuesta*. Su valor es

$$\begin{aligned} CT(f,a,b) &= \sum_{j=1}^{n} T(f, x_{j-1}, x_j) \\ &= \sum_{j=1}^{n} \frac{h}{2}\left(f(x_{j-1}) + f(x_j)\right) \\ &= \frac{h}{2} f(x_0) + \sum_{j=1}^{n-1} f(x_j) + \frac{h}{2} f(x_n). \end{aligned}$$

Si f es suficientemente suave, el error para esta cuadratura es

$$\int_a^b f(x)\,dx - CT(f,a,b) = -\frac{(b-a)h^2 f^{(2)}(\tau)}{12},$$

donde $\tau \in (a,b)$. El factor h^2 es muy importante. Indica que si el número de puntos se duplica, el nuevo error es del *mismo orden de magnitud que el anterior error dividido por 4.*

De forma análoga se define la *regla de Simpson compuesta.* Requiere para su definición que el número n de subintervalos sea par. Los detalles pueden verse en un libro de análisis numérico, por ejemplo, Kincaid y Cheney (1994) [21].

5.4 Ejercicios

Las integrales de Fresnel de tipo coseno se definen asi:

$$C(z) = \int_0^z \cos\left(\frac{\pi}{2}t^2\right) \text{ para todo real.}$$

Tales integrales y las de tipo seno que se definen análogamente, aparecen tabuladas en una gran variedad de colecciones de funciones, por ejemplo en Abramowitz y Stegun (1970) [1].

1. Construya una tabla de los valores de la función $C(z)$ para todo z de la forma $z = 0.05k$, donde $k = 1, 2, ..., 20$. *Necesariamente debe utilizar alguna rutina para integración numérica.* Dicha rutina puede crearla usted o tomarla de alguna fuente.

2. Repita el ejercicio 1. para la función

$$S(z) = \int_0^z sen\left(\frac{\pi}{2}t^2\right) \text{ para todo real.}$$

Estas son las integrales de Fresnel de tipo seno.

3. Establezca la fórmula de cuadratura compuesta de Simpson que denotaremos $CS(f,a,b)$. Puede hacerlo por deducción directa o consultando un libro de análisis numérico. Enseguida, vuelva al ejemplo 15. Repita los cálculos para las tres funciones, utilizando las fórmulas compuestas de cuadratura $CT(f,a,b)$ y $CS(f,a,b)$, para $n = 2^k$, $k = 0, 1, ..., 4$.

4. Calcule $C(1)$ y $S(1)$ por medio de las fórmulas compuestas de cuadratura $CT(f,a,b)$ y $CS(f,a,b)$, para $n = 4$.

5.5 Cuadraturas de Newton-Cotes

Buscamos una cuadratura para $\int_a^b f(x)\,dx$ de la forma $\sum_{j=0}^{n} A_j f(x_j)$, donde $x_0, x_1, ..., x_n$ son puntos distintos del intervalo $[a,b]$. Queremos además que sea exacta para polinomios hasta de grado n. ¿Será posible encontrar $A_0, A_1, ..., A_n$ tales que para todo polinomio p de grado a lo más n, la igualdad

$$\int_a^b p(x)\,dx = \sum_{j=0}^{n} A_j p(x_j)$$

es cierta? La respuesta es afirmativa y la escribimos en forma de teorema.

Teorema 16 *(Cuadratura de Newton-Cotes) Sean l_j, $j = 0, 1, ..., n$ los polinomios básicos de Lagrange sobre $x_0, x_1, ..., x_n$. Entonces*

$$A_j = \int_a^b l_j(x)\,dx, \;\; j = 0, 1, ..., n \tag{5.1}$$

son los únicos coeficientes tales que

$$grad(p) \leq n \Rightarrow \int_a^b p(x)\,dx = \sum_{j=0}^{n} A_j p(x_j). \tag{5.2}$$

dem:

Puesto que los polinomios de Lagrange tienen grado n, para ellos debe cumplirse el lado derecho de (5.2), es decir,

$$\int_a^b l_k(x)\,dx = \sum_{j=0}^{n} A_j l_k(x_j) = A_k.$$

La última igualdad se debe a (4.9). De manera que el único posible valor para cada A_j es el propuesto en (5.1).

Tomemos ahora un polinomio p de grado a lo más n. De (4.7),

$$p(x) = \sum_{j=0}^{n} p(x_j)\, l_j(x).$$

Por tanto,

$$\int_a^b p(x)\,dx = \sum_{j=0}^{n} p(x_j) \int_a^b l_j(x)\,dx = \sum_{j=0}^{n} A_j p(x_j).$$

Es apropiado observar que para cada n y cada colección de $n+1$ puntos, hay una cuadratura de Newton-Cotes.

Ejemplo 17 *La regla trapezoidal es la cuadratura de Newton-Cotes sobre* $[a,b]$ *con* $n=1$, $x_0=a$ *y* $x_1=b$.

Sin pérdida de generalidad, trabajamos en $[0,1]$. Es fácil ver que

$$\int_0^1 l_k(x)\,dx = \frac{1}{2}$$

para $k=0,1$. Es decir, para aproximar $\int_0^1 f(x)\,dx$, la cuadratura de Newton-Cotes sobre $[0,1]$ con $n=1$, $x_0=0$ y $x_1=1$ es

$$\frac{1}{2}\left(f(0)+f(1)\right),$$

que es $T(f,0,1)$.

Normalmente, esta no es la forma de encontrar los factores de ponderación A_j en una cuadratura de Newton-Cotes. Lo que se hace es recurrir al método de coeficientes indeterminados, que convierte el problema en la solución de un sistema de ecuaciones lineales.

5.6 Método de coeficientes indeterminados

Explicamos el método por medio de la solución del siguiente ejemplo.

Ejemplo 18 *Encontrar la cuadratura de Newton-Cotes sobre* $[0,1]$ *con* $n=2$, $x_0=0$, $x_1=0.5$ *y* $x_2=1$.

Buscamos una cuadratura para $\int_0^1 f(x)\,dx$ que sea de la forma

$$\sum_{j=0}^{2} A_j f(x_j)$$

y que sea exacta para polinomios de grado a lo más 2. Los coeficientes que encontremos, serán los de la cuadratura de Newton-Cotes pedida, es decir, los dados en (5.1).

Escogemos 1, x y x^2 como los polinomios de grados 0, 1 y 2 respectivamente, que nos permitan enunciar un sistema por medio del cual encontramos los coeficientes A_j de la cuadratura. Evaluamos el polinomio y la cuadratura para cada uno de los polinomios, así:

$$\text{Para } 1: \quad 1 \cdot A_0 + 1 \cdot A_1 + 1 \cdot A_2 = \int_0^1 1dx = 1$$

$$\text{Para } x: \quad 0 \cdot A_0 + 0.5 \cdot A_1 + 1 \cdot A_2 = \int_0^1 x dx = 0.5$$

$$\text{Para } x^2: \quad 0 \cdot A_0 + \frac{1}{4} \cdot A_1 + 1 \cdot A_2 = \int_0^1 x^2 dx = \frac{1}{3}.$$

Este sistema tine una única solución dada por $A_0 = A_2 = \dfrac{1}{6}$ y $A_1 = \dfrac{2}{3}$. Estamos ante la cuadratura de Newton-Cotes sobre $[0, 1]$ con $n = 2$, $x_0 = 0$, $x_1 = 0.5$ y $x_2 = 1$ y no es más que la regla de Simpson.

Esta cuadratura tiene otra particularidad, que ya conocemos desde que vimos los errores en la sección 5.2. Para x^3, la ecuación que resulta es

$$\frac{1}{8}A_1 + A_2 = \int_0^1 x^3 dx = \frac{1}{4}$$

y en efecto, los valores encontrados antes para A_1 y A_2, también satisfacen esta ecuación. Esto significa que la regla de Simpson también es exacta para polinomios de grado 3.

Nótese que la regla de Simpson **no** es una cuadratura de Newton-Cotes para $n = 3$, pues no se utilizan 4 puntos en su formulación.

5.7 Cuadratura de Gauss

Para el cálculo aproximado de la integral $\int_a^b f(x)\, dx$, tenemos cuadraturas

$$\sum_{j=0}^{n} A_j f(x_j) \tag{5.3}$$

con $n+1$ nodos y $n+1$ coeficientes de ponderación que son exactas para polinomios de grado a lo más n. Hasta ahora, los nodos están fijos de antemano y hay $n+1$ coeficientes de ponderación A_j por determinar, tantos como coeficientes tiene un polinomio de grado n. Ahora intentamos algo más: considerar nodos y coeficientes de ponderación como incógnitas, un total de $2n+2$ incógnitas. Veremos que existe una fórmula de cuadratura (5.3) que es exacta para polinomios de grado a lo más $2n+1$. Dicha cuadratura se conoce como *cuadratura de Gauss* o *gaussiana.*

Empecemos como antes, con el método de coeficientes indeterminados y un ejemplo similar al ejemplo 17.

Ejemplo 19 *Utilizar el método de coeficientes indeterminados para hallar el sistema de ecuaciones correspondiente a la cuadratura gaussiana para* $n=1$ *en* $[0,1]$.

Nuestras $2n+2$ incógnitas son x_0, x_1, A_0 y A_1. Escogemos los polinomios 1, x, x^2 y x^3 para las evaluaciones.

$$\begin{aligned}
&\text{Para } 1: \quad A_0 + A_1 = \textstyle\int_0^1 1dx = 1 \\
&\text{Para } x: \quad x_0A_0 + x_1A_1 = \textstyle\int_0^1 xdx = 0.5 \\
&\text{Para } x^2: \quad x_0^2A_0 + x_1^2A_1 = \textstyle\int_0^1 x^2dx = \frac{1}{3} \\
&\text{Para } x^3: \quad x_0^3A_0 + x_1^3A_1 = \textstyle\int_0^1 x^3dx = \frac{1}{4}.
\end{aligned} \tag{5.4}$$

Lo primero que salta a la vista es que el sistema al que llegamos es **no lineal.** Para resolverlo, podemos hacer algunas manipulaciones algebraicas o recurrir al método de Newton, que en estas notas aparece en la sección 6.15. Pero la no linealidad no es el único problema que este sistema tiene, pues es sabido que padece de inestabilidad numérica, es decir, un pequeño error en un paso puede conducir a soluciones lejanas a las verdaderas.

Los dos inconvenientes, no linealidad e inestabilidad numérica, están presentes en todos los sistemas a los que se llega si se intenta el método de coeficientes indeterminados para conseguir cuadraturas gaussianas.

¿Qué hacer entonces? Lo mismo que hizo Gauss (1777-1855) hace cerca de 200 años: utilizar polinomios ortogonales, que es lo que haremos en la sección 5.9.

5.8 Ejercicios

1. Resuelva exactamente el sistema de ecuaciones (5.4) y obtenga una fórmula de cuadratura gaussiana sobre $[0,1]$ que es exacta para polinomios de grado a lo más 3.

2. Utilice la cuadratura gaussiana que acaba de deducir para calcular las integrales del ejemplo 15.

3. Calcule $C(1)$ y $S(1)$, las integrales de Fresnel para $z = 1$, por medio de esta cuadratura gaussiana. Las integrales de Fresnel fueron definidas en la sección de ejercicios 5.4.

5.9 Polinomios ortogonales

La cuadratura de Gauss que buscamos es bastante más general que la presentada antes. Sea $w(x)$ una función positiva y continua en $[a,b]$ que llamaremos *función de ponderación* (en inglés *weight).* Queremos aproximar la integral

$$\int_a^b f(x)\, w(x)\, dx$$

por medio de una combinación lineal de valores de la función

$$\sum_{j=0}^{n} A_j f(x_j).$$

Definición 20 *Dos funciones f y g se dice que son ortogonales con respecto a w en el intervalo $[a,b]$ si*

$$\int_a^b f(x)\, g(x)\, w(x)\, dx = 0. \tag{5.5}$$

En realidad, si nos concentramos a trabajar en el espacio vectorial de funciones continuas en $[a,b]$, denotado $C([a,b])$, la expresión $\int_a^b f(x)\, g(x)\, w(x)\, dx$ define un producto interno en $C([a,b])$. En estas

condiciones, la expresión (5.5) dice que el producto interno de las funciones f y g es cero. Eso es lo que se llama ortogonalidad en espacios vectoriales.

Definición 21 *Una sucesión de polinomios ortogonales es una sucesión de polinomios* $\{p_j\}_{j=0}^{\infty}$ *en la que* $grad(p_j) = j$ *para cada* j *y tal que*

$$j \neq k \Rightarrow \int_a^b p_j(x)\, p_k(x)\, w(x)\, dx = 0.$$

Como la ortogonalidad no se altera por multiplicación por escalares no nulos, normalizamos los polinomios p_j de manera que los coeficientes de los términos de mayor grado x^j sean 1. A los polinomios con esa característica, se les llama *mónicos.*

De ahora en adelante trabajamos con sucesiones de polinomios ortogonales mónicos. El primer resultado que mencionamos es que, de haber una tal sucesión de polinomios ortogonales $\{p_j\}_{j=0}^{\infty}$, todo polinomio de grado n se puede escribir de forma única como una combinación lineal con p_0, p_1, ..., p_n.

Teorema 22 *Sea* $\{p_j\}_{j=0}^{\infty}$ *una sucesión de polinomios ortogonales mónicos (ver definición 21.) Si* $q(x) = \sum_{j=0}^{n} a_j x^j$ *es un polinomio de grado a lo más* n*, entonces existen escalares* b_j *tales que*

$$q = \sum_{j=0}^{n} b_j p_j.$$

Además, si $q = \sum_{j=0}^{n} b_j p_j = \sum_{j=0}^{n} c_j p_j$*, entonces* $b_j = c_j$ *para cada* j*.*

La demostración de este resultado es una inducción sobre n. Puede verse, por ejemplo, en Stewart (1996), [29].

El segundo resultado que nos interesa es un corolario del anterior. Consiste en notar que p_{n+1} es ortogonal a cualquier polinomio q de grado n o menor, pues lo que hacemos es escribir a q como en el teorema

anterior y multiplicar. Más precisamente: Si $q = \sum_{j=0}^{n} b_j p_j$, entonces

$$\int_a^b p_{n+1}(x)\, q(x)\, w(x)\, dx = \sum_{j=0}^{n} b_j \int_a^b p_{n+1}(x)\, p_j(x)\, w(x)\, dx = 0.$$

La igualdad a cero se da por la ortogonalidad de la sucesión de polinomios. Hemos probado

Corolario 23 *El polinomio p_{n+1} es ortogonal a cualquier polinomio de grado n o menor.*

5.10 Fórmula de recurrencia de tres términos

La existencia de sucesiones de polinomios ortogonales mónicos, no ha sido establecida aún. Intentemos su construcción.

Por ser todos mónicos, obligatoriamente p_0 es el polinomio constante 1 y $p_1(x) = x - d_1$. ¿Podremos identificar a d_1? La respuesta es afirmativa, lo hacemos por ortogonalidad. Como

$$\begin{aligned} 0 = \int_a^b p_1(x)\, p_0(x)\, w(x)\, dx &= \int_a^b (x - d_1)\, w(x)\, dx \\ &= \int_a^b x w(x)\, dx - d_1 \int_a^b w(x)\, dx, \end{aligned}$$

entonces

$$d_1 = \frac{\int_a^b x w(x)\, dx}{\int_a^b w(x)\, dx} = \frac{\int_a^b x p_0^2(x)\, w(x)\, dx}{\int_a^b p_0^2(x)\, w(x)\, dx}.$$

La re-escritura propuesta para d_1 se verá lógica en un momento, cuando encontremos una expresión general para d_{n+1}. Además, el denominador es no nulo pues el integrando es continuo y positivo.

La idea es buscar a p_{n+1} en la forma

$$p_{n+1} = x p_n - d_{n+1} p_n - e_{n+1} p_{n-1} - f_{n+1} p_{n-2} - ... \tag{5.6}$$

utilizando apropiadamente la ortogonalidad de la sucesión. Para d_{n+1},

imitamos lo hecho para d_1, es decir,

$$\begin{aligned} 0 = \int_a^b p_{n+1}(x)\, p_n(x)\, w(x)\, dx &= \int_a^b x p_n^2(x)\, dx \\ &\quad - d_{n+1} \int_a^b p_n^2(x)\, w(x)\, dx \\ &\quad - e_{n+1} \int_a^b p_n(x)\, p_{n-1}(x)\, w(x)\, dx - \ldots \end{aligned}$$

En el lado derecho, los términos después del segundo son cero por ortogonalidad. Luego

$$d_{n+1} = \frac{\int_a^b x p_n^2(x)\, w(x)\, dx}{\int_a^b p_n^2(x)\, w(x)\, dx}. \tag{5.7}$$

De nuevo, el denominador es no nulo pues el integrando es continuo y positivo.

Similarmente, usando ortogonalidad, se encuentra que

$$e_{n+1} = \frac{\int_a^b x p_n(x)\, p_{n-1}(x)\, w(x)\, dx}{\int_a^b p_{n-1}^2(x)\, w(x)\, dx} \tag{5.8}$$

y que los demás coeficientes en (24) son cero. Para los detalles puede verse Stewart (1996) [29].

Encontramos pues una *fórmula de recurrencia de tres términos* que genera una sucesión de polinomios ortogonales y mónicos asociada con el intervalo $[a, b]$ y la función de ponderación $w(x)$. Es la siguiente:

$p_0 = 1$,
$p_1 = x - d_1$,
$p_{n+1} = x p_n - d_{n+1} p_n - e_{n+1} p_{n-1},\ n = 1, 2, \ldots,$
donde d_{n+1} y e_{n+1} están dadas por (5.7) y (7.2) respectivamente.

Ya estamos listos para relacionar cuadratura de Newton-Cotes, cuadratura de Gauss y polinomios ortogonales.

5.11 Cuadratura gaussiana y polinomios ortogonales

Empezamos con un teorema sobre los ceros de p_{n+1}. Estos $n + 1$ ceros serán los nodos para la cuadratura de Gauss, de allí la importancia de este resultado.

Teorema 24 *Los ceros de* p_{n+1} *son reales, simples y pertenecen a* $[a,b]$.

La demostración puede consultarse en Stewart (1996) [29].

Este teorema establece que los ceros de p_{n+1} son una colección de $n+1$ puntos **distintos** del intervalo $[a,b]$. Queremos utilizar estos puntos como nodos para una cuadratura de Newton-Cotes asociada con el intervalo $[a,b]$ y la función de ponderación $w(x)$. Esta cuadratura aproxima $\int_a^b f(x) w(x) dx$ por una combinación lineal de valores de f de la forma

$$\sum_{j=0}^{n} A_j f(x_j).$$

Los coeficientes A_j de la cuadratura, se consiguen de forma similar a la que se usa en el teorema 5.2, a saber, definimos

$$A_j = \int_a^b l_j(x) w(x) dx, \;\; j = 0, 1, ..., n.$$

Esta cuadratura de Newton-Cotes para $\int_a^b f(x) w(x) dx$ es exacta para polinomios de grado hasta n, o sea que

$$grad(g) \leq n \Rightarrow \int_a^b g(x) w(x) dx = \sum_{j=0}^{n} A_j g(x_j). \tag{5.9}$$

Resulta que, debido a los nodos que estamos usando, esta cuadratura cumple mucho más que ésto. Demostremos que

$$grad(g) \leq 2n+1 \Rightarrow \int_a^b g(x) w(x) dx = \sum_{j=0}^{n} A_j g(x_j).$$

Sea g un polinomio de grado a lo más $2n+1$. Dividimos g por p_{n+1} y obtenemos

$$g = p_{n+1} q + r,$$

con $\text{grad}(q) \leq n$ y $\text{grad}(r) \leq n$. Luego

$$\begin{aligned}
\sum_{j=0}^{n} A_j g(x_j) &= \sum_{j=0}^{n} A_j \left[p_{n+1}(x_j)\, q(x_j) + r(x_j)\right] \\
&= \sum_{j=0}^{n} A_j r(x_j) && \text{pues } p_{n+1}(x_j) = 0 \\
&= \int_a^b r(x)\, w(x)\, dx && \text{por (5.9)} \\
&= \int_a^b \left[p_{n+1}(x)\, q(x) + r(x)\right] w(x)\, dx && \text{pues grad}(q) \leq n \\
&= \int_a^b g(x)\, w(x)\, dx.
\end{aligned}$$

Para un intervalo $[a, b]$ y una función de ponderación $w \in C([a, b])$, concluímos que:

1. Hay una sucesión de polinomios ortogonales y mónicos $\{p_j\}_{j=0}^{\infty}$ con $\text{grad}(p_j) = j$ para cada j. (Ver sección 5.10.)

2. Dado n entero positivo, el polinomio p_{n+1} tiene $n + 1$ ceros distintos en el intervalo $[a, b]$. (Ver teorema 24.)

3. Con los ceros de p_{n+1} como nodos, se puede construir la correspondiente cuadratura de Newton-Cotes para aproximar la integral $\int_a^b f(x)\, w(x)\, dx$. A dicha cuadratura se le llama *cuadratura de Gauss*. Esta cuadratura es exacta para polinomios de grado hasta $2n + 1$. Está dada por la combinación lineal

$$\sum_{j=0}^{n} A_j f(x_j),$$

con

$$A_j = \int_a^b l_j(x)\, w(x)\, dx,\ j = 0, 1, ..., n.$$

Aquí los l_j son los polinomios básicos de Lagrange asociados con los $n + 1$ nodos x_0, x_1, ..., x_n.

También podemos demostrar que los coeficientes A_j en la cuadratura de Gauss son positivos. En efecto, es claro que

$$l_j^2(x) \geq 0, \int_a^b l_j^2(x) w(x) dx > 0$$

y $\text{grad}\left(l_j^2\right) = 2n$. Por tanto, la cuadratura de Gauss es exacta para $l_j^2(x)$ y

$$0 < \int_a^b l_j^2(x) w(x) dx = \sum_{k=0}^{n} A_k l_j^2(x_k) = A_j$$

por (4.9).

Notamos $G(f, a, b, w, n)$ la cuadratura gaussiana sobre $[a, b]$, con función de ponderación $w(x)$ y con $n+1$ nodos que aproxima

$$I = \int_a^b f(x) w(x) dx.$$

Si f es suficientemente suave, el *error* cometido al utilizar esta cuadratura es

$$I - G(f, a, b, w, n) = \frac{f^{(2n+2)}(\tau)}{(2n+2)!} \int_a^b p_{n+1}^2(x) w(x) dx,$$

para algún $\tau \in (a, b)$. Una demostración de esta igualdad puede encontrarse en Kincaid y Cheney (1994) [21], sección 7.3.

5.12 Ejercicios

1. Demuestre que $\sum_{j=0}^{n} A_j = \int_a^b w(x) dx$.

2. Para intervalos y funciones de ponderación particulares, las cuadraturas de Gauss toman distintos nombres. Una de las más famosas es la cuadratura de Gauss-Legendre, que se hace con el intervalo $[-1, 1]$ y con la función de ponderación $w(x) = 1$. Los polinomios ortogonales correspondientes, se conocen como *polinomios de Legendre.* Para todas las cuadraturas de Gauss más utilizadas, los nodos y los coeficientes aparecen tabulados en Abramowitz y Stegun (1970) [1] y en compendios similares. Calcule las integrales del ejemplo 15 utilizando

las cuadraturas de Gauss-Legendre de 2 y 4 nodos, correspondientes a $n = 1$ y $n = 3$. Esta es la cuadratura que investigó Gauss originalmente.

3. Para el intervalo $[-1, 1]$ y la función de ponderación $w(x) = (1 - x^2)^{-1/2}$, la cuadratura se conoce como de Gauss-Chebyshev y a los polinomios ortogonales se les conoce como polinomios de Chebyshev. Demuestre que los coeficientes de la cuadratura de Gauss-Chebyshev son todos iguales. Encuentre el valor de ellos.

Sugerencia: Consulte Johnson y Riess (1982) [19], pag. 329.

4. Sea

$$K = \int_0^{\pi/2} \frac{dx}{\sqrt{1 - c^2 sen^2(x)}},$$

con $0 \leq c < 1$. Se sabe que para $c = 0.5$, el valor de K es 1.6858. Compruebe este hecho utilizando dos cuadraturas de Gauss distintas.

5.13 Ejercicios suplementarios

1. Aplique regla trapezoidal y de Simpson para aproximar $\mathrm{Si}(1)$, donde

$$\mathrm{Si}(z) = \int_0^z x^{-1} sen(x)\, dx.$$

Utilice $h = 2^{-k}$, con $k = 1, 2, 3$.

2. Encuentre aproximaciones al valor de la integral

$$\int_0^1 x^{-1/2} sen(x)\, dx$$

por medio de reglas compuestas trapezoidal y de Simpson, regla del rectángulo (ver ejercicio siguiente) y dos cuadraturas gaussianas diferentes. Utilice un computador o una buena calculadora.

3. **Regla del rectángulo**: sean $x_j = a + (j - 1)h$, $j = 1, 2, ..., n$ con $h = \dfrac{b - a}{n - 1}$. Entonces,

$$I = \int_a^b f(x)\, dx \cong h \sum_{j=2}^{n} f(x_j).$$

Cuando f es positiva, corresponde a aproximar la integral con la suma de las áreas de rectángulos de altura $f(x_j)$. Encuentre una expresión para el error al aproximar I con la regla del rectángulo.

4. ¿Qué tan grande debe ser n para que se cometa un error menor a 10^{-6} al aproximar $\int_0^2 e^{-x^2} dx$ con la regla trapezoidal?

5. **Regla de Simpson 3/8:** Otro esquema de integración numérica es la regla de Simpson 3/8, definida sobre tres subintervalos de la siguiente manera:

$$\int_a^{a+3h} f(x)\, dx \cong \frac{3h}{8}\left[f(a) + 3f(a+h) + 3f(a+2h) + f(a+3h)\right].$$

Frecuentemente se dice que la regla de Simpson **opaca** a la regla de Simpson 3/8. Explique la lógica de esta opinión generalizada, probando que la regla de Simpson 3/8 es exacta para polinomios de grado 3 o menor, que es lo mismo que se logra con la regla de Simpson a un costo menor en evaluaciones de funciones.

6. Encuentre aproximaciones de $\int_0^1 (1+x^2)^{-1} dx$ utilizando regla de Simpson con $h = 2^{-k}$, $k = 1, 2$. Calcule el error que comete en cada caso. Compare con las cotas de error que da la teoría.

7. Construya una regla de cuadratura de la forma

$$\int_{-1}^{1} f(x)\, dx \cong af\left(-\frac{1}{2}\right) + bf(0) + cf\left(\frac{1}{2}\right)$$

que es exacta para polinomios de grado 2 o menor.

8. Construya la cuadratura de Gauss de la forma

$$\int_{-1}^{1} f(x)\, dx \cong af(-\tau) + bf(0) + cf(\tau),$$

que es exacta para polinomios del más alto grado posible.

9. Aproxime a π por integración numérica de una integral de la forma

$$c\int_a^b \left(1+x^2\right)^{-1} dx.$$

10. Aproxime a $\log(2)$ por integración numérica de una integral de la forma
$$\int_a^b x^{-1}dx.$$

5.14 Examen de entrenamiento

Resuelva estos ejercicios sin consultar libros ni notas de clase para darse una idea del nivel de preparación que ha obtenido hasta ahora. Una calculadora sencilla es lo mínimo que debe tener a disposición y es suficiente para poder responder todos los ejercicios del examen. Claro que es preferible si dispone de una calculadora graficadora o un computador.

1. Considere la función $f(x) = |x|$ en el intervalo $[-1,1]$. Utilice $h = 2^k$, $k = -1, 0$ para aproximar $\int_{-1}^{1} f(x)\,dx$ por regla de Simpson. Compare con la solución verdadera.

2. Se quiere estimar $\int_0^{\pi} sen(x)\,dx$ por regla trapezoidal con n subdivisiones del intervalo y se desea tener un error menor que 10^{-12}. ¿Qué tan grande debe ser n de acuerdo con la teoría?

3. Utilice la cuadratura gaussiana
$$\int_{-1}^{1} f(x)\,dx \cong \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right)$$
para aproximar las integrales
$$\int_{-1}^{1} sen(3x)\,dx \text{ y } \int_1^3 \log(x)\,dx.$$

6

Ecuaciones diferenciales ordinarias

Problemas de valor inicial
Existencia y unicidad de PVIs
Método de Euler
Análisis de Error
Consistencia, estabilidad, errores de redondeo, estabilidad absoluta
Métodos de Taylor y método clásico de Runge-Kutta

6.1 Problemas de valor inicial

Los primeros problemas que consideramos son los de valor inicial, es decir, resolver numéricamente una ecuación diferencial ordinaria de primer orden para la que se conoce un punto de su curva solución. Más precisamente, sean

$$t_0 \in \mathbb{R}, \qquad y_0 = \begin{bmatrix} y_1^0 \\ y_2^0 \\ \vdots \\ y_n^0 \end{bmatrix} \in \mathbb{R}^n \text{ y}$$

$$f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n \text{ tal que } f(t,y) = \begin{bmatrix} f_1(t,y) \\ f_2(t,y) \\ \vdots \\ f_n(t,y) \end{bmatrix}.$$

Queremos encontrar una función $y : \mathbb{R} \to \mathbb{R}^n$ tal que

$$\begin{aligned} y' &= f(t,y) \\ y(t_0) &= y_0. \end{aligned} \tag{6.1}$$

La notación acostumbrada para la función incógnita y es

$$y : \mathbb{R} \to \mathbb{R}^n \text{ con } y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix} \in \mathbb{R}^n.$$

Resolver numéricamente problemas de valor inicial (PVI, de ahora en adelante), es uno de los temas en la mayoría de los libros de análisis numérico. Excelentes presentaciones del tema pueden encontrarse, por ejemplo, en Stoer y Bulirsch (1992) [30], Kincaid y Cheney (1994) [21] y Johnson y Riess (1982) [19]. Los libros Gear (1971) [11] y Henrici (1962) [14] son dos de los primeros que se han escrito sobre el tema y se consideran clásicos.

6.2 Problemas lineales y no lineales

Si cada componente f_k de f es de la forma

$$f_k(t, y) = a_{k0}(t) + \sum_{j=1}^{n} a_{kj}(t)\, y_j,$$

con $a_{kj} : \mathbb{R} \to \mathbb{R}$, $k = 1, ..., n$, $j = 0, 1, ..., n$, decimos que (6.1) es un problema lineal. En otro caso, decimos que el problema (6.1) es no lineal. Si a_{k0} es la función nula para todo k, decimos que (6.1) es un problema homogéneo. En otro caso, decimos que es un problema no homogéneo.

Los siguientes ejemplos resueltos sirven como motivación para conocer herramientas modernas de solución disponibles en MATLAB. Estas herramientas utilizan tamaños de paso adaptable y métodos de alto orden. Como se verá después, nuestra pretensión en estas notas es considerar en detalle solamente los métodos más elementales. Sin embargo, confiamos en que quienes se sientan inclinados por el tema, puedan basarse en este material y en las referencias que incluimos para analizar métodos más complicados.

Ejemplo 25 *1. Sean* $A = \begin{bmatrix} -30 & 28 \\ 0 & -2 \end{bmatrix}$ *y* $y_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. *Encontrar* $y : \mathbb{R} \to \mathbb{R}^2$ *tal que*

$$\begin{aligned} y' &= Ay \\ y(0) &= y_0. \end{aligned}$$

Este es un problema lineal homogéneo y su solución exacta es fácil de obtener. Pero también es interesante resolverlo numéricamente.
2. Encontrar $y : \mathbb{R} \to \mathbb{R}^2$ *tal que*

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -\frac{g}{b}\sin(y_1) \\ y(0) &= \begin{bmatrix} \frac{\pi}{8} \\ 0 \end{bmatrix}. \end{aligned}$$

Aqui, g y b son constantes positivas. Este es un problema no lineal homogéneo que sirve de modelo para un péndulo plano.
3. Especies en competencia: Sean $y_1(t)$ *y* $y_2(t)$ *las cantidades de animales en el tiempo t de dos especies que compiten por el mismo alimento. Se supone que la natalidad de cada especie es proporcional al número de animales de esa especie pero la mortalidad de cada especie depende de la población de ambas especies. Un sistema de este tipo es el siguiente:*

$$\begin{aligned} y_1' &= y_1(t)(4 - 0.0003y_1(t) - 0.0004y_2(t)) \\ y_2' &= y_2(t)(2 - 0.0002y_1(t) - 0.0001y_2(t)) \end{aligned} \tag{6.2}$$

Si se sabe que la población inicial de cada especie es de 10000, cuál será la solución de este sistema para $t \in [0, 4]$?
Este es otro ejemplo de problema no lineal homogéneo.

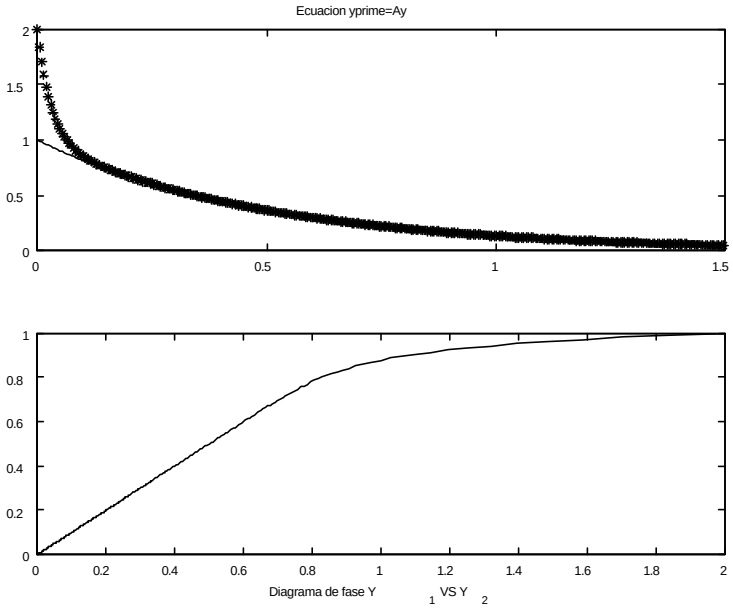
6.3 Soluciones de los ejemplos

1. Para el primer ejemplo, usamos el método de Euler por medio del M-archivo ode3.m.

```
% ode3, Herramienta didactica, Carlos E. Mejia, 2002
% uso: ode3
```

```
% solucion numerica de PVI, metodo de Euler
% rhs3: m-archivo con lado derecho en el pvi
% inte: intervalo [to,tfin]
% ci: condicion inicial y(to)
clear all
inte=[0 4];ci=[2;1];
t=linspace(inte(1),inte(2),800);
y=zeros(length(t),2);h=(inte(2)-inte(1))/length(t);
y(1,:)=ci(:)';
for j=1:length(t)-1
 y(j+1,:)=y(j,:)+h*feval('rhs3',t,y(j,:)')';
end
y1=y(:,1);
y2=y(:,2);
subplot(2,1,1);
plot(t,y1,'k*',t,y2,'k');
%set(gca,'XTick',[0 1/2 1 3/2])
axis([0 3/2 0 2]);ylabel('Y(1) y Y(2)')
title('Ecuacion yprime=Ay');
subplot(2,1,2);
plot(y1,y2,'r');xlabel('Y(1)');ylabel('Y(2)');
xlabel('Diagrama de fase Y_1 VS Y_2');
print -deps2 .\fig\ode3.eps
```

Las gráficas generadas son:

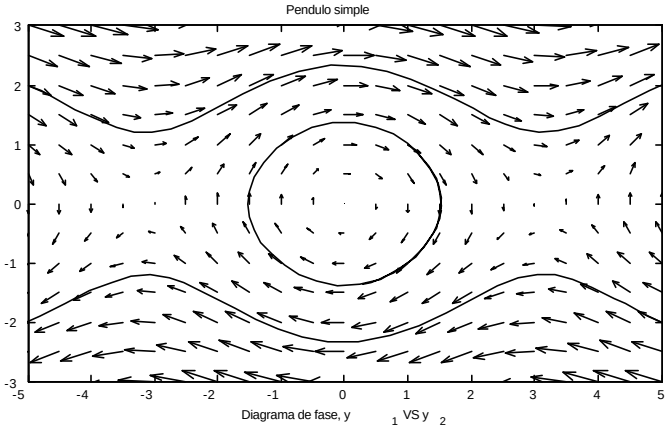


2. El segundo ejemplo se resuelve con una herramienta profesional de MATLAB, llamada ODE45. Esta es la más poderosa entre todas las rutinas de uso general que ofrece MATLAB, para la solución de PVIs. El M-archivo que utilizamos es éste:

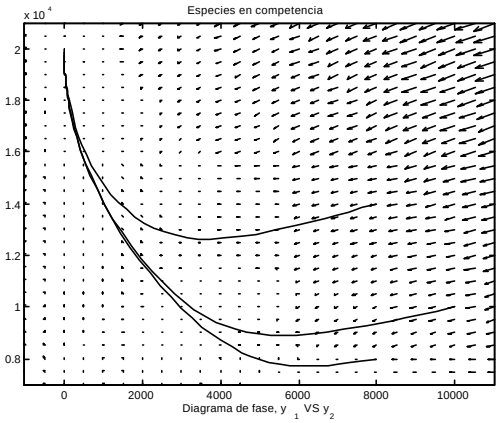
```
% ode1, Herramienta didactica, Carlos E. Mejia, 2002
% Referencia D. Higham y N. Higham, Matlab Guide
% SIAM, p. 151
tspan=[0,10];
yazero=[1;1];ybzero=[-5;2];yczero=[5;-2];
[ta,ya]=ode45('rhs1',tspan,yazero);
[tb,yb]=ode45('rhs1',tspan,ybzero);
[tc,yc]=ode45('rhs1',tspan,yczero);
[y1,y2]=meshgrid(-5:.5:5,-3:.5:3);
Dy1Dt=y2;Dy2Dt=-sin(y1);
quiver(y1,y2,Dy1Dt,Dy2Dt,'k')
hold on
plot(ya(:,1),ya(:,2),'k',...
yb(:,1),yb(:,2),'k',yc(:,1),yc(:,2),'k')
```

```
axis equal,axis([-5 5 -3 3])
title('Pendulo simple')
xlabel ('Diagrama de fase, y_1 VS y_2'), hold off
print -deps2 .\fig\ode1.eps
```

Con este M-archivo generamos la siguiente gráfica:



3. Con un M-archivo como el anterior, generamos esta gráfica para el ejemplo 3.



La gráfica sugiere la tendencia a desaparecer de la población 1 y a estabilizarse en 20.000 la otra población. Para demostrar un hecho

como éste, se recurre al estudio de equilibrios para el sistema dinámico (6.2).

6.4 Análisis del Método de Euler

Al método de Euler lo escogimos para hacerle un análisis detallado por ser el más sencillo y porque dicho análisis contiene muchas de las ideas utilizadas para el análisis de métodos menos sencillos. La sección la iniciamos con un teorema de existencia y unicidad para PVIs seguido por ejemplos ilustrativos. Después nos concentramos en el algoritmo de Euler y exponemos las definiciones y estimados básicos para el análisis de error del método. Terminamos con la presentación esquemática de otros métodos de diferencias finitas. Por razones metodológicas, el análisis lo realizamos para ecuaciones escalares, pero casi siempre la generalización al caso vectorial es prácticamente una repetición del mismo análisis.

6.5 Existencia y unicidad

Empezamos con el Teorema Fundamental de Existencia y Unicidad de Soluciones.

Teorema 26 *Sea f definida y continua en la franja*

$$S = \{(t, y) \mid a \leq t \leq b, y \in \mathbb{R}^n\},$$

a, b finitos. Además, supongamos que existe una constante L tal que

$$\|f(t, u) - f(t, v)\| \leq L \|u - v\|$$

para todo $t \in [a, b]$ y todo $u, v \in \mathbb{R}^n$ (condición de Lipschitz). Entonces para todo $t_0 \in [a, b]$ y todo $y_0 \in \mathbb{R}^n$ existe exactamente una función $y(t)$ tal que
a. $y(t)$ es continua y continuamente diferenciable para $t \in [a, b]$;
b. $y'(t) = f(t, y(t))$ para $t \in [a, b]$;
c. $y(t_0) = y_0$.

Dos ejemplos unidimensionales son apropiados en este momento.

Ejemplo 27 *1. El PVI*

$$y' = y, \quad y(0) = 1$$

tiene una única solución $y(t) = \exp(t)$. *Las hipótesis del teorema anterior se satisfacen en cualquier franja* S *que contenga al eje* y *en el plano cartesiano* $\mathbb{R} \times \mathbb{R}$.
2. El PVI

$$y' = y^2, \quad y(0) = 1 \tag{6.3}$$

tiene una única solución $y(t) = \frac{1}{1-t}$, *definida únicamente para* $t < 1$. *Esta asíntota vertical no sorprende, pues de (6.3) vemos que la función solución del PVI debe ser positiva con derivada positiva y creciente. En este caso, la existencia y unicidad de la solución no se puede obtener del Teorema 26. Recomendamos un libro especializado como Henrici (1962) [14] para más detalles teóricos.*

6.6 Solución numérica

Los métodos de diferencias finitas para el PVI (6.1), o simplemente, métodos de diferencias, consisten en encontrar aproximaciones de la solución $y(t)$ en un conjunto de puntos

$$t_0 < t_1 < t_2 < ... < t_k < ...$$

no necesariamente igualmente espaciados. Denotemos por

$$Y_0, Y_1, ..., Y_k, ...$$

a las correspondientes aproximaciones. Los cálculos para hallar Y_k se basan en una o varias de las aproximaciones previamente encontradas $Y_0, Y_1, ..., Y_{k-1}$.

Al número de aproximaciones previas que se requieren en un método dado, se le llama el *número de pasos* del método.

Si un método numérico contempla la posibilidad de cambiar de tamaño de paso de acuerdo con los resultados que va obteniendo, se dice que se trata de un *método adaptativo*. La mayoría de los métodos de diferencias programados como comandos en los principales paquetes de

software, son adaptativos. Varios de los que utiliza MATLAB son de la familia de métodos llamados de *Runge-Kutta-Fehlberg*. Sugerimos consultar Kincaid y Cheney (1994) [21] a quien desee conocer más detalles sobre estos métodos.

6.7 Método de Euler

Como la definición de métodos de diferencias finitas es esencialmente independiente del número n de funciones incógnitas, en adelante nos limitamos a presentar la teoría para una sola ecuación con una sola función incógnita. En los ejemplos, consideramos tanto ecuaciones escalares como sistemas.

Sea $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Queremos encontrar una función $y : \mathbb{R} \to \mathbb{R}$ tal que

$$\begin{aligned} y' &= f(t, y) \\ y(t_0) &= y_0. \end{aligned} \tag{6.4}$$

Para las consideraciones siguientes, supondremos que este PVI tiene *una única solución suficientemente suave,* es decir, que es *única* y tiene todas las derivadas que se requieran para el análisis. Aunque parezca muy restrictiva, esta condición únicamente facilita el enunciado de los métodos numéricos de interés y no se requiere a la hora de hacer cálculos específicos.

Los métodos de diferencias finitas que estudiamos a continuación son de un paso y trabajan con una malla igualmente espaciada, o sea $t_j = t_0 + jh$, para todo $j = 1, 2, ...$, donde h es un número real no nulo.

El primero que consideramos es el más sencillo de todos y se obtiene de la expansión de Taylor de primer orden

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2} y''(\xi), \tag{6.5}$$

donde ξ está entre t y $t+h$ y recordamos que $y'(t) = f(t, y(t))$. Si $0 < |h|$, una primera aproximación para $y(t+h)$ es $y(t) + hf(t, y(t))$. Esto da origen al *método de Euler* para aproximar la solución de (6.4)

que se define así:

$$\begin{array}{l} Y_0 = y_0 \text{ y para } j = 0, 1, ..., \text{ hacemos} \\ Y_{j+1} = Y_j + hf\left(t_j, Y_j\right) \\ t_{j+1} = t_j + h. \end{array}$$

Nótese que cada iteración se obtiene de la anterior agregando un término corrector de la forma $hG\left(t_j, Y_j; h; f\right)$. En el caso del método de Euler que acabamos de definir, G es independiente de h, pues es $f\left(t, Y\right)$. En general, los métodos de un paso se definen así:

$$\begin{array}{l} Y_0 = y_0 \text{ y para } j = 0, 1, ..., \text{ hacemos} \\ Y_{j+1} = Y_j + hG\left(t_j, Y_j; h; f\right) \\ t_{j+1} = t_j + h. \end{array} \tag{6.6}$$

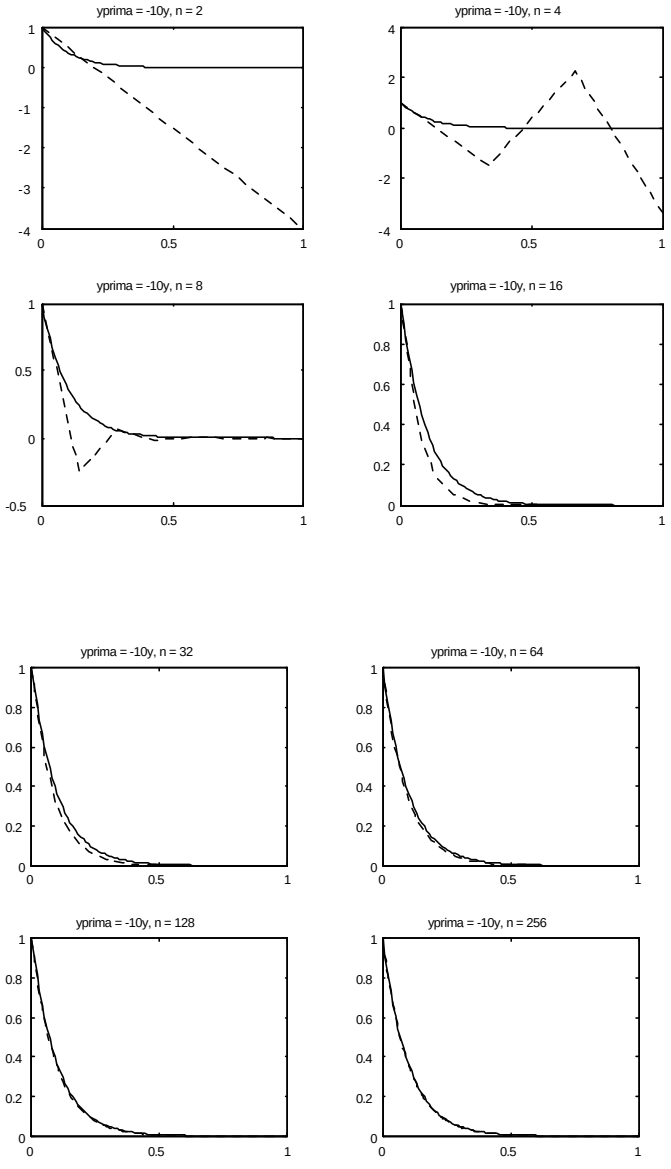
Al final de la sección vemos algunos otros métodos de un paso.

Ejemplo 28 *Resolvamos el PVI*

$$\begin{array}{rcl} y' & = & -10y \\ y\left(0\right) & = & 1 \end{array}$$

por el método de Euler. Interesa aproximar el valor $\exp(-10) = y\left(1\right)$.

Veamos algunas aproximaciones obtenidas con distintos valores de h. La solución exacta es la línea continua y la otra es la solución aproximada.


yprima = -10y, n = 2
yprima = -10y, n = 4
yprima = -10y, n = 8
yprima = -10y, n = 16
yprima = -10y, n = 32
yprima = -10y, n = 64
yprima = -10y, n = 128
yprima = -10y, n = 256

La siguiente tabla corrobora lo observado en las gráficas.

Tabla 1: Método de Euler, $y' = -10y$, $y(0) = 1$			
r	h	$\exp(-10) - Y_n$	$(\exp(-10) - Y_n)/h$
1	5.0000e-001	4.0000e+000	8.0001
2	2.5000e-001	3.3750e+000	13.5002
3	1.2500e-001	1.0644e-004	0.0009
4	6.2500e-002	4.4992e-005	0.0007
5	3.1250e-002	3.6375e-005	0.0012
6	1.5625e-002	2.2937e-005	0.0015
7	7.8125e-003	1.2790e-005	0.0016
8	3.9063e-003	6.7400e-006	0.0017

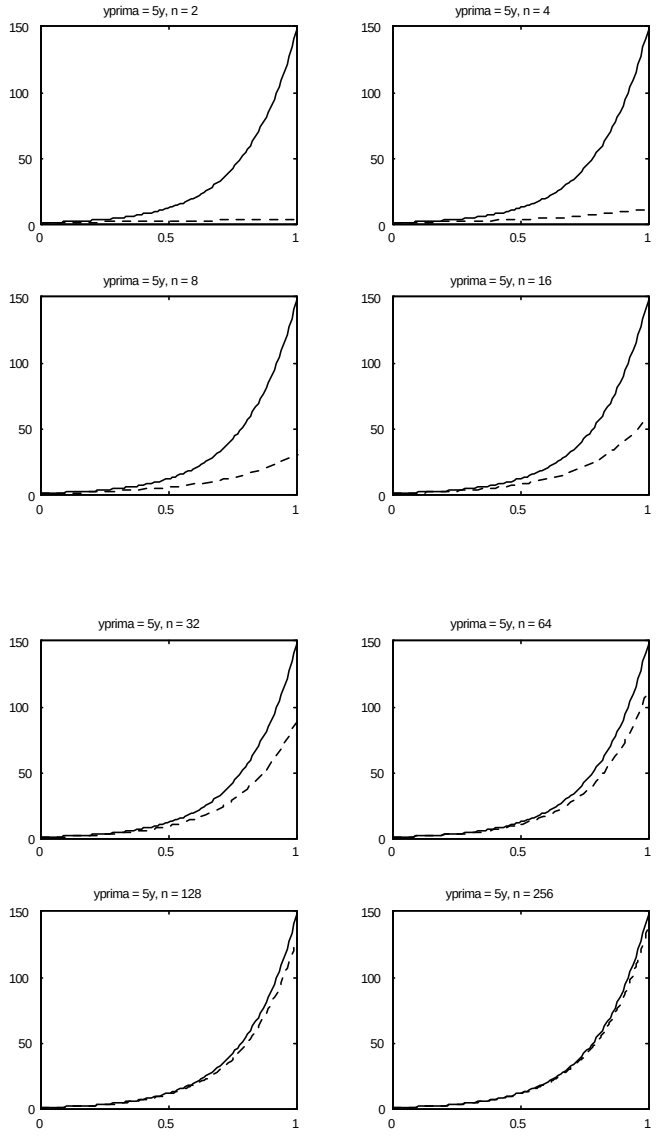
Notamos que hay valores de h para los cuales la aproximación es inaceptable y que la última columna de la tabla tiende a estabilizarse. Lo primero indica que h debe ser suficientemente pequeña y lo segundo sugiere que el orden del método es 1. Pero estas son solo observaciones que más tarde precisamos con claridad. Por ahora, consideramos un ejemplo muy parecido al anterior.

Ejemplo 29 *Resolver* $y' = 5y$, $y(0) = 1$.

La tabla 2 contiene resultados para este ejemplo, que fueron obtenidos con un M-archivo muy similar al del ejemplo 28. Esta tabla contiene una columna de error relativo debido a lo difícil que es, en general, entender la calidad de una aproximación con el error absoluto como único estimado.

Tabla 2: Método de Euler, $y' = 5y$, $y(0) = 1$				
r	h	Er. absoluto	Er. relativo	$(\exp(-10) - Y_n)/h$
1	5.0000e-001	144.9132	0.9764	289.8263
2	2.5000e-001	137.0225	0.9233	548.0901
3	1.2500e-001	118.4923	0.7984	947.9387
4	6.2500e-002	89.3265	0.6019	1429.2235
5	3.1250e-002	58.3382	0.3931	1866.8221
6	1.5625e-002	34.0847	0.2297	2181.4207
7	7.8125e-003	18.5481	0.1250	2374.1591
8	3.9063e-003	9.6934	0.0653	2481.5193

Las correspondientes figuras ilustran mejor lo que está pasando: como la solución exacta *estalla*, o sea, tiende a infinito, también debe hacerlo la solución aproximada. Se observa de nuevo que h debe ser *suficientemente pequeño* y se sigue la misma convención: La línea a trozos identifica la solución aproximada y la continua es la exacta.



6.7.1 Análisis de error

El análisis de error para métodos de un paso, y muy en especial, para el método de Euler, es nuestro siguiente punto a tratar. Los ejemplos anteriores, aunque sencillos, plantean inquietudes con respecto al tamaño de h, que aparecen también en ejemplos complejos y en otros métodos de solución.

Sea $y(t)$ alguna solución de $y' = f(t, y)$ y sean t^* y h fijos. Definimos el *error de truncación local en* t^*, denotado τ^*, por la ecuación

$$y(t^* + h) = y(t^*) + hG(t^*, y(t^*); h; f) + h\tau^*. \tag{6.7}$$

La discrepancia entre $y(t^* + h)$ y $y(t^*) + hG(t^*, y(t^*); h; f)$ está dada por el término $h\tau^*$. Esta última expresión corresponde a la fórmula de avance que propone el método numérico (6.6).

En el caso del método de Euler, la correspondiente expresión (6.7) es

$$y(t^* + h) = y(t^*) + hf(t^*, y(t^*)) + h\tau_e.$$

Por teorema de Taylor,

$$y(t^* + h) = y(t^*) + hf(t^*, y(t^*)) + \frac{1}{2}h^2 y''(\sigma), \tag{6.8}$$

donde σ está entre t^* y $t^* + h$. Es decir, $\tau_e = \frac{1}{2}hy''(\sigma)$.

En este punto es conveniente introducir la notación O, que se lee *O grande.*

Definición 30 *Si $F(t, h)$ es una función definida para $t_0 \leq t \leq b$ y para todo h suficientemente pequeño, entonces la notación*

$$F(t, h) = O(h^r)$$

para algún $r > 0$ significa que hay una constante c tal que

$$|F(t, h)| \leq ch^r.$$

La notación *O grande* se usa especialmente para comparar funciones o sucesiones. Podemos decir que si

$$F(t, h) = O(h^r),$$

entonces $F(t,h)$ converge a 0 cuando $h \to 0$ al menos tan rápido como h^r. Entre sucesiones también es útil. Por ejemplo, puesto que $\frac{n+1}{n^2} \leq 2\left(\frac{1}{n}\right)$, entonces

$$\frac{n+1}{n^2} = O\left(\frac{1}{n}\right).$$

Esto implica que $\frac{n+1}{n^2}$ converge a 0 cuando $n \to \infty$ al menos tan rápido como $\frac{1}{n}$.

De acuerdo con la definición anterior y recordando la suposición sobre y de tener un número suficiente de derivadas, podemos decir que el error de truncación local en el método de Euler cumple $\tau_e = O(h)$.

En general para métodos de un paso, si $\tau = O(h^p)$, se dice que el método es de *orden p*.

Nótese que esta clasificación depende de la suavidad de las soluciones para $y' = f(t,y)$. Puede suceder que un método de orden p para un problema $y' = g(t,y)$ no se comporte como tal por ser g una función sin la suavidad requerida. Algo análogo sucede con el método de Newton para resolver numéricamente ecuaciones de la forma $F(x) = 0$. Aunque el método se clasifica como de *convergencia cuadrática*, no siempre exhibe esa clase de convergencia.

Volvamos al análisis de error del método de un paso (6.6). Denotemos $e_j = y(t_j) - Y_j$ el *error global* en t_j y para $h = \frac{b-t_0}{n}$, sea $\tau = \max_{j=1,2,...,n} |\tau_j|$, donde τ_j es el error de truncación local en $t_j = t_0 + jh$, $j = 1, 2, ..., n$.

Teorema 31 *Supongamos que G satisface la condición de Lipschitz*

$$|G(t,u;h;f) - G(t,v;h;f)| \leq K|u-v|$$

para todo h que cumple $0 < h \leq b - t_0$, para todo $t \in [t_0, b]$ y para todo u, v, $-\infty < u, v < \infty$. Entonces

$$|e_j| \leq \frac{\tau}{K}\left[e^{K(t_j - t_0)} - 1\right].$$

Una demostración detallada de este teorema puede encontrarse en Johnson y Riess (1982) [19].

La cota del lado derecho es en general imposible de conocer por la dificultad para aproximar a τ y K. Además, a menudo no es pequeña, a no ser que h sea demasiado pequeña. Por tanto, la información que proporciona este teorema es eminentemente cualitativa y no es muy útil a la hora del cálculo numérico.

Nótese que en el ejemplo 28, $K = 10$ y $\tau \leq 50h$. El teorema anterior asegura que

$$|e_j| \leq 5h\left[e^{10} - 1\right],$$

lo cual no dice mucho.

6.7.2 Consistencia

Para continuar con el análisis de error, consideremos de nuevo el Teorema 31. Allí se afirma que un método de un paso es convergente si se cumple la *condición de consistencia*, que dice simplemente

$$\tau \to 0 \text{ cuando } h \to 0. \tag{6.9}$$

Pero hay otra forma de enunciar la condición de consistencia: de (6.7) vemos que

$$y\left(t_{j+1}\right) = y\left(t_j\right) + hG\left(t_j, y\left(t_j\right); h; f\right) + h\tau_j,$$

donde $h\tau_j$ es el error de truncación local en t_j. Esta expresión la reescribimos así:

$$\frac{y\left(t_{j+1}\right) - y\left(t_j\right)}{h} = G\left(t_j, y\left(t_j\right); h; f\right) + \tau_j.$$

Como $y'\left(t_j\right) = f\left(t_j, y\left(t_j\right)\right)$, vemos que, tomando límites, la condición de consistencia se puede expresar simplemente como

$$G\left(t_j, y\left(t_j\right); 0; f\right) = f\left(t_j, y\left(t_j\right)\right).$$

Volveremos a considerar consistencia al final de la sección, cuando veamos otros métodos de diferencias. Por ahora continuemos con el análisis.

6.7.3 Estabilidad

Pasemos ahora, basados en Atkinson (1978) [4], a enunciar un estimado de *estabilidad* para el método de Euler. En pocas palabras podemos decir que estabilidad es continuidad con respecto a los datos.

Definición 32 *Consideramos un PVI*

$$\begin{aligned} y' &= f(t,y) \\ y(t_0) &= y_0 \end{aligned}$$

y una perturbación del mismo dada por

$$\begin{aligned} z' &= f(t,z) + \delta(t) \\ z(t_0) &= y_0 + \varepsilon, \end{aligned}$$

donde $|\varepsilon| \leq \rho$ *y* $\|\delta\|_\infty \leq \rho$, *para algún* ρ. *Decimos que el método de un paso con fórmula de avance*

$$Y_{j+1} = Y_j + hG(t_j, Y_j; h; f) \ \ con\ Y_0 = y_0 \tag{6.10}$$

es estable, si las soluciones Y *y* Z *de (6.10) y*

$$Z_{j+1} = Z_j + hG(t_j, Z_j; h; f(t_j, Z_j) + \delta(t_j)) \ \ con\ Z_0 = y_0 + \varepsilon$$

respectivamente, cumplen

$$\lim_{\rho \to 0} \|Y - Z\| = 0.$$

Debe advertirse que hay varias clases de estabilidad. Para el método de Euler, chequear que se cumple esta definición es bastante rápido.

Las fórmulas de avance del método de Euler son, respectivamente,

$$Y_{j+1} = Y_j + hf(t_j, Y_j)$$

y

$$Z_{j+1} = Z_j + h[f(t_j, Z_j) + \delta(t_j)],$$

donde

$$Y_0 = y_0,\ Z_0 = y_0 + \varepsilon \text{ y } j = 0, 1, 2, ..., N-1.$$

Sea $e_j = Z_j - Y_j$. Esta diferencia satisface el siguiente estimado:

$$\max_{0 \leq j \leq N} |e_j| \leq e^{(b-t_0)K} |\varepsilon| + \frac{\|\delta\|_\infty}{K} \left[e^{(b-t_0)K} - 1\right].$$

Claramente, si $\rho \to 0$, entonces $e_j \to 0$ para todo j, es decir, el método de Euler es estable.

6.7.4 Errores de redondeo

Otra cosa es *propagación de error de redondeo.* Cómo se comporta el método de Euler al respecto? Continuamos siguiendo a Atkinson (1978) [4]. Supongamos que y_0 se conoce exactamente. Teniendo en cuenta errores de redondeo, el método de Euler para (6.4) tiene la forma

$$U_{j+1} = U_j + hf\left(t_j, U_j\right) + \varepsilon_j,$$

donde $U_0 = y_0$ y $j = 0, 1, ..., N$. Recordemos que el error de truncación en cada paso se define por la ecuación

$$y\left(t_{j+1}\right) = y\left(t_j\right) + hf\left(t_j, y\left(t_j\right)\right) + h\tau_j$$

y denotemos $E_j = U_j - y\left(t_j\right)$ el error total cometido en el paso j. Si

$$\tau = \max_{0 \le j \le N} |\tau_j| \text{ y } \varepsilon = \max_{0 \le j \le N} |\varepsilon_j|,$$

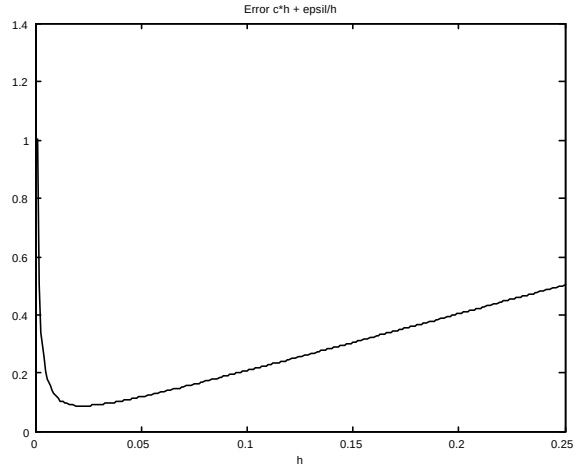
entonces el error E_j satisface

$$|E_j| \le \left[\frac{1}{K}\right]\left[e^{(t_j - t_0)K} - 1\right]\left[\tau + \frac{\varepsilon}{h}\right],$$

que, de acuerdo con (6.8) y con la hipótesis de suficiente suavidad para y, tiene la forma

$$|E_j| \le C_1\left[C_2 h + \frac{\varepsilon}{h}\right], \tag{6.11}$$

con C_1 y C_2 constantes positivas. Muy frecuentemente, la gráfica del lado derecho de (6.11) tiene la siguiente forma:



En esta gráfica se condensa una de las precauciones más importantes que se deben tener en el uso de métodos numéricos: Se quiere h pequeño para que τ sea pequeño pero no muy pequeño para que $\frac{\varepsilon}{h}$ no sea demasiado grande. Situaciones como ésta se presentan siempre que se resuelve un problema en un computador. Tomar a h demasiado pequeño obliga a hacer muchas iteraciones que invitan al error de redondeo a ser protagonista en los cálculos.

6.7.5 Estabilidad absoluta

Muy relacionado con lo que estamos tratando es el concepto de *estabilidad absoluta,* que en pocas palabras, significa que el método numérico no aumenta errores pasados. Para estudiar la estabilidad absoluta se recurre a la ecuación de prueba

$$y' = \lambda y. \tag{6.12}$$

El método de Euler aplicado a esta ecuación tiene la forma

$$Y_{j+1} = Y_j \left(1 + h\lambda\right),$$

lo cual lleva por iteración, a la ecuación

$$Y_j = Y_0 \left(1 + h\lambda\right)^j.$$

Es decir, de la única manera que el método de Euler no aumenta errores iniciales es pidiendo

$$|1 + h\lambda| < 1. \tag{6.13}$$

Al intervalo más grande de la forma $(a, 0)$ tal que si $h\lambda \in (a, 0)$ entonces se cumple (6.13), se le llama *intervalo de estabilidad absoluta.* Para el método de Euler, es, por supuesto, $(-2, 0)$. Terminamos con unos comentarios:

1. El método de Euler no puede ser absolutamente estable cuando $\lambda \geq 0$.

2. Si $\lambda < 0$, entonces $0 < h < -\frac{2}{\lambda}$. Pero de acuerdo con el apartado anterior, h no puede tomarse demasiado pequeño. Por consiguiente,

existe un intervalo de valores positivos en el que h debe situarse para obtener la mejor aproximación por medio del método numérico. Este es un enunciado de tipo práctico. Los enunciados teóricos exigen considerar $h \to 0$.

3. Los problemas escalares del tipo (6.12) son el caso 1×1 de los problemas lineales

$$y' = Ay,$$

donde A es una matriz. Estos problemas tienen interés por si mismos pero además, surgen cada que se quiere considerar la linearización de problemas no lineales, que es una herramienta muy valiosa a la hora de estudiar equilibrios de sistemas dinámicos. Sugerimos Hale y Kocak (1991) [13] para profundizar sobre el tema.

Terminamos esta sección con la forma de construir otros métodos de Taylor y con la definición del método clásico de Runge-Kutta.

6.8 Métodos de Taylor

Los *métodos de un paso* para la solución de (6.4) se definieron en (6.6). Hay muchos métodos de un paso, por ejemplo, a partir del teorema de Taylor, se pueden definir otros si la expansión (6.5) se reemplaza por una de más alto orden.

La expansión de Taylor de orden 2 es

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \frac{h^3}{3!}y'''(\eta),$$

donde η está entre t y $t+h$. Para la escritura de y'' en términos de f, se utiliza la regla de la cadena. De aquí se deduce el método de un paso conocido como método de Taylor de orden 2, que se define como en (6.6) con

$$G(t_j, Y_j; h; f) = f(t_j, Y_j) + \frac{h}{2}\left(f_t(t_j, Y_j) + f_y(t_j, Y_j) f(t_j, Y_j)\right).$$

Los subíndices t y y indican derivadas parciales con respecto a la primera y la segunda variables de f respectivamente.

El inconveniente principal que tienen los métodos de Taylor, es la necesidad de evaluar a f y a varias de sus derivadas parciales en cada iteración. Actualmente esa es una tarea menos dispendiosa gracias a los paquetes de software que permiten computación simbólica. Pero todavía el tiempo para hacer un cálculo numérico es considerablemente menor que el de hacer el mismo cómputo con cálculo simbólico.

6.9 Métodos de Runge-Kutta

Los *métodos de Runge-Kutta* son métodos de un paso que en lugar de requerir evaluaciones de derivadas parciales de f, lo que requieren es evaluaciones de f en puntos *cuidadosamente escogidos*. El más famoso de todos los métodos de Runge-Kutta es llamado *clásico*. Se define por medio de (6.6) con

$$G\left(t_j, Y_j; h; f\right) = \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right],$$

donde

$$\begin{aligned} k_1 &= f\left(t_j, Y_j\right) \\ k_2 &= f\left(t_j + \frac{1}{2}h, Y_j + \frac{1}{2}hk_1\right) \\ k_3 &= f\left(t_j + \frac{1}{2}h, Y_j + \frac{1}{2}hk_2\right) \\ k_4 &= f\left(t_j + h, Y_j + hk_3\right). \end{aligned}$$

La deducción de métodos de Runge-Kutta que requieren únicamente dos evaluaciones de la función f, es un ejercicio que recomendamos. Aparece en muchos libros, por ejemplo, Kincaid y Cheney (1994) [21] o Stoer y Bulirsch (1992) [30].

6.10 Ejercicios

1. Comprobar que el método de Taylor orden 2 y el método Runge-Kutta clásico satisfacen la condición de consistencia (6.9).

2. Demostrar que si f satisface la condición de Lipschitz del Teorema 26, entonces las funciones G de los métodos mencionados en el ejercicio anterior satisfacen la condición de Lipschitz del Teorema 31.

6.11 Problemas con valores en la frontera

Los problemas con valores en la frontera en dos puntos, PVF en lo sucesivo, son más complicados que los problemas de valor inicial, entre otras cosas porque los teoremas de existencia y unicidad son más elaborados y menos generales. Entre los investigadores del tema, destacamos a Keller, quien ha propuesto varios resultados importantes, muchos de los cuales están consignados en su libro Keller (1990) [20]. Un compendio actualizado hasta fines de los 80's es Ascher, Mattheij y Russell (1995) [3] que también recomendamos a quienes deseen profundizar en el tema.

Un PVF se define así: Encontrar una solución $y(t)$ de un sistema de n ecuaciones diferenciales ordinarias,

$$y' = f(t,y),\ y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},\ f(t,y) = \begin{bmatrix} f_1(t,y) \\ f_2(t,y) \\ \vdots \\ f_n(t,y) \end{bmatrix}$$

que satisfacen una condición de borde de una de las siguientes clases:

$$r(y(a), y(b)) = 0,$$

en la que la función r puede ser no lineal en una o ambas de sus dos componentes,

$$Ay(a) + By(b) = c,$$

donde A y B son matrices de orden n y $c \in \mathbb{R}^n$ o finalmente,

$$A_1 y(a) = c_1,\ A_2 y(b) = c_2, \tag{6.14}$$

donde A_1 y A_2 son matrices de n columnas y c_1, c_2 son vectores con el mismo número de filas de A_1 y A_2 respectivamente. A las condiciones de borde (6.14) se les llama *separadas.* En todos los casos, $-\infty < a < b < \infty$.

En estas notas, consideramos únicamente funciones y y f de valor real y condiciones de borde separadas. Es decir, nos restringimos a trabajar con ecuaciones diferenciales ordinarias de segundo orden de la forma

$$y''(t) = f(t, y), \qquad -\infty < a \leq t \leq b < \infty \tag{6.15}$$

junto con condiciones de borde

$$y(a) = A, \qquad y(b) = B, \tag{6.16}$$

donde A y B son constantes. La función f puede depender de forma lineal o no lineal de su segunda variable.

Lo primero que hacemos es una ilustración de las dificultades que se pueden presentar en cuanto a existencia y unicidad. Dos ejemplos sencillos de PVF con la forma (6.15) - (6.16) son:

$$\begin{aligned} &y'' = y \\ &y(0) = A,\ y(1) = B \end{aligned}$$

y

$$\begin{aligned} &y'' = -y \\ &y(0) = A,\ y(\pi) = B. \end{aligned} \tag{6.17}$$

En el primer caso, la única solución es

$$y\left(t\right) = C_1 e^t + C_2 e^{-t},$$

donde las constantes C_1 y C_2 son las componentes del vector solución del sistema

$$\begin{bmatrix} 1 & 1 \\ e & e^{-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix}.$$

En el segundo caso, no siempre hay solución y cuando hay, no es única, pues las soluciones son de la forma

$$y\left(t\right) = C_1 sen\left(t\right) + C_2 \cos\left(t\right),$$

pero al aplicar condiciones de borde, se llega a $A = C_2 = -B$ y no aparecen restricciones para C_1. Por tanto, las soluciones de (6.17) con $A = -B$, son

$$y\left(t\right) = C_1 sen\left(t\right) + A \cos\left(t\right),$$

con C_1 cualquier constante y si $A \neq -\ B$ no hay solución.

6.12 Diferencias finitas y método de colocación para PVFs

Existencia y unicidad
Diferencias finitas
Método de Newton
Colocación

En esta sección presentamos un teorema de existencia y unicidad, debido a Keller, y explicamos los métodos numéricos más sencillos para la solución de PVFs. La exposición la basamos en Kincaid y Cheney (1994) [21] y Stoer y Bulirsch (1992) [30]. Además, en algunos apartes, sobre todo en lo que tiene que ver con M-archivos y ejemplos, seguimos a López (2000) [22].

6.13 Existencia y unicidad

Decimos que el PVF (6.15) - (6.16) es de clase M si la función $f(t, y)$ cumple las siguientes condiciones:

a. $f(t, y)$ está definida y es continua en la franja $[a, b] \times \mathbb{R}$.

b. (Condición de Lipschitz) Existe una constante L tal que para todo $t \in [a, b]$ y cualquier par de números u_1 y u_2,

$$|f(t, u_1) - f(t, u_2)| \leq L |u_1 - u_2| .$$

c. $f_y(t, y)$ es continua y no negativa en la franja $[a, b] \times \mathbb{R}$.

Teorema 33 *(Keller) Todo PVF de clase M tiene una única solución.*

Nótese que las condiciones a. y b. de arriba son las mismas que se requieren en el teorema fundamental de existencia y unicidad de los PVI que estudiamos antes.

6.14 Diferencias finitas

Una posible discretización de diferencias finitas para este problema es como sigue: Se genera una subdivisión del intervalo $[a, b]$, en $n + 1$ subintervalos de igual longitud $h = \dfrac{b - a}{n + 1}$, de manera que los nodos

de la malla sean $t_j = a + jh$, con $j = 0, 1, ..., n+1$. Enseguida para $j = 1, 2, ..., n$, aproximamos la segunda derivada $y''(t_j)$ por el cociente

$$\frac{1}{h^2}\left(y(t_{j-1}) - 2y(t_j) + y(t_{j+1})\right)$$

y entonces resulta natural querer resolver el sistema

$$\frac{1}{h^2}\left(v_{j-1} - 2v_j + v_{j+1}\right) = f\left(t_j, v_j\right), \qquad j = 1, 2, ..., n$$

con $v_0 = A$, $v_{n+1} = B$ y para $j = 1, ..., n$, v_j son las aproximaciones de los valores $y(t_j)$ utilizados para los cálculos. Tenemos entonces un sistema, en general no lineal, de n ecuaciones con n incógnitas $v_1, v_2, ..., v_n$, que generalmente se resuelve por un método iterativo de tipo Newton. Expandido, el sistema es el siguiente:

$$\begin{array}{rcl} 2v_1 - v_2 - A + h^2 f(t_1, v_1) & = & 0 \\ -v_1 + 2v_2 - v_3 + h^2 f(t_2, v_2) & = & 0 \\ \vdots & & \vdots \\ -v_{n-2} + 2v_{n-1} - v_n + h^2 f\left(t_{n-1}, v_{n-1}\right) & = & 0 \\ -v_{n-1} + 2v_n - B + h^2 f\left(t_n, v_n\right) & = & 0. \end{array} \tag{6.18}$$

Suponemos que este sistema tiene solución para preservar sencilla esta exposición. Advertimos, sin embargo, que estudiar condiciones suficientes o necesarias para existencia de soluciones de sistemas no lineales, es un tema de investigación actual que está lleno de sutilezas y dificultades.

6.15 Método de Newton

El método de Newton para nuestro sistema se enuncia más fácilmente con notación vectorial. Denotamos por V al vector columna cuyas componentes son $v_1, v_2, \ldots, v_n$ y definimos una función de variable y valor vectorial G de la siguiente manera:

$$\begin{array}{rccc} G: & \mathbb{R}^n & \to & \mathbb{R}^n \\ & V & \longmapsto & GV \end{array}$$

con la componente $j-$ésima de GV igual al lado izquierdo de la ecuación $j-$ésima en (6.18). De esta forma, el sistema que nos ocupa se puede

denotar simplemente como $GV = 0$. Su solución por el método iterativo de Newton exige en cada iteración, la solución de un sistema lineal de ecuaciones. Más precisamente, se procede así:

$$\begin{array}{ll} V^{(0)} & \text{aproximación inicial,} \\ V^{(k+1)} & = V^{(k)} + W, \end{array}$$

donde el superíndice indica orden de iteración y W es la solución del sistema de ecuaciones

$$JW = -GV^{(k)},$$

donde la matriz J es el jacobiano de G en el punto $V^{(k)} = (v_1, v_2, \ldots, v_n)^T$, en el cual omitimos los superíndices para no recargar la exposición. Para cada iteración, J es la matriz tridiagonal dada por

$$\begin{aligned} J_{i-1,i} &= -1 \\ J_{ii} &= 2h^2 f_y(t_i, v_i) \\ J_{i,i+1} &= -1, \end{aligned}$$

para $i = 1, 2, \ldots, n$ y haciendo $v_0 = A$ y $v_{n+1} = B$.

El algoritmo numérico debe incluir un número máximo de iteraciones como criterio de parada y algún otro criterio de parada basado en lo cercanos que son los iterados consecutivos, utilizando alguna de las normas que definimos al principio.

En resumen: el método de diferencias finitas para PVF lleva a un sistema de ecuaciones, posiblemente no lineal, que generalmente debe tratarse por un método iterativo para su solución. Cuando se enuncia un método de Newton, que es uno de los más comunes y poderosos, se ve la necesidad de resolver, en cada iteración un sistema tridiagonal de ecuaciones lineales.

El siguiente ejemplo sirve para afirmar las ideas expresadas arriba.

Ejemplo 34 *Resolver el PVF* $y'' = t(y')^2$ *en* $[0, 2]$, *con* $y(0) = \dfrac{\pi}{2}$ *y* $y(2) = \dfrac{\pi}{4}$. *La solución exacta es* $y(t) = \operatorname{arccot}\left(\dfrac{t}{2}\right)$ *y la usamos para comparación con la solución obtenida por diferencias finitas.*

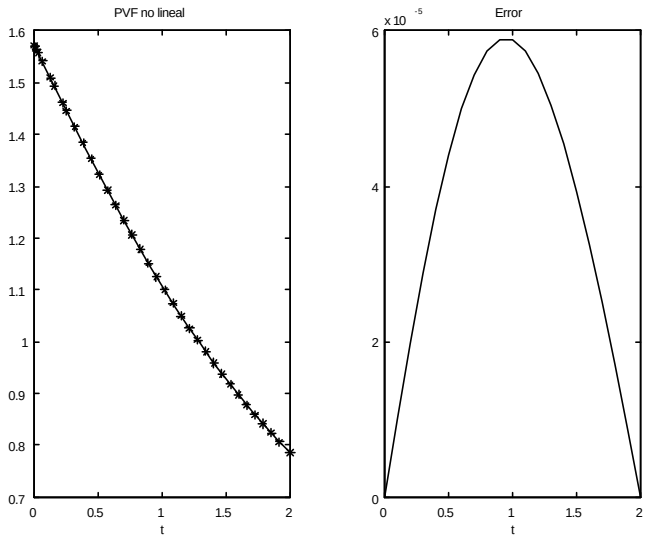
El siguiente M-archivo es tomado de López 2000 [22].

```
function pvfDFnl(fun,gun,hun,p1,p2,v1,v2,n,m,tol,wun)
```

```
% Solución numérica de P.V.F no lineal
% y''=f(x,y,y') a<= x <=b
% y(a)=v1 y(b)=v2
% por el método de diferencia finita.
% pvfDFnl('fun','gun','hun',p1,p2,v1,v2,n,m,tol,'wun')
% fun= f(x,y,y').
% gun= derivada con respecto a y de f(x,y,y').
% hun= derivada con respecto a y' de f(x,y,y').
% wun= solución exacta del P.V.F.
% p1=a p2=b
% n= número de subintervalos
% m= número máximo de iteraciones
% tol= tolerancia
a=zeros(n-1,1);b=zeros(n-2,1);c=zeros(n-2,1);
d=zeros(n-1,1);
h=(p2-p1)/n;k=1;pe=(v2-v1)/(p2-p1);
x=p1:h:p2;w=feval(wun,x');
y=v1+pe*(x'-p1);
while k<=m
for i=1:n-1
t(i)=(y(i+2)-y(i))/(2*h);
a(i)=2+(h^2)*feval(gun,x(i+1),y(i+1),t(i));
d(i)=y(i)-2*y(i+1)+y(i+2)-(h^2)*feval(fun,x(i+1),y(i+1),t(i));
end
for i=1:n-2
b(i)=-1+(h/2)*feval(hun,x(i+1),y(i+1),t(i));
c(i)=-1-(h/2)*feval(hun,x(i+2),y(i+2),t(i+1));
end
J=diag(a)+diag(b,1)+diag(c,-1);
v=J\d;
y=y+[0;v;0];
if norm(v)<tol
disp([x',w,y]);
disp([k,norm(y-w)]);
disp(' éxito');break
end
k=k+1;
```

```
  end
  if k>m
  disp('excede # iteraciones');
  end
  % Las siguientes instrucciones muestran una gráfica de la
solución
  % con las aproximaciones obtenidas
  subplot(1,2,1);
  plot(x,y,'r');
  xlabel('Tiempo');
  title('PVF no lineal')
  hold on
  fplot(wun,[p1 p2],'k*');
  zoom
  hold off
  subplot(1,2,2);
  % Gráfica de los errores
  ee=w-y;
  plot(x,ee,'k');
  title('Error');
  print -deps2 .\fig\pvfDFnl.eps
```

Con este archivo generamos las siguientes figuras:



6.16 Ejercicios

1. Repita el procedimiento de diferencias finitas explicado arriba para el PVF

$$y''(t) = f(t, y, y'), \qquad -\infty < a \leq t \leq b < \infty \tag{6.19}$$

con condiciones de borde

$$y(a) = A, \qquad y(b) = B. \tag{6.20}$$

2. Escriba un M-archivo que se encargue de resolver un PVF de la forma (6.19)-(6.20). Aplique este programa para resolver numéricamente los siguientes casos concretos:

a. $y'' + 2y' + 10t = 0$, $y(0) = 1$, $y(1) = 2$.

b. $y'' = -y$, $y(0) = 3$, $y\left(\frac{\pi}{2}\right) = 7$.

c. $y'' = 2e^t - y$, $y(0) = 2$, $y(1) = e + \cos(1)$.

3. Para los tres ejemplos propuestos en el numeral anterior, encuentre la solución exacta y repita 2. calculando, además de la solución, la distribución del error en los puntos de la malla. Calcule además la norma 2 del error.

4. En 2. utilice mallas uniformes de tamaño de paso 2^{-k}, $k = 2, 3, ..., 10$. Grafique k VS Norma 2 del error.

6.17 Método de colocación

La solución numérica que se obtiene con diferencias finitas es una lista de números $\{v_0, v_1, ..., v_n, v_{n+1}\}$. El primero y el último son valores de frontera no calculados y los otros n se obtienen como solución de un sistema de ecuaciones. Con este resultado, podemos construir una función discreta

$$\bar{F} : \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_n \\ t_{n+1} \end{bmatrix} \rightarrow \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \\ v_{n+1} \end{bmatrix}$$

con asignación $t_j \longmapsto v_j$, $j = 0, 1, ..., n, n+1$. A esta función la podemos extender a todo el intervalo con una regla sencilla como la siguiente:

$$F(t) = \begin{cases} v_0, & t_0 \leq t < t_1 \\ v_k, & t_k \leq t < t_{k+1}, \ k = 1, ..., n \\ v_{n+1}, & t_{n+1} = t. \end{cases}$$

Además, esta función se puede pensar como una combinación lineal de *B-splines de grado cero*. Para ésto, nos conviene pensar que la malla de los t_j hace parte de la malla definida para todos los reales

$$t_j = t_0 + jh, \ j = 0, \pm 1, \pm 2, ...$$

Los B-splines de grado cero se definen así

$$B_k^0(t) = \begin{cases} 1, & t_k \leq t < t_{k+1}, \ k = 0, \pm 1, \pm 2, ... \\ 0, & \text{otro caso.} \end{cases}$$

Estas funciones satisfacen

$$B_j^0(t_k) = \delta_{jk}, \text{ el delta de Kronecker}$$

y además, tienen soporte compacto. Para una descripción de los B-splines, recomendamos Kincaid y Cheney (1994) [21].

La función F se representa

$$F(t) = \sum_{j=0}^{n+1} v_j B_j^0(t). \tag{6.21}$$

Esta es apenas una de las posibles extensiones de $\bar{F}$. Pudimos pretender, por ejemplo, una extensión continua y lineal a trozos o muchas otras. La idea es que para muchas de esas extensiones F, se encuentre una colección de funciones, digamos $\{\phi_j\}$, para la que se cumple una igualdad de la forma (6.21), es decir,

$$F(t) = \sum_{j=0}^{n+1} v_j \phi_j(t). \tag{6.22}$$

El método de colocación, consiste en considerar estas ideas en orden inverso, es decir, dada una colección de funciones $\{\phi_j\}$, obtener una

función F, dada por (6.22), que aproxima la solución del PVF (6.15) - (6.16). En los casos en que el PVF es lineal, el análisis es sencillo. El problema consiste en identificar los coeficientes v_j por medio de la solución de un sistema lineal de ecuaciones. La matriz de este sistema es dada por bandas cuando las funciones ϕ_j tienen soporte compacto.

Los B-splines se usan frecuentemente por tener soporte compacto y también son muy comunes las funciones trigonométricas. En este último caso, se habla de *métodos espectrales* para la solución de ecuaciones diferenciales. Tienen la ventaja de ser rápidos en el computador pues generalmente el procedimiento numérico está basado en transformadas rápidas de Fourier. El libro Trefethen (2000) [33] es una reciente y útil referencia para métodos espectrales. El mismo autor ofrece en Internet Trefethen (1996) [32], que es un moderno libro inconcluso sobre solución numérica de ecuaciones diferenciales que también se refiere al tema de los métodos espectrales.

Presentamos las ideas del método de colocación para el PVF

$$\begin{array}{l} u'' + pu + q = w \\ u\left(0\right) = 0,\ u\left(1\right) = 0. \end{array} \tag{6.23}$$

Aquí, las funciones p, q y w se supone que son continuas en $[0,1]$ y la función incógnita u es un elemento del espacio

$$V = \left\{ v \in C^2\left[0,1\right] / v\left(0\right) = v\left(1\right) = 0 \right\}.$$

Para la definición del método, nos basamos en el operador lineal

$$Lu \equiv u'' + pu + q.$$

Escogemos un conjunto de funciones de V, digamos

$$\left\{ v_j \right\}_{j=1}^{n}$$

y un conjunto de puntos, llamados de colocación o interpolación,

$$\left\{ t_j \right\}_{j=1}^{n} \subset \left[0,1\right].$$

Queremos encontrar una función

$$v = \sum_{j=1}^{n} a_j v_j$$

tal que

$$Lv\left(t_i\right) = w\left(t_i\right),\ i = 1, ..., n.$$

A tal función la encontramos, gracias a la linealidad de L, por medio de la solución del sistema lineal de ecuaciones

$$\sum_{j=1}^{n} a_j (Lv_j)\left(t_i\right) = w\left(t_i\right),\ i = 1, ..., n \tag{6.24}$$

en el que las a_j conforman el vector de incógnitas.

La importancia del método de colocación es tal, que en MATLAB 6, release 12, hay un M-archivo llamado **bvp4c.m,** que sirve para resolver PVFs por un método de colocación. Por su parte, Trefethen (2000) [33], trae un poderoso método llamado de *colocación espectral,* que permite resolver PVFs, lineales y no lineales, de forma rápida y con gran precisión. Para este método, los puntos de colocación son los ceros de los polinomios de Chebyshev.

Concluimos esta subsección con un ejemplo recomendado por Kincaid y Cheney [21] en sección 8.10.

Ejemplo 35 *Resolver por el método de colocación el PVF* $y'' + 4y = \cos(t)$ *en* $\left[0, \frac{\pi}{4}\right]$, *con* $y(0) = y\left(\frac{\pi}{4}\right) = 0$, *utilizando las funciones de prueba* $v_{jk}(t) = t^j \left(\frac{\pi}{4} - t\right)^k$.

Debido a la necesidad de tomar derivadas hasta de orden 2, optamos por utilizar exponentes j y k de 3 en adelante. Para resolver este ejemplo preparamos un sencillo M-archivo que permite trabajar hasta con 25 puntos de colocación solamente. Mejorar esta rutina tal vez no se justifica, por las razones expuestas a continuación. La rutina es:

```
function u=pvf1(a,b,n)
% pvf1.m, metodo de colocacion
% Herramienta didactica, Carlos E. Mejia, 2002
% problema general: Ly=y''+py'+qy = w
% basado en Kincaid y Cheney seccion 8.10
% a, b: limites del intervalo
% n: numero de nodos de colocacion y de funciones de prueba
h = (b-a)/(n+1);t = a+h:h:b-h;
```

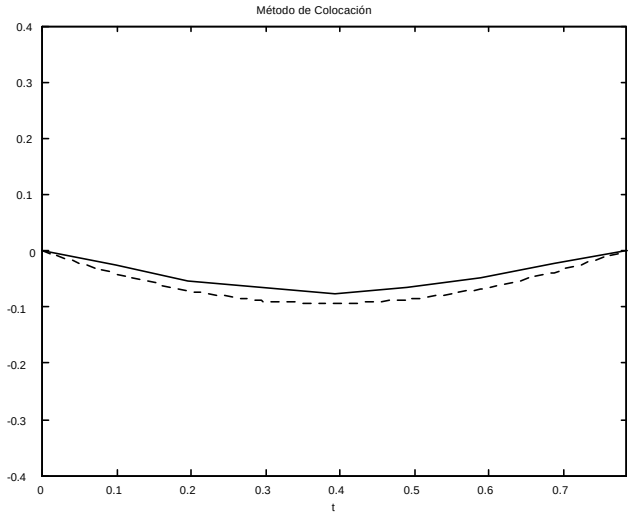
```
% coeficientes
q=4*ones(size(t));w=cos(t)';p=zeros(size(t));
for m=1:n
 [j k]=poten(m);
 lv = j*(j-1)*vf(j-2,k,t)-2*j*k*vf(j-1,k-1,t)+k*(k-1)*vf(j,k-2,t);
 lv = lv + p.*(j*vf(j-1,k,t)-k*vf(j,k-1,t)) + q.*vf(j,k,t);
 ma(:,m)=lv';
end
%c=ma\w;
c=bicgstab(ma,w,1e-4,50);
up=cell(size(c));u=zeros(size(c));
for m=1:n
 [j k]=poten(m);
 up{m}=vf(j,k,t)';
 u=u+c(m).*up{m};
end
ex=exaf(t);er=u-ex';
er2=sqrt((1/(n+2))*sum(er.^2))
ee2=norm(er)
T=[0 t pi/4];ex=[0; ex'; 0];u=[0; u; 0];
plot(T,u,'r',T,ex,'k--')
 axis([0 pi/4 -.4 .4])
 title('Método de Colocación')
 xlabel t,
 print -deps2 .\fig\col2.eps
```

Los polinomios básicos se generan por medio de la función **vf.m** y la solución exacta está en **exaf.m.** La rutina **poten.m** se ocupa de asignar exponentes j y k a cada índice m. La incluimos por completez:

```
function [j,k]=poten(m)
if m>=1 & m<6
 j=3+m-1;k=3;
elseif m>=6 & m<11
 j=3+m-6;k=4;
elseif m>=11 & m<16
 j=3+m-11;k=5;
elseif m>=16 & m<21
 j=3+m-16;k=6;
```

```
elseif m>=21 & m<26
 j=3+m-21;k=7;
end
```

Al invocar **u=pvf1(0,pi/4,7),** obtuvimos la siguiente gráfica:



Para poder resolver el sistema lineal, tuvimos que acudir a un sofisticado método iterativo para la solución del sistema de ecuaciones (6.24). Se trata de **BI-CGSTAB**, publicado por primera vez por H. A. van der Vorst en 1992. Se trata de un método de gradiente conjugado con biortogonalización, que además posee un caracter estabilizador. MATLAB incluye este método en la rutina **bicgstab.m.** Para saber más sobre este método recomendamos Trefethen y Bau (1997) [34].

Los resultados son de inferior calidad a los obtenidos antes por diferencias finitas para un problema no lineal. Hay una razón numérica de peso para esperar pobres resultados. Se trata de los polinomios escogidos como funciones básicas, que no se escogieron en ninguna familia de polinomios ortogonales. Los polinomios escogidos generan sistemas lineales muy mal condicionados.

Un caso análogo y muy famoso, es el de tratar de usar los polinomios x^j, como funciones básicas para construir el polinomio que sea la mejor aproximación de una función dada en el sentido de los mínimos cuadrados. Con la función constante 1 como función de ponderación, el sistema lineal al que se llega es uno de los más mal condicionados de

los que se tenga noticia, pues la matriz del sistema es la temida matriz de Hilbert H_n, cuyo elemento (i,j) es $\dfrac{1}{i+j+1}$. A medida que crece n, el tamaño de la matriz, también aumenta el número de condición de H_n. Más detalles de este interesante ejemplo pueden verse en Atkinson, 1978 [4], sección 4.3.

6.18 Ejercicios suplementarios

1. El problema de calcular la integral de f se puede pensar como la solución del PVI

$$y' = f(t), \; y(a) = 0, \tag{6.25}$$

cuya solución es

$$y(t) = \int_a^t f(x)\, dx. \tag{6.26}$$

Muestre que el método de Euler aplicado a (6.25) es equivalente a la regla del rectángulo para (6.26) y si se aplica a (6.25) el método clásico de Runge-Kutta de cuarto orden, es lo mismo que aplicar la regla de Simpson compuesta a (6.26).

2. Aplique el método de Euler y el método clásico de Runge-Kutta al PVI

$$y' = t^2 + (y(t))^2, \; y(0) = 1, \; t \geq 0.$$

Compare los dos resultados. Tenga en cuenta que la solución de este PVI no existe para todo t.

3. Intente resolver, por el método de Euler, los ejemplos 2. y 3. de la sección de ejemplos 25. Puede haber dificultades, debe tratar varios valores de h.

4. Considere este otro problema de especies en competencia.

$$\begin{aligned} y_1'(t) &= 0.25 y_1(t) - 0.01 y_1(t)\, y_2(t) \\ y_2'(t) &= -y_2(t) + 0.01 y_1(t)\, y_2(t) \end{aligned}$$

con condición inicial $y_1(0) = 80$ y $y_2(0) = 30$. Utilice un método de alto orden, por ejemplo **ode45** de MATLAB, para resolver

este problema y graficar su diagrama de fase. Enseguida, use el método de Euler para el mismo problema con diferentes valores de h, digamos $h = 1,\ 0.5,\ 0.25$ y 0.125. Probablemente los nuevos diagramas de fase no concuerdan con el que ya obtuvo por el método de alto orden. Explique motivos para esta discrepancia.

5. (Continuación) En el problema anterior, cambie la condición inicial por cada una de las siguientes: $(81, 30)$, $(80, 31)$, $(79, 30)$ y $(80, 29)$. Este es un test de estabilidad, determina qué tan grandes son los cambios en la solución del PVI cuando ocurren pequeños cambios en la condición inicial.

6. Considere el PVI

$$y' = -100y + 100,\ y(0) = 2,$$

cuya solución exacta es $y(t) = e^{-100t} + 1$. Este es un problema de los llamados **stiff.** Es claro que la solución decrece rápidamente de 2 a casi 1 y después se queda prácticamente horizontal. Compruebe que esta situación hace difícil resolver este problema por el método de Euler. Trate varios valores de h para convencerse. Una alternativa bastante buena es utilizar el método de Euler hacia atrás, definido por la ecuación en diferencias

$$Y_{n+1} = Y_n + hf(t_{n+1}, Y_{n+1}).$$

Este es un método implícito que permite resolver este problema stiff sin mayores contratiempos. Compruébelo!
Sugerencia: En el método de Euler, los iterados son

$$Y_n = (1 - 100h)^n + 1$$

y en el método de Euler hacia atrás, son

$$Y_n = \frac{1}{(1 + 100h)^n} + 1.$$

7. Considere el PVF

$$v'' = v^4,\ 0 \leq t \leq 1,\ v(0) = 1,\ v(1) = 1/2.$$

a. Aproxime la solución del PVF con un polinomio de grado 3 que satisface los valores $v(0)$, $v(1)$, $v''(0)$ y $v''(1)$.
b. Aproxime la solución del PVF por medio del método de diferencias finitas.

6.19 Examen de entrenamiento

Resuelva estos ejercicios sin consultar libros ni notas de clase para darse una idea del nivel de preparación que ha obtenido hasta ahora. Una calculadora sencilla es lo mínimo que debe tener a disposición y es suficiente para poder responder todos los ejercicios del examen. Claro que es preferible si dispone de una calculadora graficadora o un computador.

1. Determine las constantes de Lipschitz para las siguientes funciones en $[-1, 1]$:
   a. $f(t, y) = t^2 y$.
   b. $f(t, y) = t \exp(-y^2)$.

2. Muestre que el PVI $y' = \sqrt{|y|}$, $y(0) = 0$, tiene infinitas soluciones. Hágalo verificando que para cualquier $c \geq 0$, la función $y(t)$ es una solución:

$$y(t) = \begin{cases} 0, & t \leq c \\ \dfrac{(t-c)^2}{4}, & t > c. \end{cases}$$

   ¿Qué solución de este PVI será la que aproxima el método de Euler?

3. Resuelva por el método de Euler con $h = 0.25$, el siguiente PVI: $y' = 2ty$, $y(0) = 1$, $0 \leq t \leq 1$.

4. Convierta los PVIs siguientes, dados en términos de ecuaciones escalares de grado 2, a PVIs con variable vectorial.
   a. $y'' = -sen(y)$, $y(0) = \pi/2$, $y'(0) = 0$.
   b. $y'' - y = t$, $y(0) = 1$, $y'(0) = 1$.

7

Ecuaciones diferenciales parciales

Ecuaciones de tipo parabólico
Diferencias finitas con condiciones de borde de tipo Dirichlet
Ecuaciones no lineales y otras condiciones de borde
Consistencia, estabilidad y convergencia
Dos dimensiones
Métodos ADI

7.1 Diferencias finitas para problemas parabólicos

En esta sección nos referimos brevemente a la solución de ecuaciones diferenciales parciales por diferencias finitas. Para la exposición, nos basamos en Smith (1978) [28] y Kincaid y Cheney (1994) [21] pero recomendamos consultar también referencias más completas como Strikwerda (1989) [31] y el libro clásico Richtmyer y Morton (1967) [27]. El objetivo es motivar el estudio a partir de ejemplos sencillos de discretizaciones de diferencias finitas de problemas de tipo parabólico.

A partir de MATLAB 6, release 12, hay una rutina especial para resolver una amplia gama de ecuaciones unidimensionales de tipo parabólico o elíptico. Se trata de **pdepe.m.** Además, desde hace varios años, existe una *caja de herramientas* especializada en ecuaciones diferenciales parciales. Aquí no nos referimos a ninguna de estas alternativas que ofrece MATLAB. Quedan como temas llamativos para cuando estas notas crezcan en esta dirección.

7.2 Ecuaciones de tipo parabólico

La ecuación diferencial parcial de tipo parabólico más sencilla es la ecuación unidimensional

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},\ a < x < b. \tag{7.1}$$

La variable espacial x se toma en un intervalo $[a, b]$ y la variable temporal t se toma en el intervalo semi-infinito $[0, \infty)$. Las condiciones de borde se definen para $x = a$ y $x = b$ y para todo t. La condición inicial se define para $x \in (a, b)$ y $t = 0$. La ecuación (7.1) tiene validez en $(a, b) \times (0, \infty)$. Esta ecuación da cuenta de procesos difusivos lineales, por ejemplo, la conducción de calor. Debido a ésto, muy frecuentemente nos referimos a la variable dependiente como temperatura, aunque este modelo se puede utilizar en muchos otros procesos.

Consideremos discretizaciones uniformes en espacio y tiempo dadas por

$$x_i = a + ih,\ i = 0, 1, ... \text{ y } t_j = jk,\ j = 0, 1, ...$$

donde h y k son los tamaños de paso en las direcciones de espacio y tiempo respectivamente. Ambos son reales positivos. Es conveniente definir también

$$r = \frac{k}{h^2}.$$

7.3 Diferencias finitas

En todas las aproximaciones de diferencias finitas utilizamos la variable discreta V_i^j para representar la aproximación de u en el punto de la malla (x_i, t_j) que se calcula por medio del método numérico. Sea $h = \dfrac{b-a}{n+1}$ y supongamos que tenemos condiciones de borde de tipo Dirichlet para $x = a = x_0$ y $x = b = x_{n+1}$, es decir, los elementos V_0^j y V_{n+1}^j son conocidos para todo j. Además, definimos la variable discreta vectorial $V^j = \left[V_1^j, V_2^j, ..., V_n^j\right]$ para representar el vector de aproximaciones en el nivel temporal j. Nótese que una condición inicial significa que el vector V^0 es conocido. Posteriormente consideramos otras condiciones de borde.

7.3.1 Métodos más comunes

Distinguimos varias expresiones de diferencias finitas que aproximan (7.1).Veamos algunas:

1. Método explícito, es decir, la temperatura V_i^{j+1} se escribe en términos de temperaturas en niveles anteriores en la escala temporal.

$$\frac{V_i^{j+1} - V_i^j}{k} = \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{h^2}$$

que conduce a

$$V_i^{j+1} = V_i^j + r\left(V_{i-1}^j - 2V_i^j + V_{i+1}^j\right). \tag{7.2}$$

Sea $A = trid\,(r, 1-2r, r)$ la matriz tridiagonal de orden n con elementos diagonales $1-2r$ y elementos super y sub diagonales iguales a r. La igualdad (7.2) se puede escribir

$$V^{j+1} = AV^j \tag{7.3}$$

lo que indica que

$$V^j = A^j V^0. \tag{7.4}$$

Los iterados consecutivos se consiguen por aplicación de potencias de A a la condición inicial.

2. Método implícito, es decir, la temperatura V_i^{j+1} se escribe en términos de temperaturas en el mismo nivel y en niveles anteriores de la escala temporal,

$$\frac{V_i^{j+1} - V_i^j}{k} = \frac{V_{i+1}^{j+1} - 2V_i^{j+1} + V_{i-1}^{j+1}}{h^2},$$

que se puede escribir

$$(1+2r)\,V_i^{j+1} - rV_{i-1}^{j+1} - rV_{i+1}^{j+1} = V_i^j. \tag{7.5}$$

Sea $A = trid\,(-r, 1+2r, -r)$ la matriz tridiagonal de orden n con elementos diagonales $1+2r$ y elementos super y sub diagonales iguales a $-r$. La igualdad (7.5) se puede escribir

$$AV^{j+1} = V^j. \tag{7.6}$$

Es decir, en cada paso temporal, se debe resolver un sistema de ecuaciones lineales con matriz simétrica, tridiagonal y diagonalmente dominante (por tanto no singular).

3. Métodos mixtos dependientes de un parámetro $\alpha \in [0,1]$.

$$\frac{V_i^{j+1}-V_i^j}{k}=\alpha\frac{V_{i+1}^{j+1}-2V_i^{j+1}+V_{i-1}^{j+1}}{h^2}+(1-\alpha)\frac{V_{i+1}^{j}-2V_i^{j}+V_{i-1}^{j}}{h^2}.$$

Los dos métodos anteriores son casos particulares de éste, basta hacer $\alpha = 0$ y 1 respectivamente. El más importante de los métodos mixtos es el que corresponde a $\alpha = \dfrac{1}{2}$ que se llama de Crank-Nicolson. Está dado por la igualdad

$$\frac{V_i^{j+1}-V_i^j}{k}=\frac{1}{2h^2}\left(V_{i+1}^{j+1}-2V_i^{j+1}+V_{i-1}^{j+1}+V_{i+1}^{j}-2V_i^{j}+V_{i-1}^{j}\right)$$

que también podemos escribir

$$-rV_{i-1}^{j+1}+2(1+r)V_i^{j+1}-rV_{i+1}^{j+1}=rV_{i-1}^{j}+2(1-r)V_i^{j}+rV_{i+1}^{j}. \quad (7.7)$$

Sean $A = trid(-r, 2(1+r)-r)$ y $B = trid(r, 2(1-r), r)$ matrices de orden n. La ecuación (7.7) se escribe como el sistema

$$AV^{j+1} = BV^j,$$

con V^{j+1} como incógnita. Este es el sistema que debe resolverse para cada iteración temporal.

Ejemplo 36 *Resolver* $\dfrac{\partial u}{\partial t}=\dfrac{\partial^2 u}{\partial x^2}$ *en* $[0,2]$, *con condiciones homogéneas en los bordes y con condición inicial dada por* $u_0(x) = sen\left(\dfrac{\pi x}{2}\right)$.

La solución exacta es $u(x,t) = \exp\left(-\dfrac{\pi^2 t}{4}\right) sen\left(\dfrac{\pi x}{2}\right)$. Preparamos un M-archivo para resolver el problema con diferentes valores para r y n. La calidad o falta de calidad de los resultados, sugieren que r debe cumplir algún requisito para obtener resultados útiles. Esto lo confirmamos más adelante en la subsección 7.8.

Métodos explícito e implícito			
n	r	Error (explícito)	Error (Implícito)
10	0.5	4.1813e-002	3.0735e-002
10	0.8	2.1062e-002	4.4402e-003
20	0.5	1.1529e-002	8.6889e-003
20	0.8	5.6956e-003	1.2394e-003
30	0.5	5.2961e-003	4.0122e-003
30	0.8	8.7532e-003	6.6802e-003
40	0.5	3.0288e-003	2.2990e-003
40	0.8	3.6016e+001	1.4966e-003
50	0.5	1.9578e-003	1.4874e-003
50	0.8	2.6821e+011	1.7249e-003
60	0.5	1.3687e-003	1.0403e-003
60	0.8	4.7197e+022	1.2066e-003

Estos resultados indican que el método implícito tiende a ofrecer mejores resultados y que el método explícito ofrece algunos totalmente inaceptables. Estos últimos se obtienen siempre para $r = 0.8$ pero no para $n < 40$. Posteriormente le daremos total sentido a esta tabla. Advertimos que los malos resultados están asociados con falta de estabilidad y que en términos coloquiales, estabilidad significa que los resultados *no estallan.* La razón por la que los resultados malos empeoran a medida que crece n, es porque hay que hacer más cálculos y por tanto hay más ocasión de propagar y agrandar los errores.

7.3.2 Ejercicios

1. Consideremos el problema

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

para $x \in [0,1]$ con condición inicial, condiciones de borde y solución exacta dadas por

$$u(t,x) = [sen(\pi x)] \exp\left(-\pi^2 t\right).$$

Utilice los tres métodos propuestos en la subsección 7.3.1 y obtenga la solución para $t = 0.1$. Trabaje con varios valores de h y k de manera que r tome los valores $0.1n$, con $n = 1, 2, ..., 8$.

2. Se trata de resolver el problema

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

para $x \in [-1, 1]$ con solución exacta y condiciones de borde dadas por

$$u(t, x) = \frac{1}{2} + 2\sum_{n=0}^{\infty} (-1)^n \frac{\cos \pi (2n+1) x}{\pi (2n+1)} \exp\left(-\pi^2 (2n+1)^2 t\right)$$

y con condición inicial

$$u_0(x) = \begin{cases} 1 & \text{si } |x| < \frac{1}{2} \\ \frac{1}{2} & \text{si } |x| = \frac{1}{2} \\ 0 & \text{si } |x| > \frac{1}{2}. \end{cases}$$

Encuentre la solución para $t = \frac{1}{2}$ utilizando los tres métodos propuestos en la subsección 7.3.1. Utilice varios valores de h y k de manera que r tome los valores $0.1n$, con $n = 1, 2, ..., 8$.

7.4 Ecuaciones no lineales

Tal como vimos al principio de estas notas, el método de Newton es un recurso importante cuando se trata de afrontar un problema no lineal. Veamos un ejemplo sencillo que aparece como Ejemplo 2.7 en [28]. La temperatura u satisface la ecuación no lineal

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u^2}{\partial x^2}, \ 0 < x < 1,$$

con condición inicial $u(x) = 4x(1-x)$, $0 < x < 1$, $t = 0$ y las condiciones de borde $u = 0$ para $x = 0$ y $x = 1$, $t \geq 0$. Para estudiar el método de Newton aplicado a este problema, consideremos el siguiente M-archivo que se encarga de implementarlo:

```
function vnew=smith27ci(tfin,n);
% Herramienta didactica, Carlos E. Mejia, 2002
% ecuacion parabolica no lineal en 1-D
% ejemplo 2-7 de Smith
% metodo de Newton para hallar ceros de F(x)=0
% tfin: tiempo final
% n: numero de subintervalos en [0,1], n>2
% h: tamaño de paso en espacio
% k: tamaño de paso en tiempo
h=1/n;
k=.5*h^2; % k puede definirse en otras formas si se desea
p=h^2/k;
% x: dominio espacial, xa: dominio espacial ampliado
x=(h:h:1-h)';xa=[0;x;1];
% vold: condicion inicial
vold=4.*x.*(1-x);
vnew=vold;
t=0;
while t<tfin
 va=[0;vnew;0];
 plot(xa,va,'r');
 title('Difusion no lineal');
 xlabel('Espacio')
 hold on;
 count=floor(t/k)+1;
 % condiciones de borde son homogeneas
 % diferencias finitas tomadas en ih y (j+1/2)k
 % g: parte de F(v), depende de v en tiempo anterior
 % fv: parte de F(v) que depende de v en tiempo actual
 % jf: matriz jacobiana de F
 in2=1;it=1;
 while in2>1.e-8 & it<10
 g=zeros(n-1,1);fv=g;jf=zeros(n-1);
g(1)=vold(2)^2-2*(vold(1)^2-p*vold(1));% 1
fv(1)=vnew(2)^2-2*(vnew(1)^2+p*vnew(1))+g(1);
jf(1,1)=-4*vnew(1)-2*p;
jf(1,2)=2*vnew(2);
```

```
g(n-1)=-2*(vold(n-1)^2-p*vold(n-1))+vold(n-2)^2;% 2
fv(n-1)=-2*(vnew(n-1)^2+p*vnew(n-1))+vnew(n-2)^2+g(n-1);
jf(n-1,n-2)=2*vnew(n-2);
jf(n-1,n-1)=-4*vnew(n-1)-2*p;
for i=2:n-2
 g(i)=vold(i+1)^2-2*(vold(i)^2-p*vold(i))+vold(i-1)^2;% 3
 fv(i)=vnew(i+1)^2-2*(vnew(i)^2+p*vnew(i))+vnew(i-1)^2+g(i);
 jf(i,i-1)=2*vnew(i-1);
 jf(i,i)=-4*vnew(i)-2*p;
 jf(i,i+1)=2*vnew(i+1);
end
% solucion sistema lineal asociado con metodo de Newton
in=jf\-fv;
in2=norm(in);
% actualizacion de vold y vnew
vnew=vnew+in;it=it+1;
end
t=t+k;vold=vnew;
end
```

Explicamos ahora el método de diferencias utilizado. En el punto $\left(ih, \left(j+\frac{1}{2}\right)k\right)$ la aproximación de diferencias es

$$\frac{1}{k}\left(V_{i,j+1}-V_{i,j}\right)=\frac{1}{2h^2}\left(V_{i-1,j+1}^2-2V_{i,j+1}^2+V_{i+1,j+1}^2+V_{i-1,j}^2-2V_{i,j}^2+V_{i+1,j}^2\right)$$

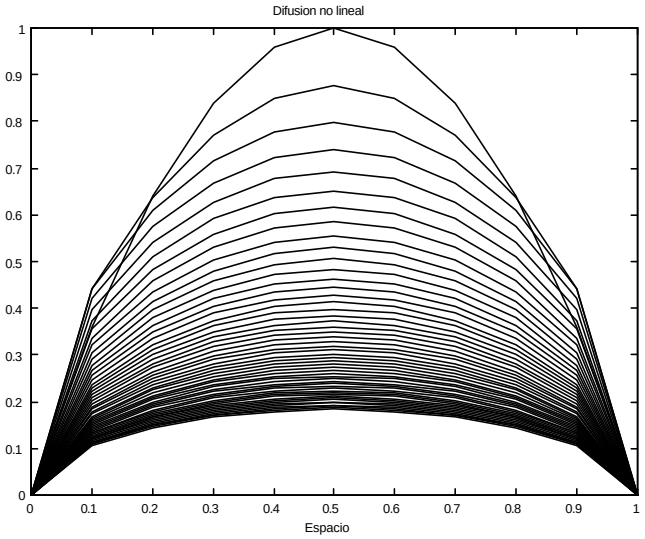
en donde las incógnitas se distinguen por el subíndice $j+1$. Esta es la igualdad que define la expresión no lineal que debemos resolver por el método de Newton. Hacemos $p=\dfrac{h^2}{k}$ y definimos las $i-$ ésimas filas de los vectores $g\left(V\right)$ y $f\left(V\right)$, de la siguiente manera

$$\begin{aligned}\left[g\left(V\right)\right]_i &= V_{i-1,j}^2-2\left(V_{i,j}^2-pV_{i,j}\right)+V_{i+1,j}^2\\ \left[f\left(V\right)\right]_i &= V_{i-1,j+1}^2-2\left(V_{i,j+1}^2+pV_{i,j+1}\right)+V_{i+1,j+1}^2\end{aligned}$$

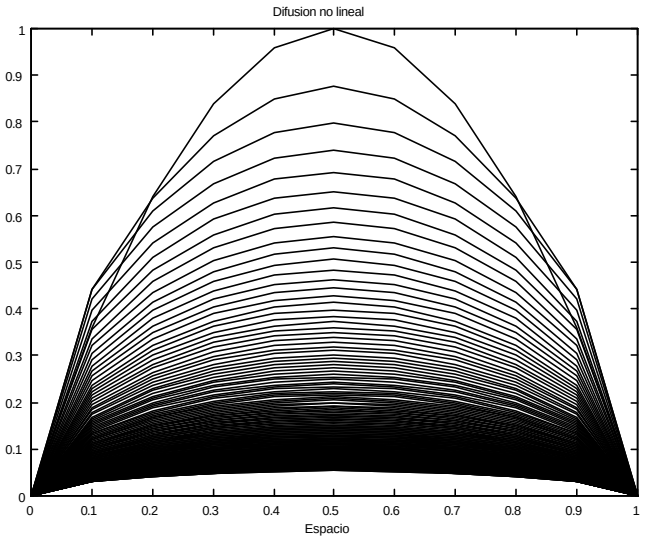
donde en g se reune lo conocido y f reune las incógnitas. Para $i=1$ e $i=n-1$ hay expresiones ligeramente diferentes debido a las condiciones homogéneas en el borde (Ver líneas marcadas % 1 y % 2 en el M-archivo de arriba. Para la expresión general, ver línea % 3.) La matriz jacobiana

es `jf` y puede verse que también requiere de armado especial en las filas extremas por las condiciones de borde.

En las gráficas que adjuntamos, es clara la difusión, la tendencia de las temperaturas calculadas es hacia la temperatura nula. Al invocar `smith27ci(0.5,10)`, se obtiene la gráfica



y al invocar `smith27ci(2,10),` se consigue la siguiente:



Naturalmente sabemos que esta presentación es apenas una invitación a consultar más acerca de cómo resolver problemas parabólicos no lineales. En Richtmyer y Morton (1967) [27], pag. 201, hay un método para resolver

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u^5}{\partial x^2}$$

que es fácilmente generalizable a la ecuación

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u^m}{\partial x^2}, \text{ para cualquier } m \text{ entero positivo.}$$

Los cálculos que presentan los autores para ilustrar su método, fueron realizados por ellos mismos en un computador Univac en el año 1954. Eran los primeros intentos por resolver problemas no lineales por diferencias finitas.

7.5 Ejercicio

Considere el problema

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u^2}{\partial x^2}, \ 0 < x < 1 \tag{7.8}$$

y suponga que la solución u satisface la ecuación no lineal

$$2u - 3 + \log\left(u - \frac{1}{2}\right) = 2\,(2t - x) \tag{7.9}$$

para todo t y x.

a. Utilizando el método de Newton para ecuaciones no lineales, obtenga a partir de (7.9) la condición inicial, las condiciones de borde y la solución para $t = 0.5$.

b. Utilice el método de Newton y el método de diferencias finitas descrito antes para aproximar la solución de (7.8) en $t = 0.5$. Los resultados para $t = 0.5$ en las partes a. y b. deben coincidir.

7.6 Otras condiciones de borde

En muchas circunstancias las condiciones de borde se expresan por medio de derivadas. Por ejemplo, la rata a la cual se transfiere calor por radiación de una superficie externa a temperatura u hacia el medio que la rodea que está a temperatura v, es proporcional a $u - v$. Más aun, Fourier descubrió que tal rata es igual a

$$-K\frac{\partial u}{\partial \eta},$$

donde K es la conductividad térmica del material. Todo ésto da lugar a una condición de borde de la forma

$$-K\frac{\partial u}{\partial \eta} = H\left(u - v\right),$$

donde H es una constante de proporcionalidad que se conoce como el coeficiente de transmisión de calor.

Para indicar el tratamiento de estas condiciones de borde por el método de diferencias finitas, presentamos el ejemplo 2.3 de Smith (1978) [28]. Nótese que las derivadas en la dirección normal, se convierten en este caso en derivadas con respecto a x.

Ejemplo 37 *Resolver por el método explícito (7.2) la ecuación*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},\ 0 < x < 1$$

que satisface la condición inicial

$$u = 1 \text{ para } 0 \leq x \leq 1 \text{ cuando } t = 0$$

y las condiciones de borde

$$\frac{\partial u}{\partial x} = u \text{ en } x = 0 \text{ para todo } t$$

$$\frac{\partial u}{\partial x} = -u \text{ en } x = 1 \text{ para todo } t.$$

Recordamos que el método de diferencias que deseamos utilizar está dado por la igualdad

$$V_i^{j+1} = V_i^j + r\left(V_{i-1}^j - 2V_i^j + V_{i+1}^j\right), \tag{7.10}$$

donde $r = \dfrac{k}{h^2}$. En $x = 0$ y $x = 1$ optamos por utilizar nodos virtuales no pertenecientes a la malla para realizar los cálculos, en la siguiente forma: En $x = 0$,

$$V_0^{j+1} = V_0^j + r\left(V_{-1}^j - 2V_0^j + V_1^j\right)$$

y la condición de borde, discretizada con diferencias centrales, nos lleva a

$$\frac{1}{2k}\left(V_1^j - V_{-1}^j\right) = V_0^j.$$

Al eliminar lo correspondiente al nodo virtual $i = -1$ de estas dos ecuaciones, llegamos a

$$V_0^{j+1} = V_0^j + 2r\left(V_1^j - (1+k)\,V_0^j\right). \tag{7.11}$$

Similarmente para $x = 1$ tenemos

$$V_{n+1}^{j+1} = V_{n+1}^j + r\left(V_n^j - 2V_{n+1}^j + V_{n+2}^j\right)$$

y la condición de borde, discretizada por diferencias centrales, nos da

$$\frac{1}{2k}\left(V_{n+2}^j - V_n^j\right) = -V_{n+1}^j.$$

Por tanto para $i = n + 1$, la ecuación en diferencias es

$$V_{n+1}^{j+1} = V_{n+1}^j + 2r\left(V_n^j - (1+k)\,V_{n+1}^j\right). \tag{7.12}$$

De (7.10) se obtienen n ecuaciones, correspondientes a $i = 1, ..., n$. De (7.11) se obtiene una más, correspondiente a $i = 0$ y finalmente de (7.12) se consigue la ecuación que corresponde a $i = n+1$. El sistema de estas $n + 2$ ecuaciones nos da la solución al ejemplo propuesto, siempre y cuando se respeten otras condiciones con respecto al valor de r, que son el tema de la sección siguiente.

7.7 Consistencia, estabilidad y convergencia

Para esta subsección adoptamos la siguiente convención:

$L(u) = f$ representa la ecuación diferencial parcial en las variables independientes x y t con solución exacta u. El operador L es lineal e incluye implícitamente las condiciones de borde.

$F_h(V) = f$ representa la aproximación de diferencias finitas con solución exacta V. Implícitamente queda establecida la dependencia $k = rh^2$ y por tanto $k \to 0$ si y solo si $h \to 0$.

v es una función continua de x y t con un número suficiente de derivadas continuas de manera que $L(v)$ se pueda evaluar en el punto (ih, jk).

La notación v_i^j significa la evaluación de v en (ih, jk). Similarmente para la solución exacta u.

$T_{i,j}(v) = F_h\left(v_i^j\right) - L\left(v_i^j\right)$ es el *error de truncación* en el punto (ih, jk).

$e_{i,j} = u_i^j - V_i^j$ es el *error de discretización* en el punto (ih, jk).

Si $T_{i,j}(v) \to 0$ cuando $h \to 0$, se dice que la ecuación en diferencias es *consistente* o *compatible* con la ecuación diferencial parcial.

En todas las situaciones que nos interesan, $T_{i,j}(v) = O(h^s)$ con $s > 0$. Si en el lugar de v escribimos la solución u, la cual restringimos a los puntos de la subdivisión, obtenemos

$$F_h(u) = f + O(h^s).$$

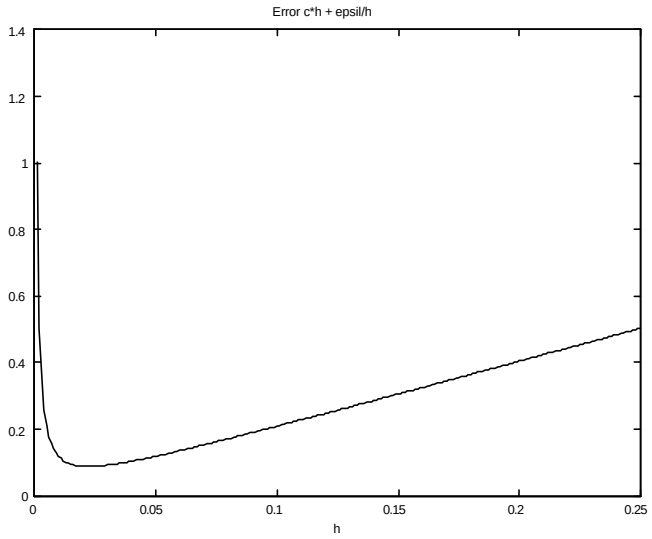
La solución V de la ecuación en diferencias no es la que se obtiene en el computador, debido a los inevitables errores en los datos y de redondeo. Digamos que la que se obtiene es la solución numérica N. Al error $V - N$ se le denomina *error de redondeo.*

El error total en el punto (ih, jk) está dado por

$$u_i^j - N_i^j = \left(u_i^j - V_i^j\right) + \left(V_i^j - N_i^j\right). \tag{7.13}$$

Si $u_i^j - N_i^j \to 0$ cuando $h \to 0$, se dice que V *converge* a u. El error total se compone de dos partes, dadas por los dos sumandos en el lado derecho de (7.13). El primer sumando está asociado con la consistencia, como vimos antes. El segundo es más complicado pues tiene que ver con redondeo y no podemos pedir simplemente que tienda a cero cuando

$h \to 0$, pues eso no ha de funcionar, como lo vimos con claridad en el análisis de estabilidad para PVI. Aquí también, estamos ante gráficas como la presentada allá que repetimos por completez.



La *estabilidad* la definimos así: Un método de diferencias finitas es estable si el segundo sumando del lado derecho de (7.13) no crece sin límite.

El *teorema de equivalencia de Lax* afirma que para aproximaciones *consistentes* de diferencias finitas, *estabilidad* es necesaria y suficiente para *convergencia*. Como se ha visto ya, la consistencia está regida por la aproximación discreta elegida y por tanto se tiene completo control a este respecto. Pero con la estabilidad no hay reglas generales, ni límites a cero. La estabilidad depende, entre otros, de los datos, de las ecuaciones y del computador. Es la propiedad más difícil de alcanzar pero cuando se alcanza, se consigue todo, pues es *equivalente* a convergencia en presencia de consistencia. Recomendamos Richtmyer y Morton (1967) [27] para una presentación formal de este importante teorema.

7.8 Análisis Matricial de Estabilidad

Volvamos al método explícito (7.10) aplicado a la ecuación

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},\ a < x < b$$

con condiciones homogéneas en el borde y una condición inicial $u_0(x)$. Supongamos que hay errores en la condición inicial y no los hay en el procedimiento de cálculo. Esta es una simplificación que hace más claro el análisis de estabilidad. Tenemos una nueva condición inicial discreta N^0 que origina nuevos iterados N^j en el proceso.

De acuerdo con (7.4),

$$V^j = A^j V^0 \text{ y } N^j = A^j N^0.$$

Por tanto el error en el paso j está dado, gracias a la linealidad, por

$$V^j - N^j = A^j \left(V^0 - N^0\right)$$

que lleva a

$$\begin{aligned} \|V^j - N^j\| &= \|A^j (V^0 - N^0)\| \\ &\leq \|A^j\| \, \|V^0 - N^0\| \end{aligned}$$

y para que este error no crezca sin límite, que es lo que llamamos *estabilidad,* requerimos que $\|A^j\|$ sea acotada cuando $j \to \infty$.

Recurrimos a un conocido teorema del álgebra lineal numérica que aquí citamos de Ortega (1990) [26], pag. 25. Antes vemos la siguiente definición.

Definición 38 *Una matriz A es de clase M si para todos los valores propios λ tales que $|\lambda| = \rho(A)$, se cumple que todos los bloques de Jordan asociados con λ son 1×1.*

Teorema 39 *$\|A^j\|$ es acotada cuando $j \to \infty$ si y solo si $\rho(A) < 1$ o $\rho(A) = 1$ y A es de clase M.*

En el caso del método explícito (7.10), estamos ante la matriz

$$A = trid\,(r, 1 - 2r, r)$$

de la que se sabe que tiene n valores propios distintos, por tanto es diagonalizable y en particular, es de clase M.

Los n valores propios distintos de A son

$$\lambda_m = 1 - 4rsen^2\left(\frac{m\pi}{n+1}\right),\ m = 1, 2, ..., n.$$

y queremos

$$|\lambda_m| \leq 1 \text{ para todo } m.$$

Nótese que

$$|\lambda_m| \leq 1 \text{ si y solo si } r \leq \frac{1}{2sen^2\left(\frac{m\pi}{n+1}\right)}$$

y esta última desigualdad es cierta si y solo si $r \leq \frac{1}{2}$.

Los métodos de diferencias que son estables para todos los valores de r se llaman *incondicionalmente estables*. Los demás, como el método explícito que acabamos de considerar, se denominan *condicionalmente estables.*

7.9 Ejercicios

Considere los métodos de diferencias definidos en la subsección 7.3.1.

1. Compruebe que son consistentes con la ecuación diferencial.

2. Compruebe que los valores propios de la matriz A asociada con el método (7.2) son los listados arriba.

3. Realice un análisis matricial de estabilidad para el método implícito (7.5) y para el método de Crank-Nicolson. Demuestre que ambos son incondicionalmente estables.

4. Para el método explícito (7.2) aplicado a la ecuación

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},\ a < x < b$$

con condiciones homogéneas en el borde y una condición inicial $u_0(x)$, demuestre directamente que la condición $0 < r \leq \frac{1}{2}$ es suficiente para

estabilidad, probando que lleva a soluciones acotadas para la ecuación en diferencias.

Sugerencia: Haga $h = \dfrac{b-a}{n+1}$ y $r = \dfrac{k}{h^2}$, y note que la condición $0 < r \leq \dfrac{1}{2}$ lleva a

$$\begin{aligned} \left|V_i^{j+1}\right| &\leq |1-2r| \left|V_i^j\right| + r\left[\left|V_{i-1}^j\right| + \left|V_{i+1}^j\right|\right] \\ &= (1-2r)\left|V_i^j\right| + r\left[\left|V_{i-1}^j\right| + \left|V_{i+1}^j\right|\right] \\ &\leq (1-2r)\max_i \left|V_i^j\right| + 2r\max_i \left|V_i^j\right| \\ &= \max_i \left|V_i^j\right|. \end{aligned}$$

Después haga $W^j = \max_i \left|V_i^j\right|$ y concluya que

$$W^{j+1} \leq W^j \text{ para todo } j = 0, 1, ...$$

lo que implica

$$W^j \leq W^0 \text{ para todo } j$$

7.10 Dos dimensiones

La más sencilla de las ecuaciones parabólicas en dos dimensiones es

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2},\ a < x < b,\ c < y < d, \tag{7.14}$$

con condición inicial dada en el rectángulo cuando $t = 0$ y condiciones de borde en los cuatro lados del rectángulo.

Los métodos vistos antes se pueden extender para este problema de forma completamente natural, sin embargo, la solución en el computador es laboriosa y no recomendable. Trabajamos con la siguiente convención: Los parámetros de discretización espacial son h_1 y h_2, correspondientes a las direcciones x y y respectivamente, los puntos en el eje x son $a + ih_1$, los puntos en el eje y son $c + jh_2$, el espaciamiento en tiempo está regido por el parámetro k como antes y la variable discreta de la ecuación en diferencias es de nuevo V. Un punto de la malla espacio-temporal está dado por $(a + ih_1, c + jh_2, sk)$, donde i, j, s son enteros.

Para escribir de forma compacta las ecuaciones en diferencias que corresponden a (7.14), utilizamos la siguiente convención:

I es el operador de diferencias correspondiente a la identidad.

D_i^2 es el operador de diferencias correspondiente a la segunda derivada en la dirección de x si $i = 1$ y y si $i = 2$. Entonces, por ejemplo, D_1^2 se define por la igualdad

$$D_1^2 V^s = \frac{1}{h_1^2}\left(V_{i-1,j}^s - 2V_{i,j}^s + V_{i+1,j}^s\right).$$

La extensión del método explícito (7.10) para (7.14) es

$$\frac{1}{k}\left(V_{i,j}^{s+1} - V_{i,j}^s\right) = D_1^2 V^s + D_2^2 V^s.$$

La validez de este método está dada por la condición de estabilidad, que es

$$k\left(\frac{1}{h_1^2} + \frac{1}{h_2^2}\right) \leq \frac{1}{2}.$$

Esta condición obliga a trabajar con un parámetro k exageradamente pequeño, lo cual no es deseable. De manera que no es recomendable utilizar este método.

De forma similar se pueden generalizar a la ecuación (7.14) el método implícito y el método de Crank-Nicolson. Este último es

$$\frac{1}{k}\left(V_{i,j}^{s+1} - V_{i,j}^s\right) = \frac{1}{2}\left[D_1^2 V^s + D_2^2 V^s + D_1^2 V^{s+1} + D_2^2 V^{s+1}\right].$$

El método de Crank-Nicolson para (7.14) es incondicionalmente estable, lo cual es conveniente, pero no es recomendable porque para cada paso en el sentido temporal, o sea de nivel s a nivel $s + 1$, el esfuerzo computacional que se requiere es grande.

7.10.1 Métodos ADI

En esta sección introducimos brevemente los métodos ADI, llamados así por la sigla en inglés que corresponde a Alternating Direction Implicit. Estos métodos fueron desarrollados a partir de 1955 por Peaceman, Rachford y Douglas, a quienes en la década de los sesenta se

unieron otros investigadores, entre los cuales mencionamos a Hubbard, D'Yakonov, Mitchell y Fairweather. Se trata de métodos potentes, especialmente si la ecuación está definida en un rectángulo, que mejoraron en gran medida la velocidad y exactitud de los cálculos y se usan intensamente desde antes de conocerse con precisión las condiciones para su validez. Sobre rectángulos y con condiciones de borde sencillas, estos métodos son incondicionalmente estables.

Uno de los métodos ADI más utilizados es conocido en la literatura como de Peaceman y Rachford y utiliza la importante estrategia predictor-corrector. Al resultado del paso predictor se le denota con el superíndice $s + \frac{1}{2}$ pero no significa que corresponda a los valores de la temperatura buscada en los puntos con coordenada temporal $\left(s + \frac{1}{2}\right) k$.

Este método se define así:

$$\begin{aligned} \left(I - \frac{k}{2}D_1^2\right) V^{s+\frac{1}{2}} &= \left(I + \frac{k}{2}D_2^2\right) V^s \\ \left(I - \frac{k}{2}D_2^2\right) V^{s+1} &= \left(I + \frac{k}{2}D_1^2\right) V^{s+\frac{1}{2}}. \end{aligned}$$

Cada uno de los pasos consiste en la solución de un sistema lineal de ecuaciones con matriz tridiagonal. Terminamos con un ejemplo trabajado en detalle y un ejercicio del mismo estilo del ejemplo pero que debe trabajarse con coordenadas cilíndricas para volverlo un problema unidimensional.

Ejemplo 40 *(Johnson y Riess, 1982, ejemplo 8.4) Una placa cuadrada determinada por el producto de intervalos* $[-1, 1] \times [-1, 1]$ *se encuentra a temperatura* 0 *cuando* $t = 0$ *y se sumerge en un medio de temperatura* 1. *Encuentre la distribución de temperatura de la placa en los tiempos* 0.5, 1 *y* 1.8.

Solución. No tenemos solución analítica pero el sentido común dice que la placa, eventualmente, debe adquirir la temperatura del medio. Preparamos el siguiente M-archivo para la solución de este ejercicio por el método ADI.

```
function vnew= edp2dadi3(n,tfin)
% Herramienta didactica, Carlos E. Mejia, 2002
% ecuacion u_t=u_xx+u_yy
% Johnson y Riess, ejemplo 8.4
% edp2dadi3
% dominio: cuadrado [-1,1]x[-1,1]
% es 0 cuando t=0, rodeado de medio a temperatura 1
% n: numero de subdivisiones en x e y
% tfin: tiempo en el que se pide la solucion
% metodo adi de Peaceman y Rachford
rh=zeros(n,1);
h=2/(n+1);
k=h/2;r=k/h^2;ro=2/r;
vold=zeros(n);vnew=vold;vint=vold;
% I-(k/2)D_2
sub=-ones(n-1,1);
d=(ro+2)*ones(n,1);
id2=diag(sub,-1)+diag(d)+diag(sub,1);
% I-(k/2)D_1 (distinto al anterior si espaciamiento en
% x es diferente de espaciamiento en y)
sub=-ones(n-1,1);
d=(ro+2)*ones(n,1);
id1=diag(sub,-1)+diag(d)+diag(sub,1);
t=0;
while t<=tfin
% metodo adi
% lado derecho predictor
for j=1:n
if j==1
rh(:)=1+(ro-2)*vold(:,1)+vold(:,2);
elseif j==n
rh(:)=vold(:,n-1)+(ro-2)*vold(:,n)+1;
else
rh(:)=vold(:,j-1)+(ro-2)*vold(:,j)+vold(:,j+1);
end
% solucion predictor
rh(1)=rh(1)+1;rh(n)=rh(n)+1;
```

```
v=id1\rh;
vint(:,j)=v(:);
end
% lado derecho corrector
for i=1:n
if i==1
rh(:)=1+(ro-2)*vint(1,:)+vint(2,:);
elseif i==n
rh(:)=vint(n-1,:)+(ro-2)*vint(n,:)+1;
else
rh(:)=vint(i-1,:)+(ro-2)*vint(i,:)+vint(i+1,:);
end
% solucion corrector
rh(1)=rh(1)+1;rh(n)=rh(n)+1;
v=id2\rh;
vnew(i,:)=v(:)';
end
%
t=t+k;vold=vnew;
end
```

La siguiente es una copia de los resultados que este programa entrega en pantalla.

```
>> vnew=edp2dadi3(8,1.8)
vnew =
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
1.0000 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 1.0000
1.0000 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 1.0000
1.0000 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 1.0000
1.0000 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 1.0000
1.0000 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 1.0000
1.0000 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
```

Este resultado es evidencia de la capacidad del método para resolver el problema. Debe tenerse cuidado con n o t grandes, pues el exceso de cálculos puede llevar a resultados anómalos mayores de 1 en algunos nodos y todo el proceso fracasa.

7.10.2 Ejercicio

Se tiene una esfera con centro 0 y radio 1 que en el tiempo $t = 0$ tiene temperatura 0. La esfera se sumerge en un medio de temperatura 1. Encuentre la distribución de temperatura en la esfera para $t = 0.5$, 1 y 1.8.

Sugerencia: Utilice coordenadas cilíndricas y suponga que la temperatura depende solamente de la distancia del punto al centro o sea que no depende ni de z ni de θ. Esta suposición se conoce como simetría circular, que también implica que $\left.\dfrac{\partial u}{\partial r}\right|_{r=0} = 0$.

La ecuación que nos ocupa es

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2}$$

y por simetría circular, se convierte simplemente en

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r}.$$

Esta ecuación tiene una singularidad cuando $r = 0$. Pero como $\left.\dfrac{\partial u}{\partial r}\right|_{r=0} = 0$, esta singularidad no ofrece dificultades pues, la expansión de Taylor de $\dfrac{\partial u}{\partial r}$ en torno a $r = 0$, nos permite concluir que $\left.\dfrac{1}{r}\dfrac{\partial u}{\partial r}\right|_{r=0} = \left.\dfrac{\partial^2 u}{\partial r^2}\right|_{r=0}$.

Más detalles sobre ésto pueden encontrarse en Smith, 1978 [28], pag. 40.

7.11 Ejercicios suplementarios

1. Utilice el método explícito, el método implícito y el método de Crank-Nicolson para resolver el problema

   $$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - u - 2e^{-t}$$

   para $x \in [0, 1]$ con condición inicial, condiciones de borde y solución exacta dadas por

   $$u(t, x) = x^2 e^{-t}.$$

2. Modifique los programas de computador que usó en el ejercicio anterior para que sirvan para el siguiente problema que incluye condiciones de borde con derivadas.

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

para $x \in [0,1]$ con condición inicial y solución exacta dadas por

$$u(t,x) = [sen(\pi x) + \cos(\pi x)] \exp\left(-\pi^2 t\right)$$

y con condiciones de borde

$$\frac{\partial u}{\partial x}(t,0) = \pi u(t,0)$$

$$\frac{\partial u}{\partial x}(t,1) = \pi u(t,1).$$

3. Resuelva el problema parabólico no lineal

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - u - x^2 - u^3,$$

con condiciones de borde homogéneas y condición inicial $u(0,x) = sen(\pi x)$.

4. Utilice el método ADI para resolver el problema

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2},$$

para $0 < x < 2$, $0 < y < 1$, con

$$u(0,x,y) = sen(\pi y)\, sen(2\pi x),$$

$$u(t,0,y) = u(t,2,y) = 0,$$

$$u(t,x,0) = u(t,x,1) = 0.$$

Sugerencia: La solución exacta es

$$u(t,x,y) = \exp\left(-5\pi^2 t\right) sen(2\pi x)\, sen(\pi y).$$

7.12 Examen de entrenamiento

Resuelva este ejercicio sin consultar libros ni notas de clase para darse una idea del nivel de preparación que ha obtenido hasta ahora. Una calculadora sencilla es lo mínimo que debe tener a disposición y es suficiente para poder responder el ejercicio. Claro que es preferible si dispone de una calculadora programable o un computador.

Enuncie en el mayor detalle posible, el método de Crank-Nicolson para resolver el siguiente problema:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},$$

con condiciones de borde y condición inicial dadas por

$$\begin{aligned} u\left(t,0\right) &= 0, \\ u\left(t,1\right) &= 1, \\ u\left(0,x\right) &= sen\left(\pi x\right) + x. \end{aligned}$$

Debe especificar el sistema lineal al que se llega y sería muy bueno si lo puede resolver, aunque sea solamente con matrices pequeñas.

8

Material de interés

Referencias generales
LAPACK
Templates y otras colecciones de software
MATLAB y otros

8.1 Referencias generales

Entre los libros de Métodos Numéricos que conocemos, mencionamos Stewart (1996) [29], Kincaid y Cheney (1994) [21], Mathews y Fink (2000) [23], Burden y Faires (1998) [6], Chapra y Canale (1999) [7] y James, Smith y Wolford (1985) [18] como libros básicos y los libros Isaacson y Keller (1994) [17], Atkinson (1978) [4] y Stoer y Bulirsch (1992) [30] como más avanzados.

Para el tema de Algebra Lineal Numérica específicamente, están, por ejemplo, Noble y Daniel (1989) [25], Trefethen y Bau (1997) [34] y Demmel (1997) [10]. Libros que se ocupan de computación científica son también referencias importantes. Entre ellos están Golub y Ortega (1993) [12] y el libro inconcluso Trefethen (1996) [32].

Enseguida nos referimos a herramientas computacionales que se justifica tener en cuenta a la hora de enfrentar los problemas de álgebra lineal numérica que se originan en la solución numérica de ecuaciones diferenciales. Las herramientas son de dos clases: las que se encargan de problemas generales o *cajas negras,* en las que el usuario confía aunque no entienda los algoritmos en los que están basadas y las que son hechas a la medida, con alto nivel de intervención por parte del usuario.

Todas las herramientas mencionadas aquí son de alta calidad, todas trabajan en ambiente UNIX o LINUX y la mayoría también trabajan en ambiente WINDOWS, pero ese no es su ambiente nativo y a menudo hay que hacer muchos ajustes para lograr que funcionen bien en esas

plataformas.

8.2 LAPACK

Entre las herramientas para problemas generales, destacamos a LAPACK (Linear Algebra Package) que es una colección de subrutinas escritas en FORTRAN 77 para resolver los problemas matemáticos más comunes que surgen a partir del modelamiento y que se enmarcan en el campo del álgebra lineal numérica. El desarrollo de LAPACK es una tarea que se empezó antes de 1980 y aun no termina. Al principio hubo dos colecciones de software, LINPACK que se ocupaba principalmente de la solución de sistemas lineales y de la descomposición en valores singulares y EISPACK, que se dedicaba a los problemas de valores propios. LAPACK es el sucesor de ambos y a LAPACK le siguen otras colecciones, como ScaLAPACK, por ejemplo, de manera que el esfuerzo por dotar a la comunidad de herramientas computacionales de primer nivel y de dominio público, continúa.

En Mejía, Restrepo y Trefftz [24], tuvimos la oportunidad de presentar a LAPACK a un público general y aquí reproducimos algunas de las ideas que expusimos allá. La colección LAPACK, descrita en LAPACK Users' Guide [2], se puede obtener libremente en NETLIB en la dirección

$$< http://www.netlib.org/lapack/ > .$$

Allá también se puede obtener la mejor documentación disponible sobre LAPACK, incluyendo la Guía mencionada arriba. Las subrutinas de LAPACK se basan en llamadas a unas subrutinas más sencillas que se conocen por la sigla BLAS (Basic Linear Algebra Subprograms). Hasta el momento se dispone de subprogramas BLAS de tres niveles: Nivel 1, publicado en 1979, que se encarga de operaciones de vector con vector. Nivel 2, publicado en 1988, que se ocupa de operaciones de matriz con vector. Nivel 3, publicado en 1990, que se dedica a operaciones entre matrices.

La construcción de LAPACK con base en los subprogramas BLAS genera un alto nivel de estandarización, proporciona gran rapidez de cálculo y hace más fácil hacer un seguimiento cuando se presentan dificultades. La utilización de LAPACK es universal, no solo directa-

mente, sino también como motor de cálculo en software con interfase gráfica como OCTAVE, <http://www.octave.org>, desarrollado en la Universidad de Wisconsin.

LAPACK incluye rutinas para resolver sistemas de ecuaciones lineales, sistemas de ecuaciones lineales por mínimos cuadrados, problemas de valores propios y problemas de valores singulares.

8.3 Templates y otras colecciones

Los autores de *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* [5], algunos de los cuales participan también del proyecto LAPACK, se orientan hacia la descripción de algoritmos y ofrecen programas fuente en diversos formatos, entre ellos FORTRAN, C y lenguaje MATLAB. Ellos consideran su colección de rutinas como de las hechas a la medida, pues proporcionan los medios para que el usuario adapte las rutinas a sus necesidades.

La preocupación de los autores es explicar los detalles, en ocasiones nada triviales, de los métodos iterativos más importantes en años recientes. Aunque incluyen métodos estacionarios, ellos están ahí solo por completez. Los temas principales del libro y por tanto los objetivos de los principales *templates,* son los métodos iterativos no estacionarios basados en subespacios de Krylov y diversas técnicas de precondicionamiento. Tanto el libro como las rutinas en los diferentes formatos, se pueden obtener en NETLIB, en la dirección

$$< http : /www.netlib.org/templates/ > .$$

Mencionamos finalmente otras colecciones importantes que no están en el dominio público.

La colección IMSL, ofrecida por Visual Numerics, con dirección internet

$$< http : //www.vni.com/products/imsl/ > .$$

Las colecciones ofrecidas por NAG, Numerical Algorithms Group, con dirección internet

$$< http : //www.nag.co.uk/ > .$$

Numerical Recipes, con dirección internet

$$< http://www.nr.com/ >,$$

las cuales están respaldadas por un grupo de libros muy bien escritos en los que se explican en detalle los algoritmos. Los libros se pueden conseguir en Internet libres de costo o se pueden comprar, junto con el software, a precio moderado.

8.4 MATLAB y otros

Los paquetes de software que integran una ágil interfase con algoritmos de calidad para una gran cantidad de problemas matemáticos, aparecieron hacia fines de los ochentas y mantienen su ascenso desde entonces. Para muchas tareas, estos paquetes son hoy la alternativa más económica en materia de tiempo pero no siempre en asuntos de dinero. Lo más importante que ofrecen es la capacidad de manipulación simbólica, la cual promete seguir siendo por un tiempo largo, esencial en la computación científica.

Entre estos paquetes, mencionamos a MATLAB, MATHEMATICA, MAPLE y MUPAD entre los de carácter comercial, con direcciones internet

$< http://www.mathworks.com/ >,$
$< http://www.wolfram.com/ >,$
$< http://www.maplesoft.com/flash/index.html >$ y
$< http://www.mupad.de/ >$

respectivamente.

Entre los paquetes de dominio público que por cierto, utilizan a LAPACK como base para sus rutinas de cálculo, están SCILAB y OCTAVE, con direcciones internet

$< http://www-rocq.inria.fr/scilab/ >$ y
$< http://www.octave.org >$.

Referencias

[1] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*, Dover, 1970.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide*, SIAM, 1992.

[3] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, 1995.

[4] K. E. Atkinson, *An Introduction to Numerical Analysis*, Wiley, 1978.

[5] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhoui, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1993.

[6] R. L. Burden and J. D. Faires, *Análisis Numérico*, International Thomson, sexta ed., 1998.

[7] S. C. Chapra and R. P. Canale, *Métodos Numéricos para Ingenieros*, McGraw Hill, tercera ed., 1999.

[8] W. Cheney and D. Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole Publishing Co., 1980.

[9] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, segunda ed., 1984.

[10] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

[11] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, 1971.

[12] G. Golub and J. M. Ortega, *Scientific Computing, An Introduction with Parallel Computing*, Academic Press, 1993.

[13] J. Hale and H. Kocak, *Dynamics and Bifurcations*, Springer-Verlag, 1991.

[14] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley, 1962.

[15] ——, *Elements of Numerical Analysis*, Wiley, 1964.

[16] D. J. Higham and N. J. Higham, *MATLAB Guide*, SIAM, 2000.

[17] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Dover, 1994.

[18] M. L. James, G. M. Smith, and J. C. Wolford, *Applied Numerical Methods for Digital Computation*, Harper and Row, tercera ed., 1985.

[19] L. W. Johnson and R. D. Riess, *Numerical Analysis*, Addison-Wesley, segunda ed., 1982.

[20] H. B. Keller, *Numerical Solution of Two Point Boundary Value Problems*, SIAM, 1990.

[21] D. Kincaid and W. Cheney, *Análisis Numérico*, Addison-Wesley Iberoamericana, 1994.

[22] B. López, *Solución numérica de problemas con valores en la frontera en dos puntos*, Trabajo de grado, carrera de Matemáticas, Universidad Nacional de Colombia, Medellín, 2000.

[23] J. H. Mathews and K. D. Fink, *Métodos Numéricos con MATLAB*, Prentice Hall, tercera ed., 2000.

[24] C. E. Mejía, T. Restrepo, and C. Trefftz, *Lapack una colección de rutinas para resolver problemas de algebra lineal numérica*, Revista Universidad EAFIT, 123 (2001), pp. 73–80.

[25] B. Noble and J. W. Daniel, *Algebra Lineal Aplicada*, Prentice Hall, tercera ed., 1989.

[26] J. M. Ortega, *Numerical Analysis, a second course*, SIAM, 1990.

[27] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Interscience Publishers, segunda ed., 1967.

[28] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, segunda ed., 1978.

[29] G. W. Stewart, *Afternotes on Numerical Analysis*, SIAM, 1996.

[30] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer Verlag, segunda ed., 1992.

[31] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth y Brooks/Cole, 1989.

[32] L. N. Trefethen, *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*, disponible en http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/pdetext.html, 1996.

[33] ——, *Spectral Methods in MATLAB*, SIAM, 2000.

[34] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.

[35] H. van der Vorst, ed., *Copper Mountain Conference Proceedings*, SIAM, 2001. Publicadas como Vol. 23, No. 2 de SIAM Journal of Scientific Computing.